COMPUTER MODELLING GROUP LTD.

GEM USER GUIDE

GEM

Compositional & Unconventional Reservoir Simulator

Version 2016

cmgl.ca

16.GE.M3

This publication and the application described in it are furnished under license exclusively to the licensee, for internal use only, and are subject to a confidentiality agreement. They may be used only in accordance with the terms and conditions of that agreement.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic, mechanical, or otherwise, including photocopying, recording, or by any information storage/retrieval system, to any party other than the licensee, without the written permission of Computer Modelling Group.

The information in this publication is believed to be accurate in all respects. However, Computer Modelling Group makes no warranty as to accuracy or suitability, and does not assume responsibility for any consequences resulting from the use thereof. The information contained herein is subject to change without notice.

Copyright © 2016 Computer Modelling Group Ltd.

All rights reserved.

Reprise License Manager (RLM)
Copyright © 2006-2016, Reprise Software, Inc.
All rights reserved.

GEM uses Intel(R) Compilers.

GEM, CMG, and *Computer Modelling Group* are registered trademarks of Computer Modelling Group Ltd. All other trademarks are the property of their respective owners.

Computer Modelling Group Ltd.

200, 1824 Crowchild Trail N.W.

Calgary, Alberta Canada T2M 3Y7

Tel: (403) 531-1300 **Fax:** (403) 289-8502 **E-mail:** cmgl@cmgl.ca

Preface

GEM is CMG's advanced general equation-of-state compositional simulator which includes options such as equation-of-state, dual porosity, CO_2, miscible gases, volatile oil, gas condensate, horizontal wells, well management, complex phase behavior and many more. GEM was developed to simulate compositional effects of reservoir fluid during primary and enhanced oil recovery processes.

This User Guide presents the important mechanisms which occur during gas injection processes and the equations that describe those mechanisms. The techniques used in GEM to solve these equations are also described. This User Guide is aimed at reservoir engineers who want to use GEM to design gas injection processes for enhanced oil recovery. It requires some basic knowledge of reservoir engineering and some rudimentary exposure to reservoir simulation. This User Guide provides a step-by-step procedure for preparation of input data for this program. A tutorial section is provided as well as a set of appendices.

Every attempt has been made in the preparation of this User Guide to provide the user with all the necessary details. If questions arise, please contact:

Computer Modelling Group Ltd.
200, 1824 Crowchild Trail N.W.
Calgary, Canada
T2M 3Y7
Telephone: (403) 531-1300 Fax: (403) 289-8502 E-mail: cmgl@cmgl.ca

Confidentiality: All components of CMG technology including software and related documentation are protected by copyright, trademark and secrecy. CMG technology can be used only as permitted by your license from CMG. By the license, you have agreed to keep all CMG technology confidential and not disclose it to any third party. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic, mechanical, or otherwise, including photocopying, recording, or by any information storage/retrieval system, to any party other than the licensee, without the written permission of Computer Modelling Group.

Corrections/Errors: CMG ENDEAVORS TO PRODUCE TECHNOLOGY OF THE HIGHEST QUALITY; NEVERTHELESS ERRORS OR DEFICIENCIES IN SUCH TECHNOLOGY ARE INEVITABLE. IF YOU FIND AN ERROR OR DEFICIENCY, YOU ARE REQUESTED TO PROVIDE DETAILS OF IT AND ILLUSTRATIVE DATA SET(S) TO CMG SUFFICIENT TO PERMIT CMG TO REPRODUCE THE ERROR OR DEFICIENCY. CMG SHALL ENDEAVOR TO REMEDY A DEFICIENCY IN A TIMELY MANNER AND SHALL PERIODICALLY REPORT TO YOU AS TO THE STEPS BEING TAKEN TO REMEDY THE DEFICIENCY. THE RESPONSE TIME FOR A DEFICIENCY MUST BE PRIORITIZED FOR THEIR GENERAL APPLICATION TO CMG MEMBERS AND WHETHER THEY FORM PART OF A CMG PROGRAM. CMG DOES NOT WARRANT THAT DEFICIENCIES WILL BE REMEDIED.

Limited Liability: CMG does not warrant the accuracy or usefulness of the technology and software - Refer to your license.

Contents

Introduction 1

Important Changes in GEM 2016.10 1
Important Changes in GEM 2015.10 9
Important Changes in GEM 2014.10 16
Important Changes in GEM 2013.10 23
Important Changes in GEM 2012.10 29
Important Changes in GEM 2011.10 33
Important Changes in GEM 2010.10 39
Important Changes in GEM 2009.10 42
Important Changes in GEM 2008.10 46
Important Changes in GEM 2007.10 50
Important Changes in GEM 2006.10 53
Important Changes in GEM 2005.10 54
Important Changes in GEM 2004.10 56
Important Changes in GEM 2003.10 57
Important Changes in GEM 2002.10 58
Important Changes in GEM 2001.10 59
Important Changes in GEM 2000.10 61
Introduction to GEM 62

Tutorial 65

Introduction 65
Data Groups in the Keyword Input System 66
How to Document Your Data Set 67
How to Do a Restart 68
Controlling Contents of the Output Print File 70
Controlling Contents of the Simulation Results File 71
Describing Your Grid System 72
Describing Refined Grid 74
Entering Null Blocks 75
Using Dual Porosity/Dual Permeability 76
Using the Aquifer Option 77
Problems with Small Timesteps or Long Execution Times 78
Simulating a Single Phase Reservoir 81

Horizontal Wells 82
Vertical Equilibrium Calculation 83
Defining Wells 85
How to Shut In a Well and Reopen It 86
Using the Wellbore Model 88
Operating and Monitoring Constraints 89
Entering Well Indices 91
Stopping a Simulation Run 92
Guidelines for Setting Up Well Data 93
Recurrent Data from Other Sections 95
Parallel GEM 96
Saline Water Injection 98
Using the Oil Wet Option 99
Two and Three Point Scaling of Relative Permeability Values 105
Modeling Flow in Fractured Shale Gas Reservoirs 108
Modelling Non-Darcy Flow in Hydraulic Fractures Accurately Using a Grid Based Approach 121
Introducing a Hydraulic Fracture in Recurrent Data 128
Using the Thermal Option in GEM 129
One-way Geomechanics Calculations using Post-Processing of Prior Simulator Response (GEOPOSTPRO) 132

Keyword Data Entry System **137**

Introduction to Keyword System 137
Comments (Optional) 142
Blank Lines (Optional) 143
Scan Mode for Checking Errors 144
Include Files (Optional) 145
Controlling Data File Listing (Optional) 146
Changing the Comment Indicator (Optional) 147
Changing the Keywords by Using Translate Rules (Optional) 148
User Block Address 149
Input of Grid Property Arrays 150
Entering Matrix Grid Properties 152
Entering Fracture Grid Properties 153
Entering Refined Grid Properties 154
Assign Data to Block Groups 155
Assign Data to Structured Block Groups 157
Assign Property to Fracture Zone (Conditional) 159
Assigning Grid Properties to all Elements 161
Constant Value Arrays 162
Array Input in IJK Notation 163
Array Input of Values that Vary in the I Direction 165

Array Input of Values that Vary in the J Direction....166
Array Input of Values that Vary in the K Direction....167
Values that Vary for Most or All Grid Blocks....168
Values Stored in Binary Form....169
J and K Direction Data from I Direction....171
Modifying Array Data (Conditional)....172
Interpolating Table Data (Optional)....176

Input/Output Control **177**

Notes on Input/Output Control....177
Command-Line Arguments (Optional)....178
Input/Output File Names (Optional)....184
Run-Time Dimensioning (Optional)....189
Interrupt and CTRL-C Control (Optional)....192
Project Main Title (Optional)....193
Project Second Title (Optional)....194
Project Third Title (Optional)....195
Case Identification (Optional)....196
Check Only (Optional)....197
Diary Format (Optional)....198
Input Data Units (Optional)....201
Load Balancing Status Printing for Non-Solver Tasks (Optional)....203
Level and Class Structure Printing for Parallel Linear Solver (Optional)....204
Data Range Checking (Optional)....205
Maximum Number of Error Messages (Optional)....206
Separator Stage Composition Output (Optional)....207
Restart Record Writing Frequency (Optional)....208
Restart Timestep (Optional)....211
Output Printing Frequency (Optional)....213
Items in the Output Print File (Optional)....215
Selection of Results File Format (Optional)....223
Selection of Additional Plain Text SR2 Main File (Optional)....224
Floating-Point Precision for Writing Output to SR2 Main Files (Optional)....225
Simulation Results File Format (Optional)....226
Simulation Results Writing Frequency (Optional)....228
Items in Simulation Results File (Optional)....231
CO_2 Inventory Output (Optional)....248
Geochemical Reactions Output (Optional)....249
Well, Aquifer and Sector Summary Tables (Optional)....250
Debug Output (Optional)....251
Seek Parent Chain of Prior Simulator Response for Geo-Post Processing (Optional)....253

Reservoir Description 255

Notes on Reservoir Description 255
Fundamental Grid Definition (Required) 256
K Direction Index (Optional) 263
Convert Cartesian Grid to Corner Point (Optional) 264
Nine-Point Spatial Discretization (Optional) 265
Block Dimensions for the I Direction (Conditional) 267
Block Dimensions for the J Direction (Conditional) 269
Block Dimensions for the K Direction (Conditional) 271
Depth (Conditional) 273
Depth to the Tops of Grid Blocks (Conditional) 275
Depths to Centre of Pay (Conditional) 277
Depths to Top of Block (Conditional) 279
Grid Tilt Angles (Conditional) 281
Corner-point Node Locations (Conditional) 283
Convert Corner Point Grid to Node Based Grid (Conditional) 288
Corner Point Depths for Corner Point Grids (Conditional) 289
Lateral Corner Point Locations for Corner Point Grids (Conditional) 291
Line-Based Corner Point Locations for Corner Point Grids (Conditional) 293
Complete Corner Point Locations for Corner Point Grids (Conditional) 295
Local Refined Grid (Conditional) 298
Define Planar Fracture Local Grid (Conditional) 310
Planar Fracture Template 314
Dual Porosity (Optional) 320
Dual Permeability (Optional) 321
Dual Porosity Subdomain Method (Optional) 322
Assigning Fractional Volumes to Subdomain blocks (Optional) 324
Subdomain to Subdomain Flow Transmissibility Multipliers (Optional) 326
Dual Porosity MINC Method (Optional) 328
Shape Factor Calculation (Conditional) 329
Matrix-Fracture Transfer Calculation (Conditional) 331
Fracture Spacing (Conditional) 332
Null Block Indicator (Optional) 334
Porosity (Required) 336
Pore Volume (Optional) 338
Rock Compressibility 339
Pore Volume Modifiers (Optional) 341
Block Geometry Modifiers (Optional) 343
Permeabilities (Required) 351
Netpay (Optional) 353
Net to Gross Multipliers (Optional) 355
Transmissibility Multipliers (Optional) 356
Diagonal Transmissibility Multipliers (Optional) 359
Transmissibility Multipliers for Lower Indexed Cell Blocks (Optional) 361

Transmissibility Multiplier for Matrix-Fracture Flow (Optional) 363
Transmissibility Multiplier Action (Optional).....364
Transmissibility Multiplier Regions (Optional).....366
Inter Region Transmissibility Multiplier (Optional).....367
Transmissibility Values (Optional).....369
Faults (Optional).....372
Aquifers (Optional).....374
Aquifer Pressure Influence (Conditional).....380
Pore Volume Cut-Off Threshold (Optional).....382
Pinch Out Array (Optional).....383
Pinchout Tolerance (Optional).....385
Corner Point Tolerance (Optional) 387
Block Groups (Optional) 388
Structured Block Groups (Optional).....390
Sectors (Optional).....392
Sector Array (Optional) 394
Sector Assignment via Names and an Array (Optional).....395
Flux Sectors (Optional).....396
Irregular Connection (Optional) 398
Special Connections (Optional).....401
Interblock Matrix-Fracture Connection (Optional).....403
Fault Array (Optional) 405
Compaction/Dilation Rock Type (Optional).....407
Compaction/Dilation Rock Compressibility (Optional) 410
Compaction Rock Table (Optional).....411
Compaction Hysteresis Rock Table (Optional) 413
Compaction Irreversibility Flag (Optional) 416
Dilation Rock Table in Elastic Zone (Optional) 417
Dilation Rock Table in the Plastic Zone (Optional) 419
Dilation Rock Table in Unloading Zone (Optional) 421
Dilation Rock Table in Recompacting Zone (Optional).....423
Dilation Rock Table in Reloading Zone (Optional).....425
Treat CCPOR / CROCKTAB / CROCKTABH as Depletion Parameters/
Tables (Optional) 433
Compaction/Dilation Rock Region (Optional) 435
Palmer and Mansoori Parameters (Optional).....441
Fault Transmissibilities (Optional) 447
Cap Rock Leakage Using Wells (Optional).....449
Thermal Rock Properties (Optional).....452
Heat Loss (Optional).....453

Other Reservoir Properties **457**

Summary of Other Reservoir Properties.....457
Indicate End of Grid Definition (Conditional).....458

Thermal Expansion Coefficient (Optional) 459

Component Properties **461**

Notes on Component Properties 461
Fluid Model (Required) 464
Number of Components (Required) 465
Component Name (Required) 466
Equation of State Set (Optional) 468
Equation of State Model Type (Optional) 472
User Component Properties (Conditional) 473
Equation of State Omega Parameters (Optional) 476
User Component Interaction Coefficients (Conditional) 477
Parameters for Hydrocarbon Interaction Coefficients (Optional) 478
Critical Property Correlations for User Components (Conditional) 479
Volume Shift Parameters (Optional) 480
Heating Values (Optional) 482
Equation of State Parameters at Surface Conditions (Optional) 485
Reservoir Temperature (Required) 487
Identity of Single Phase Fluid (Optional) 489
Average Saturation Pressure (Optional) 492
Viscosity Correlation Specification (Optional) 493
Parameters for Computing Viscosity of Hydrocarbon Fluids (Optional) 497
Aqueous Phase Properties (Optional) 501
Aqueous Phase Viscosity with Polymer Components (Conditional) 509
Nonlinear Mixing Rule for Aqueous Phase Viscosity in Presence of Polymer Components 512
Effect of Salinity on Polymer Viscosity (Conditional) 514
Convert Velocities to Shear Rates (Conditional) 515
Viscosity Shear Dependence Option (Optional) 516
Shear Effects on Polymer Viscosity (Conditional) 517
Shear Effects on Polymer Viscosity – Tabular input (Conditional) 519
Implicit calculation of shear effects (Conditional) 520
Reference Polymer Concentration (Conditional) 521
Solubility Data (Optional) 522
Gas-Phase Diffusion in Fractured Reservoirs (Optional) 526
Flash Method Technique (Optional) 528
Method for Evaluating Entries of the Jacobian Matrix (Optional) 531
Molecular Diffusion (Optional) 533
Asphaltene Precipitation Model (Optional) 537
Solid Adsorption (Optional) 539
Irreversible Asphaltene Model (Optional) 541
Solid Deposition Model (Optional) 543
Disallow Solid-Model Inheritance (Optional) 545
Coal Matrix-to-Cleat Modelling Parameters(Optional) 546

Gas Adsorption for Water Filled Blocks (Optional) 549
Selection of Model Accounting for Coal Fracture Porosity and Absolute Permeability Changes (Optional) 551
Trace Component (Optional) 552
EOS Cubic Root Selection (Optional) 553
H2O Vaporization (Optional) 554
GHG Option - Aqueous Components 555
GHG Option - Mineral Components 556
GHG Option – Reactions 557
GHG Option – Reactions 560
GHG Option – Reaction Equilibrium and Rate Parameters 561
GHG Option - Equilibrium-Rate-Annihilation (ERA) Matrix 564
GHG Option – Setting of Initial Mineral Equilibrium Reaction Rates and Aqueous Molalities 565
GHG Option - Aqueous Phase Model (Optional) 568
GHG Option – Factors for CO2 Inventory (Optional) 570
Polymer Degradation (Conditional) 571
Polymer Computation Options (Optional) 573
Ion-Exchange – Component Definitions 574
Polymer Option – Component Definition 577
Surfactant – Component Definition 578
Resistance Factor Calculation (Optional) 579
Thermal Option (Optional) 581
EOS Enthalpy Calculation Method (Optional) 582
Critical Delta Density Threshold(Optional) 583

Rock-Fluid Data **585**

Notes on Rock-Fluid Data 585
Start of Rock-Fluid Property Input (Required) 591
Tabular Relative Permeabilities 592
Interpolation Component (Optional) 594
Interpolation Set Number and Parameters (Optional) 596
Foam Component Name (Conditional) 599
Interfacial Tension Table 600
Foam-Model Definition 602
Basic Foam Interpolation Parameters 603
Foam Dry-Out Interpolation Parameters 608
Factor for Foam Water Relative Permeability (Optional) 611
Water-Oil Relative Permeability Table (Conditional) 612
Liquid-Gas Relative Permeability Table (Conditional) 618
Rock Compressibility for Each Rock Type (Optional) 624
Hysteresis Parameters (Optional) 625
WAG Relative Permeability Hysteresis Modelling (Optional) 630
Rock Type (Optional) 636

Trapped Oil Saturation (Optional) 637
Interfacial Tension Effects on Relative Permeability (Optional) 638
Velocity Dependent Relative Permeability Option (Optional) 641
Single Phase Gas Relative Permeability Flag (Optional) 646
Method for Evaluating 3-Phase Kro (Optional) 647
Saturation Endpoints for each Grid Block (Optional) 650
Leverett J Function (Optional) 655
Maximum Relative Permeability/Capillary Pressure Values for Each Grid Block (Optional) 658
Water-Oil Capillary Pressure (J Function) Shift for Each Grid Block (Optional) 661
Cation Exchange Capacity for Each Grid Block (Optional) 662
Cation Exchange Capacity through Correlation (Optional) 663
Multicomponent Adsorption from the Gas Phase – Langmuir Adsorption Input (Optional) 665
Multicomponent Adsorption from the Gas Phase – Tabular Adsorption Data Entry (Optional) 667
Multicomponent Adsorption from the Aqueous Phase –Langmuir Adsorption Input (Optional) 672
Multicomponent Adsorption from the Aqueous Phase – Tabular Adsorption Data Entry (Optional) 674
Aqueous Phase Adsorption Controlling Data (Optional) 677
Rock Mass Density (Optional) 679
Velocity-Dependent Dispersion (Optional) 680
Coal Bed Methane Features (Optional) 682
Flag to Use Alternate Relative Permeability Table Definition for Coal Bed Methane (Optional) 685
Subdomain Reinfiltration (Optional) 686
Single-Phase Relative Permeability Interpolation (Optional) 688
Non-Darcy Flow in Reservoir (Optional) 690
Non-Darcy Coefficient Correction Factor (Conditional) 693
STARS Scaling Option (Optional) 694
Langmuir Maximum Adsorption Parameter Multiplier (Optional) 695
Three-point Endpoint Scaling (Optional) 696
Table Based Scaling: Saturation Endpoints for Relative Permeability (Optional) 699
Table Based Scaling: Maximum Relative Permeability Values (Optional) 704
Residual Oil Saturation Calculation for STONE 1 Relative Permeability Model (Optional) 709
Saturation Endpoint Interpolation (Optional) 716

Initial Conditions 719

Notes on Initial Conditions 719
Initial Conditions Identifier (Required) 727
Vertical Equilibrium or User Input Selection (Required) 728
Number of Initialization Regions (Optional) 733

Specification of Initialization Regions (Optional) 734
Oil Zone and Gas Cap Compositions (Conditional) 735
Initial Reservoir Fluid Properties (Conditional) 738
Initial Reservoir Temperature (Optional) 740
Reference Depth and Reference Pressure (Conditional) 741
Depth to Water-Oil, Gas-Oil, and Water-Gas Contacts (Conditional) 742
Water Saturation Below Water-Oil Contact (Optional) 745
Initial Reservoir Fluid Formation Volume Factors (Optional) 746
Separator for Initial Fluid-In-Place Calculations (Optional) 747
Specification of Non-EOS Stream Density Calculations (Optional) 753
Standard Pressure, Temperature, and EOS Set (Optional) 757
Table Defining a Gas Plant (Optional) 759
Critical Depth (Conditional) 763
Variation of Composition with Depth (Conditional) 764
Datum Depth Specification (Optional) 766
Specification of Capillary Pressure Values at Contacts (Optional) 768
Water Saturation Input Used with Gravity-Capillary Initialization (Conditional) 770
Non Equilibrium Water and Oil Saturation Used with Gravity-Capillary Equilibrium Initialization (Conditional) 772
Specification of Method for Computing Oil Saturations in Gas Zone 773
Critical Temperature Multiplier (Optional) 775
Initial Aqueous/Mineral Concentrations 777

Numerical Methods Control 779

Notes on Numerical Methods Control 779
Numerical Methods Control Identifier (Optional) 781
Maximum Timesteps (Optional) 782
Maximum CPU Seconds, Maximum Elapsed Seconds (Optional) 783
Maximum and Minimum Timestep Size (Optional) 784
Normal Variation in Variables Per Timestep (Optional) 785
Adaptive-Implicit Switching (Optional) 788
Convergence of Newton's Method (Optional) 791
Maximum Newton Cycles for Constraint Equation Residual Checking (Optional) 794
Maximum Newtonian Cycles (Optional) 795
Underrelaxation Option (Optional) 796
Convergence Tolerance for Linear Solver (Optional) 797
Orthogonalization (Optional) 798
Solver Equation Ordering (Optional) 799
Solver Factorization Degree (Optional) 800
Pivot Stabilization (Optional) 801
Modified ILU Factorization (Optional) 802
Maximum Iterations (Optional) 803
Numerical Dispersion Control (Optional) 804
AIMSOL/PARASOL (Optional) 805

Number of Parasol Classes (Optional) 806
Red-Black Ordering Check for Parasol (Optional) 807
Factorization Degree Within Parasol Classes (Optional) 808
Factorization Degree Between Parasol Classes (Optional) 809
Parasol Class Partitioning Pattern (Optional) 810
Parallel Equation and Residual Building (Optional) 816
Load-Balancing in Parallel Equation and Residual Building (Optional) 818
Parallel Jacobian and Residual Building based on Parasol Classes (Optional) 820
Number of Threads (Optional) 822
Number of Threads for Parallel Grid-Processing (Optional) 823
Number of Threads for Parallel Geomechanics-Processing (Optional) 824
Setting Schedule for OPENMP Parallelization (Optional) 825
Automatic Setting of KMP_AFFINITY 828
Solver Pointer Generation (Optional) 830
Constraint Equation Residual Reduction (Optional) 831
Iterations for Constraint Equation Residual Reduction (Optional) 832
Cache Alignment (Optional) 833
Fully Implicit Alignment (Optional) 834
Adaptive Implicit Alignment (Optional) 835

Geomechanics **837**

Summary of Geomechanical Model 837
Geomechanical Model Identifier (Optional) 846
3D Finite Element 847
Independent Geomechanics Grid 848
Independent Geomechanics Graphics 855
Global Amalgamation 859
Extending Geomechanics Grid 862
Plane Strain Option 867
Deformation Rock Type 868
Plastic Model Formation Properties 869
Temperature-dependent Properties 871
Porosity-Dependent Properties 873
Pressure-Dependent Properties 875
Stress-Dependent Properties 877
STARS: Solid Component Properties 879
Yield Criterion 883
Cap Model 885
Cap Model 1 886
Nonlinear Constitutive Model 890
Nonlinear Elastic Constitutive Model 1 891
Nonlinear Elastic Constitutive Model 2 893
Creep Model 900
Creep Model 1, 2 902

STARS: Pseudo Dilation Model 906
Generalized Plasticity Model 909
Single Surface Failure Model 915
Modified Cam Clay Model 921
Jointed Rock 926
Thermal Expansion Coefficient 930
Non-Uniform Formation Properties 931
Matrix Permeability Multiplier 933
Fracture Permeability Multiplier 939
Barton-Bandis Fracture Permeability 942
Fracture Direction 946
Dilation Relative Permeabilities 947
Other Dilation Properties 949
Well Radius 950
Stiffness Matrix Calculation Option 951
Deformation Solution Control 952
Geomechanics AIMSOL Control 956
Dimension Over-Rides (Optional) 959
Initial Stress Distribution (2D) 960
Initial Stress Distribution (3D) 962
Geomechanical Reference Block 967
Prescribed Boundary Conditions (2D) 968
Prescribed Boundary Conditions (3D) 974
Point Loads (2D) 978
Point Loads (3D) 982
Distributed Edge Loads (2D) 984
Distributed Surface Loads (3D) 988
Gravity Loads (2D) 993
Gravity Loads (3D) 995
Fixed Null Block 997
Fixed Cap Rock 999
Geomechanics Domain 1000
Pressure Boundary Domain 1002
Coupling Options 1005
Geomechanical Coupling Factor 1009
Pressure Tolerance Multiplier 1010
Coupling Update Times 1011
Porosity Calibration 1013
Iterative Coupling to Fluid Flow 1015
Boundary Stress Unloading 1017
STARS: Geomechanics Post Processing 1018
GEM: Geomechanics Post Processing of Prior Simulator Response (Optional) 1021
GEM: Data Source from Prior Simulator Response for Geo-Post-Processing (Optional) 1023

Well and Recurrent Data 1025

Notes on Well and Recurrent Data for the 2016.10 Release 1025
Notes on Well and Recurrent Data for the 2015.10 Release 1026
Notes on Well and Recurrent Data for the 2014.10 Release 1028
Notes on Well and Recurrent Data for the 2013.10 Release 1030
Notes on Well and Recurrent Data for the 2012.10 Release 1032
Notes on Well and Recurrent Data for the 2011.10 Release 1033
Notes on Well and Recurrent Data for the 2010.10 Release 1034
Notes on Well and Recurrent Data for the 2009.10 Release 1036
New Keywords and Options for the 2008.10 Release 1037
Notes on Well and Recurrent Data for the 2007.10 Release 1038
Notes on Well and Recurrent Data for the 2006.10 Release 1041
Notes on Well and Recurrent Data for the 2005.10 Release 1042
Notes on Well and Recurrent Data for the 2004.10 Release 1043
Notes on Well and Recurrent Data for the 2003.10 Release 1044
Notes on Well and Recurrent Data for the 2002.10 Release 1045
Notes on Well and Recurrent Data for the 2001.10 Release 1046
Notes on Well and Recurrent Data for the 2000.10 Release 1047
Placement of Keywords in the Well and Recurrent Data 1048
Note on Recycling and Well Stream Compositional Monitoring 1050
Miscellaneous Notes 1051
Well Management and Group Control 1053
Specifying the Well and Group Control Hierarchy 1055
Notes on Group Well Control 1056
Limitations of the Well Management and Group Control Module 1060
Recurrent Data from Other Sections 1062
Well and Recurrent Data Identifier (Required) 1064
Well Change Date (Conditional) 1065
Well Change Time (Conditional) 1066
Well Change First Timestep Size (Optional) 1067
Maximum and Minimum Timestep Size (Optional) 1068
Setting Grid Blocks to Implicit or Explicit (Optional) 1069
Setting Well Blocks and Neighbours to Implicit (Optional) 1071
Set Frequency of Initialization of Bottom-Hole Pressure (Optional) 1073
Group Identification (Optional) 1076
Define Reporting Group (Optional) 1080
Well Identification (Required) 1082
Well Type Definition (Required) 1085
Shut in Wells above Formation (Optional) 1086
Well Status Definition (Optional) 1088
Shut and Reopen a List of Wells (Optional) 1090
Tubing Data for Injectors (Conditional) 1092
Tubing Data for Producers (Conditional) 1095
Wellbore Temperature (Optional) 1100

Set Application Mode of Correlation for Well Head Pressure (Optional).....1101
Set Number of Points for WHP Root Search (Optional) 1103
Composition of Injected Fluid (Conditional).....1105
Well Operating Constraints (Required) 1107
Default Bottom-hole Pressure Constraint Values (Optional) 1118
Set Injection Composition (Optional) 1119
Well Current Constraint (Optional) 1121
Maximum Number of Continue-Repeat (Optional).....1122
Monitored Well Constraints (Optional).....1123
Alter Primary Well Operating Constraint Value (Optional).....1131
Alter Well Constraint Value (Optional).....1133
Operating Constraints in History Matching Mode (Optional).....1136
Alter Observed Flow Rates for History-Matching (Optional).....1138
Well Head Method (Optional) 1140
Well Head Iteration (Optional).....1142
Simultaneous BHP and Well Head Iteration (Optional).....1144
Perforations in Inactive Blocks (Optional) 1146
Well Backflow Model (Optional) 1148
Data for Workover Action for Wells (Optional).....1150
Resetting Well Operating Constraint after Value Change (Optional) 1153
Gas Lift Option (Optional).....1155
Gas Lift Control (Optional) 1158
Gas Lift Optimization (Optional).....1160
Well Separators (Conditional) 1163
Specification of Non-EOS Stream Density Calculations (Optional) 1169
Table Defining Gas Plant (Optional) 1174
Standard Pressure, Temperature, and EOS Set (Optional) 1178
Well Element Geometry (Optional).....1180
Location of Well Completions (Conditional) 1183
Auto Thresholding for Well indices (Optional).....1192
Well Geometry Parameters (Optional) 1194
Geometric Data for Deviated Well Completions (Conditional) 1196
Simplified Geometric Data for Deviated Well Completions (Conditional).....1200
Relative Permeability Data for Well Completions (Optional).....1202
Pressure Gradients for Calculation of Pressure Differences Between Completions (Conditional).....1207
User-Specified Reference Depth for Well BHP (Optional).....1210
User-Specified Pressure Gradient For Reference Depth for Well BHP (Optional).....1212
Gas Recycling Option (Optional) 1214
Group Production Constraints (Optional).....1216
Group Injection Constraints (Optional) 1221
Fuel Consumption Fraction for Group Gas Recycling (Optional).....1231
Fuel Component Mask for Group Gas Recycling (Optional).....1233

Maximum Fuel Consumption Rate for Group Gas Recycling (Optional)..... 1235
Make-up Gas Composition for Group Gas Recycling (Optional)..... 1237
Maximum Make-up Gas Rate for Group Gas Recycling (Optional)..... 1239
Gas Make-up Target for Group Gas Recycling (Optional)..... 1241
Gas Producing Group for Group Recycling / Voidage Replacement (Optional)..... 1243
Injection Group for Group Recycling/Voidage Replacement (Optional)..... 1245
Recycled Gas Component Mask for Group Gas Recycling (Optional)..... 1247
Maximum Re-injection Rate for Group Gas Recycling (Optional) 1249
Sales Rate Fraction for Group Gas Recycling (Optional)..... 1251
Sales Component Mask for Group Gas Recycling (Optional) 1253
Maximum Sales Rate for Group Gas Recycling (Optional)..... 1255
Maximum Water Make-up Rate for Group Water Recycling (Optional)..... 1257
Water Make-up Target for Group Water Recycling (Optional)..... 1259
Water Producing Group for Group Recycling (Optional) / Production Group for Voidage Replacement by Water Injection..... 1261
Maximum Re-injection Rate for Group Water Recycling (Optional)..... 1263
Monitored Group Constraints (Optional)..... 1265
Defining Group Production or Injection as Going Through a Manifold (Optional) 1269
Pressure-Constraint Translation for Manifolds (Optional)..... 1271
Specification of Hydraulics Tables for Calculating Pressure Difference Between Manifold and Surface (Optional)..... 1273
Manifold Depth for Calculating Pressure Difference Between Manifold and Surface (Optional) 1275
Group Artificial Lift Quantity Value (Optional)..... 1277
Well Artificial Lift Quantity Value (Optional)..... 1279
Priority List for Automatic Drilling of Wells (Optional) 1281
Drainage Radius of Wells for IPPDRINP based Autodrill Algorithm (Optional) 1285
Alternate Phase for Ranking Wells on Auto Drill (Optional) 1287
Rate Cutoff Thresholds for Auto Drill (Optional)..... 1290
Group Apportionment Options (Optional)..... 1292
Apportionment Method for Meeting Group Targets (Optional) 1294
Priority Formulae for Apportionment (Conditional)..... 1298
Guide Rates for Groups or Wells (Optional)..... 1302
Flag for Specifying Groups or Wells Under or not Under Group Apportionment (Optional)..... 1305
Well/Group On-time Fraction (Optional)..... 1307
Sets/Alters a Well’s Phase Productivity (Optional) 1311
Individual Layer Controls (Optional)..... 1316
Hydraulic Pressure Table (Conditional)..... 1318
Allow a Set of Keywords to be Processed When a Specified Condition (Trigger) is Satisfied (Optional)..... 1326
Alter Well Constraint Value via a Multiplier or an Increment (Optional) 1350

Group Production Constraints Multiplier (Optional) 1354
Group Injection Constraints Multipliers (Optional) 1356
Allow a Cycling Group to be Defined (Optional) 1358
Allow Cycling Group Data to be Revised (Optional) 1369
Allow Cycling Group Data to be Revised (Optional) 1371
Select Cycling Group Cycle Part to Start and End Cycling Group Control (Optional) 1373
Control Layer Lumping Option (Optional) 1375
Report layer lumping option (Optional) 1381
Control layer lumping option (Optional) 1383
Control layer lumping option (Optional) 1385
Control layer lumping option (Optional) 1387
Selection of Instantaneous Vs. Actual Rates Lumps (Optional) 1388
Declaring Cap Rock Leakage Wells (Optional) 1389
Salinity of Injected Water (Optional) 1391
Temperature of Injected Fluid (Conditional) 1393
Dynamic Grid Amalgamation Control (Optional) 1394
Custom Well Output (Optional) 1400
Terminate Simulation (Required) 1402

Appendix A **1403**

Theoretical Outline 1403
Well Model 1407
Figure A-1: Well Fraction and Geometrical Factor for Various Common Geometries 1409
Figure A-2: Multiblock Well Completion 1410
Wellbore Model 1411
Aquifer Model 1412
Table A-1: Built in Aquifer Influx Function 1414
Solubility Model 1415
Dual Porosity / Dual Permeability Model 1417

Appendix B **1421**

Figure B-1: Numbering of the Grid System 1421
Figure B-2: Cylindrical Reservoir 1422
Figure B-3: Calculation of Gravity Components for a Tilted Grid System 1423
Figure B-4: Radial (Cylindrical) Coordinates 1424
Figure B-5a: 2-Dimensional Variable Thickness Grid System (8x1x3) 1425
Figure B-5b: 3-Dimensional Variable Thickness Grid System (8x3x2) 1426
Figure B-6: Discretized Reservoir with Faults 1427
Figure B-7: Relative Permeability Curves 1428
Figure B-8: Recycling Option in GEM 1429

Appendix C **1431**

References 1431

Appendix D **1435**

Coupling GEM with Surface Network Models 1435
General Considerations and Guidelines 1435
Preparation of the GEM Data Set 1437
Multiple Reservoirs 1440
Coupling with GAP 1441
Coupling with FORGAS 1451
GEM Results Accessible from RESOLVE Script 1460
Total Numbers 1460
Well Variables 1460
Group Variables 1461
Sector Variables 1462

Appendix E **1465**

Compositional Simulation for Sequestration of CO_2 and Other Greenhouse Gases in Saline Aquifers 1465
Thermodynamic Equilibrium 1466
H_2O Vaporization 1470
Gas and Aqueous Phase Properties 1472
Reaction Stoichiometry 1473
Chemical Equilibrium 1474
Mineral Dissolution and Precipitation Reactions 1475
Component Material Balance Equation 1477
Elimination of the Chemical Equilibrium Reaction Rates 1478
Solution Method 1480
References 1482

Appendix F **1485**

Thermal Option in GEM 1485

Appendix G **1491**

Ion Exchange in GEM 1491

Appendix H **1495**

Polymer Option in GEM 1495
Implementation in GEM 1497
Aqueous phase viscosity in presence of polymers 1497
Polymer adsorption 1504

Inaccessible pore volume.....1504
Permeability reduction.....1504
Residual adsorption level.....1505
Polymer degradation.....1505

Keyword Index **1507**

Introduction

Important Changes in GEM 2016.10

SIMULATOR CHANGES

Geochemistry Input Format Version 2 (*GEOCHEM_V2)

The geochemistry data input is simplified for a better user experience and functionality:

1. The stoichiometry of chemical equilibrium, dissolution and precipitation, and ion exchange reactions is now directly obtained from the reactions which are specified using new keywords *REACTION-CHEM, *REACTION-RATE or *REACTION-IEX. Input of keywords *STOICHIOMETRY, *STOICH-IEX is not required.
2. If necessary, the aqueous components are re-ordered by GEM to obtain canonical form of the stoichiometry.
3. The equilibrium rate annihilation (ERA) matrix is calculated by GEM internally allowing easy addition of new or deletion of existing reactions in a data set.
4. Polymer components (identified via *COMPNAME-POLYMER) and aqueous surfactant components (identified via *COMPNAME-SURFACTANT) out of the aqueous components, can now coexist with geochemistry reactions, but such components themselves should not be part of the reactions.
5. Ion-exchange components are re-ordered internally, if necessary, to make sure exchange reactions are correctly recognized.

See template data-sets gmghg027, gmghg028, gmghg029, gmplm003 and gmsmo097.

Enhancement in the Empirical Foam Model

The empirical foam option now offers two models via a new keyword

*FOAM-MODEL *INTREL | *MULTREL

While *INTREL is the pre-existing model based on interpolation of relative permeabilities, the new option, invoked with keyword *MULTREL uses dimensionless foam parameter to directly modify the gas relative permeability. For the new option, additional keyword *FMCOMP is used to designate a surfactant component that would generate foam, and an optional keyword *FMKRW may be used to specify a multiplication factor for water relative permeability k_{rw}.

A new factor is added to the calculation of foam parameter FM to capture the permeability dependence. It is optionally invoked by two new keywords *FMPERM1 and *FMPERM2.

See template data-set gmsmo094.

Relative Permeability Table-Based Endpoint Scaling

Table-based endpoint scaling is now available in GEM. The saturation and relative permeability end points can be specified via newly introduced keywords, e.g., *TSORW, *TKROCW etc. The block-based end point keywords are renamed as *BSORW, *BKROCW etc. The older keywords (*SORW etc.), though still supported, are now deprecated.

See template data-set gmsmo098.

Hysteresis Option for Oil Relative Permeability in Gas-Liquid Table

New keyword *HYSKROG allows specification of maximum possible imbibition residual oil saturation (gas-liquid table) for calculation of hysteresis in oil relative permeability curve. This value is used to evaluate the shape and path of all imbibition curves which leave the drainage curve at any saturation. Existing keyword *HYSKRO is now deprecated. In its place *HYSKROW is expected.

See template data-set gmsmo096.

Enhancement in Stone 1 Model for 3-phase Oil Relative Permeability Calculation

Keywords *SOMWAT, *SOMGAS, *SOMGASI, *SOMWATI allow the user to enter a table of residual oil saturation as a function of water/gas saturation to be used in the STONE 1 three phase oil relative- permeability model. Contributions from either drainage or imbibition or both are accounted for depending on whether hysteresis on k_{row}, or k_{rog} or both are specified.

See template data-sets gmsmo095 and gmsmo096.

Smoothening of End Point Relative Permeabilities

Smoothening of end-point relative permeability values between connate and critical saturations is now optionally available through power-law options *POWERQ and *POWERC for both for SWT and SGT/SLT tables. These options are provided through new keywords that follow *SWT and *SGT/*SLT keywords, i.e,

*SMOOTHEND {*LINEAR | *QUAD | *CUBIC | *QUADGEM | *POWERQ I *POWERC | *ON | *OFF}

See template data-set gmsmo078.

Flexibility in 3-Point Relative Permeability Scaling

A better flexibility is provided in the 3-point relative permeability scaling behavior. New keywords *3PTSCAL_KRW, *3PTSCAL_KROW, *3PTSCAL_KRG, and *3PTSCAL_KROG allow turning on of 3-point scaling for individual relative permeability curves.

See template data-set gmsmo090.

Saturation End Point Interpolation

A two-stage method for interpolation between two sets of rock-fluid curves is introduced which is activated by keyword *INTERP_ENDS. In this method first, the saturation (horizontal axis) endpoints are interpolated separately and applied to the two normalized curves. Then the curve values (vertical axes) are obtained at the required saturation.

See template data-set gmsmo099.

Oil-Water Capillary Number based Relative Permeability Interpolation

The new keyword *INTCOMP *CAPNOW facilitates interpolation of relative permeability curves based on oil-water capillary number. A user specified table of interfacial tension versus a key component mole fraction in aqueous phase (*IFTTABLE) is used to obtain the interfacial tension used in the capillary number calculation.

New keywords *INTLOG and *INTLIN can be used for logarithmic or linear table-lookup of interfacial tension table.

See template data-sets gmsmo091, gmsmo092, and gmsmo093.

Aqueous Phase Salinity Calculation

A new keyword *COMPNAME-SAL is introduced to specify salinity components from out of the aqueous components in the model. The composition of salinity components in aqueous phase is then used to calculate salinity in molality (mol / kg H_2O). In absence of this keyword salinity calculation with NaCl, Na^{++}, or Cl^- ions is still done as before.

Writing of Restart based on Elapsed Times

The new keyword *WRST *ELAPSED-HRS *eltime* allows restart to be written after passage of specified amount of elapsed time in the simulation.

Stopping the Simulation after a Specified Time

Use new keyword *MAXCLK to stop the run after a specified time. It is similar in functionality as the existing keyword *MAXCPU, but applies to elapsed clock-time and may be useful for models being run with parallel processing options.

OUTPUT ENHANCEMENTS

Aqueous Phase Molar Density

Added output of water molar density via sub-keyword *RHOW to Simulation Results and out files through *OUTSRF *GRID; *OUTSRF *SPECIAL; and *OUTPRN *GRID.

Oil-Water and Gas-Water Capillary Number and Interfacial Tension

Output of oil-water and gas-water capillary number and interfacial tension to Simulation Results and out files is available via new sub-keywords *CAPNOW, *CAPNGW, *SIGMAOW, and *SIGMAGW through *OUTSRF *GRID; *OUTSRF *SPECIAL; and *OUTPRN *GRID.

Connection Based Flux Output

Connection based oil, gas, and water fluxes can now be written out to Simulation Results file using new keyword *FLUXCON. The information can then be accessed by ResultsReports using *CONNECTION-FOR keyword.

New FORMAT for Simulation Results Output

GEM now supports writing of Simulation Results output to a new hierarchical data format system called SR3. The SR3 file is a single Simulation Results file with suffix ".sr3" and is used by the new CMG product "Results" introduced in 2016.This file stores the same information as the IRF/MRF files but is optimized for speed of access by "Results" (see keywords *SRFORMAT and *FILENAME). For the 2016.10 release of GEM, the default is

to write out both SR3 as well as IMRF files (i.e., *SR3_IMRF). This may increase the amount of output and in some cases marginally impact the run time. If necessary, use IMRF or SR3 format (see *SRFORMAT or corresponding command line options) to optimize the output size and the run time.

Eventually SR3 format will replace IMRF format.

GEOMECHANICS CHANGES

Normal Effective Stress vs. Fracture/Matrix Permeability

A table-lookup procedure for fracture or matrix permeability multipliers versus normal effective stress is added. The multipliers are calculated due to change in geomechanical response. The data is input in a tabular form in x, y, and z directions. The first column corresponds either to normal effective stress or normal total stress in a given direction while the second column corresponds to the fracture permeability multiplier in that direction. A table-lookup is performed to get the permeability multiplier for a computed value of the stress.

The keywords for fracture permeability multipliers are: *GPERMNEX, *GPERMNTX, *GPERMNEY, *GPERMNTY, *GPERMNEZ, *GPERMNTZ.

The keywords for matrix permeability multipliers are: *GPERMNEX_MT, *GPERMNTX_MT, *GPERMNEY_MT, *GPERMNTY_MT, *GPERMNEZ_MT, *GPERMNTZ_MT.

See template data-sets gmgmc064, and gmgmc065.

Jointed Rock Mass Modeling

A jointed rock mass is a compound material including intact rocks (continuum) and jointed rocks (dis-continuum) containing numerous joints or small fractures in different directions. A method is now available wherein an equivalent constitutive model is constructed to include all joint characteristics, thereby mimicking solutions obtained by a discrete element method. It is currently available for elastic materials only; extension to elasto-plastic materials is planned for future.

See template data-sets gmgmc066-gmgmc068.

Deformation Magnification Control for Visualization

The deformation can be magnified by using the new keywords *MAGNIFYDI, *MAGNIFYDJ and *MAGNIFYDK in the corresponding directions. These keywords will not affect the computation but only help to visualize a deformed reservoir more clearly; especially for any reservoir undergoing very small deformation. Without using these keywords, the deformation is not magnified.

Extending Geomechanics Grid

The extending grid is designed so that additional grid blocks could be added automatically to the geomechanics grid in either of the I, J, or K directions. The option can be used in geomechanics models with or without the *GEOGRID.

See template data-sets gmgmc052, gmgmc053, and gmgmc059.

GRID CHANGES

Hydraulic Fracture Data Input Simplification

A number of changes in grid-input are introduced to simplify input of hydraulic fracture data, through use of *BLOCKGROUP, *SBLOCKGROUP *PLNRFRAC_TEMPLATE, *PLANRFRAC, and other new keywords.

User-Specified Gridblock Pore Volume

By default, the cell pore volume is calculated internally from the available porosity, net-to-gross ratio, and volume modifier. The user can now override the calculated pore volume of certain or all blocks with specified values directly through new keyword *PORVOL.

Direct Input of Transmissibility Values

New array keywords *TRANVI, *TRANVJ and *TRANVK now make it possible to directly input transmissibilities in I, J, and K directions, respectively.

Pore Volume Cutoff for Fracture Gridblocks

New keyword *PVCUTFR allows you to specify pore-volume cut-off value for fracture gridblocks. Note that existing keyword *PVCUTOFF applies the pore-volume cut-off to the matrix gridblocks only.

Controlling Inter-block Matrix-Fracture Connectivity

New keyword *NOINTERMF allows the user to skip the inter-block matrix-fracture connections when a fracture block is null but its collocated matrix block is non-null in a dual-continua model.

See template data-sets gmfrr010, and gmfrr011.

Pinch-out Array and Pinch-out Tolerance

Keywords *PNCHOUTARAAY and *PINCHOUT-TOL can now be used together in a data file. A grid-block satisfying either of the above two criteria will be pinched-out by the simulator.

WELL MANAGEMENT CHANGES

*BHPDEFAULT keyword

For wells that do not have an explicit definition of the bottom-hole pressure constraint, the default values can now be assigned by the user by a global keyword at the beginning of a run (*BHPDEFAULT *PROD | *INJE *val*). This keyword does not affect wells that already have the BHP constraint defined through keyword *OPERATE.

Individual Well Completion Layer Control

The new keyword *LAYER-CTRL allows modification of layer status and / or well indices for individual well completions without having to redefine the entire well. (*LAYER-CTRL *wn* action, where action = *OPEN | *CLOSE | *WIMULT *val* | *WIMULTO *val*).

Allow *AUTOLAYER in Injector's *BACKFLOW Monitor Constraint

Now *AUTOLAYER action is permitted in injector's *BACKFLOW monitoring constraint.

New Keyword *RECYCLE_GEM

A new keyword, *RECYCLE_GEM, is introduced as an alias to the GEM-only (primary) keyword *RECYCLE. The latter was conflicting with the general sub-keyword of *GCONI. Primary keyword *RECYCLE is still supported but is deprecated.

TEMPLATE DATASET CHANGES

The following Table lists new or significantly modified template files.

GMGHG027	Use of *GEOCHEM_V2 for Geochemistry Input Data.
GMGHG028	Polymer Injection in a Model with Geochemistry.
GMGHG029	Empirical Foam Model with Geochemistry
GMFRR010	Default Inter-block Matrix-Fracture Connections when Fracture block is Null and Collocated Matrix Block is Non-Null
GMFRR011	Use of *NOINTERMF to override default Inter-block Matrix-Fracture Connections when Fracture Block is Null and Collocated Matrix Block is Non-null
GMSMO078	Demonstrate *POWERQ and *POWERC in smoothening of Relative Permeability Endpoints.
GMSMO090	Use of three-point Endpoint Scaling keywords for Individual Relative Permeability curves.
GMSMO091	Surfactant Injection in Aqueous Phase; o-w Capillary Number based Relative Permeability Interpolation. Logarithmic Interpolation of IFT table.
GMSMO092	Surfactant Injection in Aqueous Phase; o-w Capillary Number based Relative Permeability Interpolation. Linear interpolation of IFT table.
GMSMO093	Companion dataset to GMSMO091; No surfactant injected in the aqueous phase.
GMSMO094	Empirical Foam Model 2 (*MULTREL) – Keyword Usage.
GMSMO095	Modify Oil Relative Permeability for Stone-1 using SOMGAS and SOMGASI tables, in Presence of Hysteresis in Oil Relative Permeability from Oil-Water table.
GMSMO096	Modify Oil Relative Permeability for Stone-1 using SOMWAT and SOMWATI tables, in presence of Hysteresis in Oil Relative Permeability from Gas-Oil table.
GMSMO097	Use of *GEOCHEM_V2 for Ion-Exchange Reactions.
GMSMO098	Demonstrate use of Table-based Endpoint Scaling of Relative Permeability Curves.
GMSMO099	Demonstrate use of *INTERP_ENDS for a two staged interpolation of relative permeability.
GMSMO100	Combination of Table- and Block-based Relative Permeability End-Points in Interpolation.

GMPLM003	Polymer Injection with Degradation Effects and Geochemistry
GMGMC052	Demonstrate use of Extending Geogrid feature.
GMGMC053	Demonstrate use of Extending/Amalgamating Geogrid feature
GMGMC059	Use of Extending and Amalgamating Grid – Output of Radial and Tangential Displacement and Effective Stress.
GMGMC064	Table lookup: Fracture Perm Multiplier vs. Normal Effective Stress.
GMGMC065	Table lookup: Matrix Perm Multiplier vs. Normal Effective Stress.
GMGMC066	Test/Illustrate Jointed-Rock Model *ROCKJOINT with Full Jointing.
GMGMC067	Test/Illustrate Jointed-Rock Model *ROCKJOINT with Partial Jointing.
GMGMC068	Rock Joints Used on geomechanical Grid.
GMGMC069	Output of Shear and Tensile Safety Factor in a Model with Amalgamating and Extending Grid.
GMWWM117	Demonstrate use of individual well-layer control actions.

MISCELLANEOUS CHANGES/BUG FIXES

Bug-Fixes

1. Fixed the use of keywords *BSWCON, *BSGCRIT, *BSWCRIT, *BSOIRW, *BSOIRG, *BSGCON, *BSLCON, *BKRWIRO, *BKRGCL, *BKROGCG defined via *PLNRFRAC_TEMPLATE and applied in recurrent section via *PLNRFRAC. Fixed restart runs associated with *PLNRFRAC keyword in the recurrent section.
2. Fixed a bug in distance calculation in *IRCONNECT.
3. A bug in output of $\log_{10}$(saturation index) is fixed. The simulation calculations are not affected. Fixed derivative calculation for log(activity coefficient).
4. Fixed a bug in injected aqueous component amount when keyword *INCOMP *AQUEOUS is used.
5. Fixed a bug in calculation of aquifer contribution in presence of chemical equilibrium reactions
6. Bug fix for group control to ensure a group production target will not be exceeded after being switched to other *MAX group constraints
7. Fixed a crash caused by simultaneous use of *TRANSF and *TRANSIJ+ etc.
8. Enhanced dimensioning message related to *ZDPEPTH entries in the initialization section.
9. Fixed an error in the printing banner of *SIGMAGW.
10. Fixed special history output of oil molar density (*RHOO).
11. Fixed parallel processing of large dynamic-gridding models (greater than 1 million grid blocks) while using 32 or more cores.

12. For relative permeability interpolation using *INTCOMP *AQUEOUS, output the value of interpolating parameter (*KRINTER) in molality, to be consistent with the input units.

Data Incompatibilities with Previous Versions of GEM

1. Restarts generated from previous versions are not compatible with this version of the code due to additional reading/writing of data.
2. Block-based saturation endpoint keywords *SWCON, *SGCRIT, *SORW, *SORG, *SWCRIT, *SOIRW, *SGCON, *SLCON and *SOIRG are deprecated. Use *BSWCON, *BSGCRIT, *BSORW, *BSORG, *BSWCRIT, *BSOIRW, *BSGCON, *BSLCON and *BSOIRG instead.
3. Block-based relative permeability endpoint keywords *KRWIRO, *KROCW, *KRGCL,*KROGCG *KRWRO, *KROCRW, *KRGRL, *KROGCRG are deprecated. Use *BKRWIRO, *BKROCW, *BKRGCL,*BKROGCG, *BKRWRO, *BKROCRW, *BKRGRL, *BKROGCRG instead.
4. Hysteresis option keyword *HYSKRO is deprecated, use *HYSROW in its place.

Important Changes in GEM 2015.10

SIMULATOR CHANGES

Nine-Point Discretization Option

Nine-point discretization option is now available in GEM. It allows computation of convective flows in diagonal directions to reduce the impact of grid orientation effects. The classic method of Yanosik and McCracken (1979) as modified by Shiralkar (1990), and Shiralkar and Stephenson (1991), is implemented. The option is activated by keyword *NINEPOINT (*IJ | *IK |*OFF). It comes with certain known restrictions – dual porosity, radial, refined grid, and dynamic grid models are not supported.

See template data sets gmsmo080 through gmsmo084.

Empirical Foam Modeling Option (Beta Feature)

An empirical foam option is now available in GEM. It allows interpolation between relative permeability sets based on foam characteristics like mobility reduction factor, capillary number effects, foam dry out etc.

In GEM at least one surfactant component needs to be defined (via *COMPNAME-SURFACTANT) from amongst the aqueous components. The surfactant component concentration is the primary interpolant that gets modified by mobility reduction factor and other inputs.

This option is being released as a Beta feature – no Builder support is available at this time.

See template data sets gmsmo085 and gmsmo086.

Simultaneous Use of Hysteresis and IFT in Miscible Models

Simultaneous use of hysteresis and interfacial tension in miscible flood models is now activated by specifying keyword *SIGMA and keywords *HYSRG or *HYSKRO. The maximum residual gas saturation for the miscible curve can be specified by keyword *HYSKRGM, whereas maximum possible imbibition residual oil saturation for the miscible curve can be specified by keyword *HYSKROM. In absence of*HYSKRGM, and *HYSKROM; *HYSKRG, and *HYSKRO values for the immiscible cure are used for the miscible curves as well.

See template data set gmsmo087.

Smoothening of End Point Relative Permeabilities

Smoothening of end-point relative permeability values between connate and critical saturations is now optionally available through a number of options both for SWT and SGT/SLT tables. These options are provided through new keywords that follow *SWT and *SGT/*SLT keywords, i.e.,

*SMOOTHEND {*LINEAR | *QUAD | *CUBIC | *QUADGEM | *ON | *OFF}

Prior to this only *LINEAR interpolation was available for K_{rw} and K_{row} and only *QUADGEM option was available for K_{rg} and K_{rog}. Now *QUADGEM is default method for all the four relative permeabilities. It prevents unexpected water saturation development in

certain datasets where relative permeabilities suddenly jump from zero to non-zero values. If desired, older behavior for K_{rw} and k_{row} can be obtained by using *SMOOTHEND *LINEAR.

See template data sets gmsmo078, and gmsmo079.

Change in Behavior of 3-Ppoint Relative Permeability Scaling

The 3-point relative permeability scaling behavior is made consistent with IMEX and STARS. Presence of individual scaling keywords like *KRWRO invokes 3-point scaling only for K_{rw} etc. *3PTSCALING keyword however invokes 3-point scaling for all the four relative permeability curves.

Enhancements in the *DEPTH_AVE Initialization Option

The *SWINIT keyword has been made compatible with the *DEPTH_AVE initialization option, previously it was only available for the *BLOCK_CENTER option. The *DEPTH_AVE implementation does not scale the oil water capillary pressure to maintain equilibrium; it adds the appropriate amount of capillary pressure to the block to maintain equilibrium.

This is akin to the simulator determining the correct entry capillary pressure for each block based on the desired block saturation (*SWINIT).

This has the benefit of allowing the option to function even if $P_{cow} = 0.0$ and usually results in capillary pressures which are less altered than by scaling (*BLOCK_CENTER).

The global mole fraction in *DEPTH_AVE initialization can now be entered via any of the *ZDEPTH, *ZGLOBAL or *ZGLOBALC in addition to existing input options *ZOIL and *ZGAS. This provides considerable flexibility in composition specification for the *DEPTH_AVE initialization option. Earlier *ZDEPTH option was only available for the *VERTICAL *COMP initialization.

See templates gmsmo074 through gmsmo077 for these enhancements.

Flux Sector Calculations and Output

Flux sector calculation and output reporting has been implemented in GEM. The purpose of flux sectors defined by *FLUX_SECTORNAMES and *FLUX_ISECTOR is to report net fluid flux (a) into each reservoir region and (b) between adjacent regions. Each flux sector is identified in reports by its name as well as its number. For computational purposes a flux boundary consists of a list of inter-block connections, where the two blocks of each connection are members of the two flux sectors. Each reported quantity is summed over all the individual inter-block connections that make up that boundary. Each flux boundary's report will appear as one column in the text output.

See sample data file gmsmo088 for keyword usage and evaluation of flux sector output in output file and in Simulation Results file.

Avoiding IFT Discontinuity during Phase Disappearance in Miscible Floods

A method is implemented to ensure smooth transition of interfacial tension (IFT) in IFT based relative permeability modification in miscible processes. A threshold value of interfacial tension is entered below which the two phases are considered to be miscible. The hydrocarbon phase compositions and phase densities are recorded when this occurs. These are later used to compute pseudo IFT when one of the phases vanishes. If the IFT thus calculated is above threshold the fluid is considered immiscible for relative permeability determination. On the other hand if the IFT is found to be below threshold, the fluid is

considered miscible and the miscible relative permeabilities are used. The grid blocks are continuously tracked through a tracking array.

See sample data file gsmsmo089 for usage of the above method during two-phase miscible to single phase transition.

Parallelization – Cutting Plane Direction Override

For models using *PPATTERN *AUTOPSLAB, a cut-plane direction override has been added via command-line option '-pslabdir i | j | k' which will force the cutting plane to be is the specified direction. This is useful for models where automatic determination of cutting plane via transmissibility in the perpendicular direction is not optimal. The cut-plane direction is printed to the output file along with the end-time run statistics.

Speedup of Models with Multiple Well Recompletions

Pre-scanning of well/perforations is implemented to reduce time in calculation of linear solver pointers especially in case of models with a large number of re-completions during the recurrent period. The option is invoked by command-line switch '-perfscan'. No additional keyword is required in the data file.

OTHER CHANGES

SORM Input Checking

The fatal error message is changed to a one-time warning for blocks where trapped oil saturation exceeds initial oil saturation. The trapped oil saturation is set equal to initial oil saturation for such blocks. Also checking of gas filled blocks is skipped and trapped oil saturation is set equal to zero in these blocks.

Checking Applicability of 9-point and Flash Method Inner

The data file is checked and user is cautioned against using the above features with unsupported options.

OUTPUT ENHANCEMENTS

EOS Region Map

For models with multiple EOS regions, the EOS region map is now available for visualization in RESULTS at time 0. No keyword is required for this output.

Polymer Option Control Parameters

Polymer option control parameters are now written to the output file at the start of the run. Echo of input data in a user friendly format is available via *OUTPRN *RES *ALL.

Output of Relative Permeability Interpolant

For models using relative permeability interpolation feature, output of the interpolant value to RESULTS via keyword *OUTSRF *GRID *KRINTER is added. It is also available to be output to out file via *OUTPRN *GRID *KRINTER and to the special history via *OUTSRF *SPECIAL *KRINTER 'block_id' for a specified grid block.

Separator Output

Functionality to optionally output separator stage liquid and vapor mole fractions for producers is added to the out file.

See modified template data set gmwwm026 that outputs separator stage liquid and vapor mole fractions for the production wells.

Status of Wells in Special History

Well-status for the field is made available as a special history with keyword *OUTSRF *SPECIAL *WELL-STATUS in the I/O section. It outputs total number of wells on production, wells on water injection, wells on gas injection, and shutin wells. See modified template data set gmpar003 that uses this new keyword.

Geochemistry Output

Output of activity and activity coefficient for aqueous components and log10 (saturation index) for mineral components is added for GHG models. See modified template data file gmghg006 that uses new keywords to output these items.

GRID CHANGES

Hydraulic Fracture Data Input Simplification

A number of changes in grid-input are introduced to simplify input of hydraulic fracture data, through use of *BLOCKGROUP, *PLNRFRAC_TEMPLATE, *PLANRFRAC, and other new keywords. See changes to the Reservoir Description chapter of the manual and new template data sets gmfrr008, and gmfrr009. A separate document is also available on hydraulic fracture keyword simplification in the ‘doc’ directory of the simulator installation area.

GEOMECHANICS CHANGES

Parallel Processing of Geomechanics Matrix Building and Residual Calculations

The geomechanics Jacobian matrix building and residual computation part of the code is now parallelized. It offers significant reduction in geomechanical calculation time in a parallel GEM runs. No additional keyword is required for taking advantage of parallel geomechanics processing, i.e., a GEM model with coupled geomechanics will perform matrix building and residual calculation for geomechanics in parallel, when usual parallel processing keywords are invoked.

Permeability Coupling

A new keyword *NOCOUPERM is added that allows all deformation-dependent permeability data to be ignored in a coupled flow and geomechanics model. It disables permeability coupling globally without the need to remove all the deformation-dependent permeability keyword data. This is useful when evaluating the effect of permeability variations when such data has been specified for a number of rock types. To disable permeability coupling only for selected rock types, the corresponding data must be removed for those rock types.

See modified template data set gmgmc006 that uses this keyword.

WELL MANGEMENT CHANGES

General Form of *PERF Keyword Reading

The *PERF keyword reading has been generalized to allow layer-wise input of all the parameters involved in Well Index calculations. The new layer table structure is very flexible and backward compatible.

See template data sets gmwwm112, and gmwwm113.

User Specified Limits on Well Indices

Auto well index thresholding is implemented to take action on out-of-range WIs. If WI exceeds the value specified by second value of the range, it is set to the value of the second range. For any WI less than first range it is either set to first range or set to zero depending on the qualifying sub keyword. For mean and median options, the range checking is done after multiplying the specified values to mean or median of layer well indices.

The range regulation is available on a well-wise basis and can be turned off in the Well and Recurrent Data section when necessary, via new keyword *WIRANGE.

See template data set gmwwm114.

Custom Output of WI and WI Statistics

A new keyword *WELPRN is now available for customized output of well indices and well index statistics for all or selected number of wells that can be turned on or off as desired. The statistics may be used as a guide for selecting parameters for *WIRANGE.

See template data set gmwwm114.

WHP Stability

A new WHP stability monitor constraint (*RATSTAB) has been introduced. This monitor is active in a region where the BHP vs. rate has a negative slope for producers or positive slope for injectors while WHP is held constant.

See template data set gmwwm112.

BHO, BHG and BHL Constraints for Producers

Bottomhole Oil (BHO), Bottomhole Gas (BHG) and Bottomhole Liquid (BHL) constraints are made available for production wells.

Hydraulic Radius for Wellbore Friction Calculations

A new sub-keyword *HYRADIUS of *HEAD-METHOD is added to introduce the hydraulic radius for the gravity-friction head calculations. The frictional part in the head calculations now uses the hydraulic radius which will override the default well radius from either *GEOMETRY or *PERF *RW.

See template data set gmwwm115.

Enhanced Hydraulic Tables for Wellhead Temperature Calculation

Keyword *PUTBE1 has been enhanced to read and process columns related to wellhead temperature. This along with new keyword *TWELLBORE allows computation of well head temperature for isothermal as well as non-isothermal models using hydraulic tables. See template data set gmwwm116.

TEMPLATE DATASET CHANGES

The following Table lists new or significantly modified template files.

GMSMO072	Template to illustrate 3 point vertical (rel perm) scaling
GMSMO073	Template to illustrate 3 point vertical (rel perm) scaling – consistency checks.
GMSMO074	Composition input options in DEPTH_AVE initialization; use of *ZOIL
GMSMO075	Composition input options in DEPTH_AVE initialization; use of *ZDEPTH
GMSMO076	Composition input options in DEPTH_AVE initialization; use of *ZGLOBALC
GMSMO077	Specifying initial saturation with *SWINIT in DEPTH_AVE initialization.
GMSMO078	Use of Relative Permeability End Point Smoothing.
GMSMO079	Linear Smoothing of Relative Permeability End Points.
GMSMO080	Grid Orientation Effect in Miscible Displacement – Parallel Grids, 5 point.
GMSMO081	Grid Orientation Effect in Miscible Displacement – Diagonal Grids, 5 point.
GMSMO082	Grid Orientation Effect in Miscible Displacement – Parallel Grids, 9 point.
GMSMO083	Grid Orientation Effect in Miscible Displacement – Diagonal Grids, 9 point.
GMSMO084	Grid Orientation Effect in Miscible Displacement – Use of keywords *TRANSIJ- and *TRANSINJ+ in 9-point discretization.
GMSMO085	Empirical Foam Model – Keywords Usage.
GMSMO086	Companion data set to GMSMO085 with Foam effects switched-off.
GMSMO087	Simultaneous Use of IFT dependent Relative Permeabilities and Hysteresis.
GMSMO088	Flux Sector Output – Keyword Usage
GMSMO089	Handling of two-phase miscible to single-phase transition for relative permeability calculations with IFT dependence.
GMPAR003	Use of keyword *OUTSRF *SPECIAL *WELL-STATUS
GMPLM001	Polymer slug injection in a light oil reservoir
GMPLM002	Water injection in a light oil reservoir. A companion dataset to gmplm001 for comparison.
GMGHG006	Output of activity, activity coefficient for aqueous components and log10(saturation index) of mineral components.
GMGMC006	Use of keyword *NOCOUPERM

GMFRR008	Use of *BLOCKGROUP for Hydraulic Fracture Input simplification.
GMFRR009	Use of *PLNRFRAC_TEMPLATE and PLNRFRAC for Hydraulic Fracture Input Simplification.
GMWWM026	Demonstrate use of keyword *WELSEP-ICPRT to output separator liquid and vapor stream compositions.
GMWWM112	Wellbore Instability Monitoring via *WHYSTAB and *RATSTAB.
GMWWM113	Demonstrate Use of the General Form of Perforation Input.
GMWWM114	Demonstrate Use of *WIRANGE and *WELPRN for Auto-Thresholding of Well Indices.
GMWWM115	Frictional Head Calculations with Hydraulic Radius.
GMWWM116	Use of enhanced hydraulic tables for well head temperature calculation.

MISCELLANEOUS CHANGES/BUG FIXES

Bug-Fixes

Fixed a bug related to a condition where ADSMAX specified is same as the value entered in adsorption table.

Fixed incorrect handling of residual adsorption level (*ADRT) handling in certain cases

Fixed dimensioning parameters for relative permeability graphics printed to the out file.

Fixed output format for grid-block number in *OUTSRF *RES *ALL.

Fixed parallel JACPR runs with keywords specified in the data file.

Data Incompatibilities with Previous Versions of GEM

The previous secondary keyword *KRPERF (had to immediately follow *PERF/*PERFV keyword) has now become a primary keyword for which an optional well list can be specified. This allows for easy redefinition of the relative permeability set numbers and/or end points without having to re-define the entire well completions.

Restarts generated from previous versions are not compatible with this version of the code due to additional reading/writing of data.

Important Changes in GEM 2014.10

SIMULATOR CHANGES

Polymer Option (Beta Feature)

Polymer flood option is now available in GEM. It allows

a) Change in aqueous phase viscosity in presence of polymer(s) using three optional aqueous phase viscosity models, i.e., two non-linear and one linear model.

b) Shear effects on aqueous phase viscosity – explicit as well as implicit.

c) Salinity effect on aqueous phase viscosity in presence of polymer(s)

d) Polymer adsorption on rock surface and its effect on mobility of fluids.

e) Effect of (in)-accessible pore volume and residual resistance factor.

f) Polymer degradation.

This option is being released as a Beta feature in version 2014.10 – no Builder support is available at this time. Keyword documentation and sample template data sets are in the Polymer_Option_in_GEM directory.

Adsorption of Aqueous Phase Components on Rock Surface

GEM now allows adsorption of aqueous phase components on the rock surface based on Langmuir isotherm or tabular adsorption input similar to the existing gas-phase adsorption option. In case of aqueous-phase adsorption however, mole fraction instead of partial pressure of the adsorbing component is used. Also unlike gas-phase adsorption, a residual adsorption level can be specified (via keyword *ADRT), accessible pore-volume (via keyword *PORFT) and residual resistance factor (via keyword *RRFT) for the aqueous phase adsorption. Note that aqueous- and gas-phase adsorptions are mutually exclusive.

See template data sets gmsmo062, gmsmo063, gmsmo064 and gmsmo069.

3-Point Scaling of Saturation and Relative Permeability

Three-point scaling of saturations and relative permeability values can now be invoked by keyword *3PTSCALING *ON in rock-fluid section. All grid blocks will then use 3 point saturation scaling instead of the default 2 point scaling. The keywords required for specifying saturation end points on a per grid block basis such as *SWCON, *SWCRIT etc. are available from earlier versions of GEM. If these keywords are not present in the data file, corresponding values are taken from the table entries. For vertical (k_r) scaling 4 new keywords *KRWRO, *KROCRW, *KRGRL, and *KROGCRG are introduced. A tutorial chapter “Two Point and Three Point Scaling of Relative Permeability Values” is available for ready reference.

See template data sets gmsmo070, gmsmo071, gmsmo072, and gmsmo073.

QNSS-TO-NR Switching for Improved Flash Convergence

For difficult (near critical) fluids, an approach is now available to optionally switch from Quasi Newton Successive Substitution iterations to Newton-Raphson iterations based on sum

of square residual error and number of QNSS iterations. See keyword *QNSS-TO-NR in Component Properties section, and template data set gmsmo068.

Optional output of flash convergence status for each grid-block is available through newly introduced sub-keywords *ITERVLE and *NBLKFAIL of *OUTSRF and *OUTPRN.

Solid Model Extension to Multiple EOS Sets

The solid model feature is now extended to multiple EOS sets. By default solid model parameters specified for the first EOS set will be inherited by subsequent EOS sets. Any of above parameters could then be modified in EOS set 2 and beyond to have a different solid model behavior.

A new keyword *NO-INHERIT-SOLID-EOS1 is introduced to prevent the default inheritance, if required. That means EOS set 1 may have a solid model while EOS set 2 may not, and EOS set 3 may have a completely different solid model than EOS 1 and so on.

See template data sets gmspr028, gmspr029, gmspr030, and gmspr031.

Geochemistry Enhancements

Equilibration of chemical equilibrium reactions and mineral reactions at the start of the simulation is now facilitated via optional keywords *CHEM-EQUIL-SET and *EQUIL-REACT-RATE respectively. This feature for

a) Chemical equilibrium reactions calculates a factor to the activity-product such that chemical equilibrium is honored when aqueous components molalities are input using keyword *MOLALITY-AQUEOUS in the initialization section;
b) Mineral reactions calculates a correction factor to the activity-product so that the initial reaction rates for all minerals are zero at time zero.

A new keyword *ACTIVCOEF-SOL is introduced that allows setting of activity coefficients of soluble components to be equal to salting-out coefficients.

A new keyword *ACTIVCOEF allows input of activity coefficients for all components in the aqueous phase.

See template data sets gmghg025, and gmghg026.

Ion Exchange

An empirical porosity based correlation is added to calculate Cation Exchange Capacity (CEC), via an optional keyword *CEC-CALC in Rock-Fluid section.

See template data set gmsmo061.

Stability and Run Time

A new convergence criterion based on convergence of block residuals via keyword *CONV-RESONLY is added to ensure that block residuals are fully converged before timestep is accepted as converged. See template data set gmghg025.

The adaptive implicit switching keyword *AIM is now permitted in both Numerical Methods Control and Recurrent and Well Data sections.

Parallelization

Improvement in computation of recurrent grid amalgamation, refinement, and inheritance calculations is made for models using adaptive gridding, leading to reduced run-time in these models. No additional input is required to invoke this option. At this point this is a GEM only feature.

New keyword *LBPRINT controls the optional printing of load-balancing status for parallel runs for non-solver tasks like Jacobian building, variable update, convergence checking etc. New keyword *LCSPRINT controls the optional printing of level and class structure for the parallel linear solver, PARASOL.

Aquifer Definition

A sub-keyword *RESBND is added to main keyword *AQUIFER that places the aquifer connections on the reservoir boundary defined as the first non-null block encountered when scanning in from the grid-boundary.

See template data set gmsmo065.

Other CHANGES

SORM Input Checking

Now block scaling along with existing table end-points are taken into account in checking of the input value for SORM.

BLOCKGROUP

New grid-array sub-keyword *BG allows assignment of per-cell data to block groups defined via new keyword *BLOCKGROUP in the Reservoir Description data.

Restart

Issue a fatal error message when an attempt is made to restart a run with parallel option *JACPAR *ON (-jacpar), from an RST file that was not generated using the same option. Similarly a fatal error message will be issued for a RST file generated using *JACPAR *ON (-jacpar), but being run in serial or in an incompatible parallelization mode.

JACPAR Parallelization Option

Disallow use of *JACPAR *ON (-jacpar) with partitioning options *PARTITION, *PPARTITION, *GPARTITION, and *APARTITION to prevent unexpected results.

Command-line Input

Ability to flag erroneous command-line arguments and reporting of unused command-line arguments is added.

Date Format

Dates are now written on to log files up to 10^7-1 years to accommodate very long duration simulations.

Output Enhancements

Well Stream Aqueous Phase Output

GEM now has ability to output component rates, and other properties like molar/mass density, viscosity etc. of the aqueous phase at reservoir as well as at surface conditions. These can be optionally output and/or plotted for the well-stream using a new sub-keyword *AQPHASEPROP of *OUTPRN *WELL and a series of other new sub-keywords of *OUTSRF *SPECIAL.

See template data set gmsmo060.

Equivalent Fraction in Ion Exchange

A new sub-keyword *EQVFRIEXN to the output (*OUTSRF *GRID, *OUTPRN *GRID) and special history (*OUTSRF *SPECIAL) is available for ion-exchange models. It is equivalent fraction of a specified component that is determined using maximum CEC over all the grid-blocks. See template gmsmo061.

Block-wise Total Gas In Place and its Sum over Layers

Block-wise total gas in place is available now in output and Simulation Results files, via *OUTSRF *GRID *TGIP and *OUTPRN *GRID *TGIP. See template data file gmsmo042.

Consistency in Reporting of In-Place Fluids

Reporting of in-place fluids for sectors is revised so that *PSPLIT output is consistent with fluids calculated during initialization.

GEOMECHANICS

Reaction Force: The reaction force specified by a new sub-keyword *REACFORCE of keyword *OUTSRF *GRID; *OUTSRF *SPECIAL or *GOUTSRF *GGRID, is the sum of all reaction forces acting on nodes of that element (host grid cell).

Safety Factor: Safety factor can be used as a measure of how close a material is to failure. This quantity is available at different locations and times, so it can be used to reveal trends in space and time. For example, at a specified time you may wish to find the minimum value of safety factor over the whole grid. Alternately, at a specified location you could observe the trend of safety factor over time to see if and how it approaches failure. Similar to *REACFORCE, the safety factor specified by sub-keyword *SAFACTOR corresponds to the stresses at the center of the element (host grid cell).

Young's Modulus after Failure: For Non-Linear Elastic Constitutive Model 2, two new keywords, viz., *FYOUNGAF and *POISSRATIO are introduced. *FYOUNGAF is the coefficient factor used to compute Young's Modulus after failure. *POISSRATIO forces Poisson's ratio to be fixed at this value instead of being recalculated. See sample data files gmgmc005, gmgmc007 and gmgmc010.

Cap Model: For Cap Model 1, a new keyword *GCAPMOD allows modification of Young's Modulus on the cap hardening surface. See sample data file gmgmc060.

New Option in Stress Dependent Properties: It is now possible to specify formation geomechanical properties that are dependent on minimum principal effective stress using new keyword *GRMPRSTAB. See sample data file gmgmc061.

Specification of per Grid-Block Data for Geomechanical Properties

The Young' modulus, Poisson's Ratio, Cohesion, Friction Angle, and Thermal Expansion Coefficient can now be input on a per grid-block basis using array keywords *YOUNGMAP, *POISSONMAP, *COHESIONMAP, *FRICANGMAP, and *THEXPMAP.

See template data sets gmgmc062, and gmgmc063.

WELL MANAGEMENT

Generalized Input of Well Geometry Parameters

New keyword *WELGEO provides an alternate way of setting well element geometry parameters for listed wells.

See template data set gmwwm111.

Thermal Hydraulic Tables

A sub-keyword *ENT (denoting enthalpy) is added to existing keywords *PWELLBORE and *IWELLBORE in the Recurrent and Well Data section to allow use of thermal hydraulic tables.

See template data set gmwwm108.

Injected Gas Composition Setting and Current Constraint

A new keyword *INCOMPWL is added for setting of injected lift-gas composition for producers or injected fluid composition for hydrocarbon (solvent) injectors. New keyword *WCURREN enables wells to run on their operating constraints without checking for constraint violation and possible switching.

See template data set gmwwm106.

Enhancement of Trigger Capability

Triggers are now available on separator oil and gas stream composition for *ON_GROUP specification (*MPLS,*MPVS,*MPWS). See template data set gmtrg017.

Simplification in Layer Lump Input Specification

Range is now allowed in specification of layer lumps. See template data set gmwwm110.

Wildcarding for Group Names

'*' Wildcard characters may now be used in the group names.

Keyword Default Change

*APPOR-METHOD – The default has changed to *INGUDE from *IP

Template Dataset Changes

The following Table lists new or significantly modified template files.

GMSMO060	Demonstrate output of well-stream quantities for aqueous phase.
GMSMO061	Demonstrate use of *CEC-CALC for computing CEC based on an internal correlation.
GMSMO062	Test adsorption of a component from aqueous phase – Langmuir isotherm.

GMSMO063	Test adsorption of a component from aqueous phase – Langmuir isotherm and residual adsorption level specification.
GMSMO064	Test adsorption of a component from aqueous phase – Tabular adsorption input and residual adsorption level specification.
GMSMO065	Demonstrate use of *AQUIFER *RESBND that attaches aquifer to first set of active blocks if the boundary blocks are null.
GMSMO066	Demonstrate use of *AQUIFER *RESBND that attaches aquifer to first set of active blocks if the boundary blocks are null; for comparison with the equivalent *AQUIFER *BOUNDARY model.
GMSMO067	Test data set using *AQUIFER *BOUNDARY for comparison with the results of gmsmo066 that uses *AQUIFER *RESBND.
GMSMO068	Template to demonstrate use of flash convergence keyword *QNSS-TO-NR and related output.
GMSMO069	Demonstrate effect of (In) Accessible pore volume (*PORFT) and Residual Resistance Factor (*RRFT) in a model with adsorption in aqueous phase.
GMSMO070	Template to illustrate 3 point scaling and contrast it with 2 point horizontal scaling option.
GMSMO071	Template to illustrate 2 point scaling and contrast it with 3 point horizontal scaling option.
GMSMO072	Template to illustrate 3 point vertical (rel perm) scaling.
GMSMO073	Template to illustrate 3 point vertical (rel perm) scaling – consistency checks.
GMGMC060	Demonstrate use of keyword *GCAPMOD in a triaxial test where Young's modulus changes in the hardening cap section.
GMGMC061	Illustrate use of minimum principal effective stress (*GRMPRSTAB) to compute geomechanical properties.
GMGMC062	Application of per-cell data via keywords *YOUNGMAP, *POISSONMAP and *FRICANGMAP.
GMGMC063	Illustrate/Verify *YOUNGMAP – Distributed Young's Modulus.
GMSPR028	Multi-EOS asphaltene model. Demonstrate default inheritance of solid-model from the first EOS set.
GMSPR029	Multi-EOS asphaltene model. Use of keyword *INHERIT-SOLID-OFF to disallow inheritance of solid model from first EOS set to subsequent EOS sets.
GMSPR030	Multi-EOS asphaltene model. Asphaltene model with two different solid-model parameters in two EOS sets and no solid-model in the third EOS set.
GMSPR031	Multi-EOS asphaltene model. Default inheritance (from first EOS set) and partial modification of solid-model parameters in second and subsequent EOS sets.
GMSPR032	Use of keywords *BLOCKGROUP and *BG in a hydro-fractured shale reservoir.
GMTRG017	Demonstrate use of *MPVS *MPLS *MPWS for trigger *ON_GROUP.

GMWWM106 Use of keywords *INCOMPWL and *WCURRCN, to reset hydrocarbon injector fluid composition; and application of primary well constraint at all times without the usual switching to the most restrictive constraint. With *WCURRCN The well is shutin if primary constraint is not met.

GMWWM107 Use of *PWELLBORE *MODEL for wellbore hydraulics. For comparison with results of gmwwm108 that uses *PWELLBORE *TABLE with enthalpy as fourth variable.

GMWWM108 The wellbore correlation is converted to thermal PTUBE1 with enthalpy as the 4th variable.

GMWWM109 Group target apportionment with *INGUIDE, with potentials at different BHP.

GMWWM110 Input simplification for LAYER LUMP specification.

GMWWM111 Demonstrate use of *WELGEO keyword for specifying well geometry parameters

MISCELLANEOUS CHANGES

Radial Grids

Coordinate representation for radial grid including tilted radial grid was fixed.

Outputs

An error in Gas compressibility output to Simulation Results files was fixed

An error in unit of Molecular Weight written to Simulation Results file was fixed.

Change in Keywords / Simulator Behavior

Keyword *DENROCK that earlier existed in Component Properties section is no longer supported. For rock density *ROCKDEN (a grid block based array) present in rock-fluid section will henceforth be used for all the calculations involving rock density.

Keywords *NONDARCY and *KLINKENBERG that were deprecated in Component Properties Section since 2009 are now removed from Component Properties section. These will continue to be supported in the Rock-Fluid section.

Keyword *PERM-VS-POR is re-named as *RFCALC. Data files with *PERM-VS-POR will continue to run, however, keyword *PERM-VS-POR is deprecated and will be removed in future versions of GEM.

Stack-Size on win_x64

Stack size for win_x64 executable was increased to ~5 MB.

Data Incompatibilities with Previous Versions of GEM

Restarts generated from previous versions are not compatible with this version of the code due to additional reading/writing of data.

Important Changes in GEM 2013.10

SIMULATOR CHANGES

Ion-Exchange Modeling

The geochemical modeling capabilities of GEM have been enhanced to include simulation of ion-exchange mechanism which involves replacement of one ion by another at the solid surface. The implementation allows modeling of low-salinity water flooding in GEM.

See templates gmsmo056.dat, and gmsmo057.dat.

CBM Modeling

The CBM modeling capability has been enhanced by

a) Adding a new priority ranking criterion based on production-potentials for queuing up the wells to an automatic drilling schedule. See templates gmsmo058, gmsmo059,

b) Allowing the adsorption input to be modified by a per-gridblock based multiplier to account for ash content, quality of the coal etc. See template gmsmo055, and,

c) Simplifying the relative permeability input for CBM models by not requiring the input of the oil phase relative permeabilities. See template gmsmo054.

Hydrocarbon Viscosity Defaulting

The system for defaulting viscosity parameters is now the same as for other EOS data: the first EOS set uses internally stored defaults or calculated values for any data that is not specified.

Dynamic Dimensioning of Reaction Related Arrays

The geochemical reaction related arrays in GEM are now dynamically dimensioned to use storage more efficiently. It also removes the limit on number of reactions to be used for chemical equilibrium and mineral reactions, that was earlier restricted to 50.

Temperature Dependent Volume Shifts

GEM now supports the use of temperature dependent volume shifts that can be activated by input of linear temperature-dependent volume correction for each component. See template gmsmo053.

Use of Compaction Tables in Recurrent Data

The compaction table number (*CTYPE) can now be redefined in recurrent data with certain restrictions. See templates gmsmo051 and gmsmo052.

Other New Keywords

New Keywords Related to Parallel Processing

New keywords are added that enable alignment of linear solver arrays during parallel processing to improve efficiency (*AI_ALIGN, *FI_ALIGN and *CACHE_ALIGN). *AI_ALIGN is automatically turned on for runs undergoing frequent change in implicitness pattern. OpenMP scheduling for such runs is also adjusted, unless optionally set by the user.

A new sub-keyword of *PPATTERN, called *AUTOP2D has been added to allow partitioning of linear solver domains in two dimensions in a user-friendly manner. See template gmpar007.

Flexibility in Restarting a GEM Run

Two new options (*RESTIME and *RESDATE) have been added to make restarting of GEM run more user-friendly.

Simulator Performance Indicators

Simulator performance indicators like, elapsed time, elapsed time rate, cumulative time steps, cumulative Newton cycles, cumulative solver iterations, cumulative timestep cuts, Newton cycles per timestep, solver iterations per Newton cycle, material balance error, average implicitness, and average memory usage for each time step can now be written to Simulation Results file as a special history for comparison between two or more runs. New sub-keyword *SIMPERF of *OUTSRF *SPECIAL allows output of all the above simulator performance indicators to be written to the special history. It is also possible to output them individually through new sub-keywords *ELTSCUM, *ELTSRATE, *TSTEPCUM, *NCYCCUM, *SOLITCUM, *TSCUTCUM, *NCYCPTS, *SOLITPN, *MBERROR, *AVGIMPL, or *MEMUSAGE.

Output Enhancements

*OUTSRF and *OUTPRN sub-keywords *OILH, *GASH *WATH and *FLUIDH have been added to facilitate output of oil, water, gas and total fluid enthalpies to the Simulation Results/output files.

*OUTSRF *SPECIAL sub-keywords *WSTRMOLDEN and *WSTRMASDEN have been added to facilitate output of well-stream molar and mass densities. New sub-keyword *TAVG of *OUTSRF *SPECIAL allows output of average reservoir temperature to Simulation Results file.

GEOMECHANICS CHANGES

New keyword *STRESS_RETURN controls the method used to compute stress return to the yield surface. Previously the method was always *SUBSTEP, but new sub-keyword *GCPA allows you to specify the Cutting Plane Algorithm appoach for Mohr-Coulomb and Drucker-Prager models. See templates gmgmc054 and gmgmc055.

New keyword *GMONETONE specifies one-to-one mapping between the host grid and the geomechanics grid. This option allows for faster mapping between the two grids, especially when the host grid has local grid refinement. See template gmgmc056.

New keywords *GRSTRESTAB and *GRPRESTAB specify formation geomechanical properties that depend on effective mean stress and reservoir pressure respectively. See templates gmgmc057, and gmgmc058.

New keywords *GAMALI, *GAMALJ and *GAMALK define a geomechanics grid that is an amalgamated copy of the host simulator grid. See template gmgmc059.

The following new *GOUTSRF *GGRID sub-keywords let you plot radial and tangential stresses and displacements for 3D radial grid. These sub-keywords are available also with *OUTSRF *GRID. See template gmgmc059.

STRESRAD:	Radial total stress
STRESTAN:	Tangential total stress
ESTRESRAD:	Radial effective stress
ESTRESTAN:	Tangential effective stress
DISPLRAD:	Radial displacement
DISPLTAN:	Tangential displacement

GRID PROCESSING ENHANCEMENTS

The grid-processing changes include,

a) Parallelization of grid-module connection building task for large models with corner-point grids. New keyword *GRIDPAR can optionally specify the number of threads to be used for parallel grid-processing. For runs using parallel processing for Jacobian building and/or for linear solver the number threads for grid-processing are inherited via existing keywords and use of *GRIDPAR is not required.

b) New keyword *TRANSMULT allows specification of behavior of transmissibility multiplier keywords to be accumulative or replaceable.

c) New keyword *CRNCON_NODE allows the corner-point grid to be converted into a node-based grid for internal representation within the simulator. This option has been found to speed-up grid processing tasks especially for large models, at the expense of some increase in storage requirements.

d) New keyword *INTER_REGION_TM specifies the inter region transmissibility multiplier between two adjacent regions defined by the new keyword *TRANS_MULT_REGION keyword. This option can be used to reduce or eliminate flow between two regions.

WELL MANGEMENT CHANGES

Enhancement of Trigger Capability

A number of additional keywords are added to enhance the existing capability of triggers.

a) In addition to the instantaneous values of rates or production ratios, it is now possible to specify a moving average of a rate, GOR etc. over a specified time period using the keyword *AVRGTIME as a trigger condition. See template gmtrg016.

b) Ability to specify trigger conditions on lift gas for wells and/or groups. The lift gas rates, cumulative and production ratios such as GOR can be specified as a trigger condition. See template gmtrg015.

c) Ability to specify maximum and minimum in saturations, pressure or temperature for a sector. This adds to the previously available average saturation, pressure and temperature in a sector. See template gmtrg014.

Enhancement to AUTO DRILL Option

a) New keyword *DRILLT allows the user to enter a time delay between successive well automatic drilling operations. It applies to drilling priority specified via *DRILLQ keyword to the case where priority list is generated by a set of internal user selected algorithms. See template gmsmo058, gmwwm103.

b) Ability to specify the maximum number of wells with auto drill status that can be opened at one time. See template gmwwm103.

c) Ability to specify a drainage radius for each of the listed production wells. The drainage radius is used to calculate in place amounts for each well and surface stream. See template gmsmo058.

d) Ability to specify an alternate phase for ranking wells with auto drill status.

New Operating Constraint for Wells

New operating constraints (WCUTBACK) are added to allow the well run on a reduced rate target of the specified control phase whenever it violates the base value of GOR, GLR, WOR, WGR or PRS constraint. See template gmwwm102.

Operating Constraints in History Match Mode

A new special history-matching mode has been added to allow treatment of (all) observed surface phase rates as a history matching constraint. The default is to convert the specified surface rate constraints into a total reservoir fluid rate. This control mode is useful for making the well produce/inject the correct amount of voidage from/to the reservoir or sectors before the mobility ratios are fully matched. See keywords *OPERATE-HIST and ALTER-HIST and template gmwwm101.

New Keywords *WTINCR, *GCONPINCR, *GCONIINCR

*WTINCR allows the modification of any previously specified well constraint by applying an incremental value to that constraint. *GCONPINCR and *GCONIINCR allow the same incremental modification for production and injection groups.

Template Dataset Changes

The following Table lists new or significantly modified template files.

GMSMO051	Demonstrate use of compaction table redefinition in recurrent section – LGR in Reservoir Description Section.
GMSMO052	Demonstrate use of compaction table redefinition in recurrent section – LGR in Recurrent Data Section.
GMSMO053	Demonstrate use of temperature-dependent volume shifts.
GMSMO054	CBM Modeling with simplified rel-perm input (via *KRCBM keyword).
GMSMO055	CBM Modeling – use of multiplying factor to adsorbed amounts calculated by Langmuir isotherm.
GMSMO056	Ion-Exchange Modelling. A low salinity water flood is simulated.
GMSMO057	A companion template to gmsmo056. Ion-exchange modeling is turned-off for comparison with results from gmsmo056.
GMSMO058	CBM reservoir development with AUTODRILL. User defined drainage radius of wells is used as an input to prioritize drilling of wells.
GMSMO059	CBM reservoir development with AUTODRILL. A companion template to gmsmo058. Uses instantaneous production potential to prioritize drilling of wells.

GMGMC054	Use of cutting plane algorithm (CPA) for plastic stress return.
GMGMC055	Use of sub-stepping approach for plastic stress return.
GMGMC056	Use of keyword *GMONETONE with overlapping corner-point grids.
GMGMC057	Use of *GRSTRESTAB (Stress-Dependent Mechanical Properties).
GMGMC058	Use of *GRPRESTAB (Pressure-Dependent Mechanical Properties).
GMGMC059	Use of amalgamating and contracting grid – output radial and tangential displacement; and effective stress.
GMTRG014	Use of maximum reservoir-pressure criterion to control voidage replacement factor.
GMTRG015	Use of gaslift rate; and gaslift+gas production rate as trigger conditions.
GMTRG016	Use of average time as trigger condition over which a moving average of a quantity is to be calculated.
GMWWM101	Use of *OPERATE-HIST/*ALTER-HIST keywords.
GMWWM102	Use of WCUTBACK constraint option.
GMWWM103	Prioritization of drilling queue by ranking algorithm based on product of oil rate and oil in place (*DRILLQ *IPPDRNINP).
GMWWM104	Companion template to gmwwm103. The wells are assigned priority on the drill queue based on the IPP (instantaneous production potential).
GMWWM105	Illustrate the ability to blend in a user-defined drilling priority list with the IPP (instantaneous production potential) based ranking algorithm.

OTHER IMPROVEMENTS

Mass Consistent Initialization

A bug was fixed in mass-consistent initialization for certain situations wherein oil could show up in gas-cap or free gas could show up in oil zone for models without capillary pressure

Restart

Fix in models where the parent run switched to using numerical derivatives.

Multiple Continuua Models

Fixed fatal error condition for MINC / subdomain that otherwise did not allow these models to run with TRANSI ... etc. in recurrent section.

GEM-Surface Network Link

Fixed material balance error in GEM- Surface network linked models where multiple wells were completed in same grid-block.

Ensured consistency in handling of cross-flow in generation of IPRs for GEM-surface network coupled runs.

GHG Models

An error in the reported amount of supercritical CO_2 was fixed.

Error messages related to GHG input in the initialization section were fixed.

Geomechanics

Simultaneous use of independent geomechanical grid and dynamic reservoir gridding can produce unexpected results, and is therefore now disallowed.

Fixed sign of mean effective stress in special history output.

Simultaneous Use of IFT and Hysteresis Options

A warning message is added to caution user on the simultaneous use of interfacial tension and hysteresis in a model.

Grid Processing

Fixed code to prevent failure to establish connection through pinch-out blocks in certain corner-point grid models.

Improvement in detection of overlapping refinement.

Interrupt Handling

Improved CTRL-C interrupt handling behavior of Aix platforms.

Data Incompatibilities with Previous Versions of GEM

Restarts generated from previous versions are not compatible with this version of the code due to additional reading/writing of data.

Important Changes in GEM 2012.10

SIMULATOR CHANGES

Relative Permeability Interpolation

With this feature, multiple relative permeability tables can be defined within one rock type, where each table corresponds to one value of a specified interpolant. Interpolants available are: Volume fraction of deposited asphaltene, volume fraction of mineral, concentration of aqueous species, mole fractions of components in hydrocarbon phases and temperature. Interpolation is linear by default, but may be weighted for non-linear behavior.

This feature may be used to model wettability change due to deposition of solids or aqueous composition change. See templates gmspr025.dat, gmspr026.dat and gmspr027.dat.

Viscosity Models by EOS Set

The hydrocarbon viscosity models have been generalized so that each EOS set can use a different type of viscosity correlation, as well as all input component viscosity data and correlation coefficients.

With this modification, all viscosity models will correctly handle variations in temperature, so there are no restrictions on viscosity model type for thermal models. See template data set gmsmo046.dat.

Varying Initialization Methods by Regions

Different initialization methods can be specified for different initialization regions for CBM models (e.g. user-input for matrix and depth-average vertical equilibrium for fractures).

See template data set gmsmo045.dat.

Other New Keywords

Critical Region Flash Skipping

The new keyword CRITDELD allows user control of the threshold density difference used for flash skipping. Prior to this release, GEM would skip the flash for a single phase block if the difference between the existing phase density and the incipient phase density was less than 25 kg/m3

Geomechanics Enhancements

The *GEOGRID option was enhanced to interface with a fluid-flow grid that has local grid refinement together with a natural-fracture grid option. See template data sets gmgmc049.dat and gmgmc050.dat.

The *GEOGRID option was enhanced to handle *GNULL blocks correctly. See template data set gmgmc051.dat.

Improved Initialization of Skewed Corner-point Grids

A very fine scale (optional) horizontal integration option has been introduced into the DEPTH_AVE integration option to be used with skewed corner-point grids. Sub-keyword *FINE_INTG enables this option for corner-point grids. See template data set gmsmo045.dat.

Output Enhancements

*OUTSRF and *OUTPRN subkeywords *ZALL, *YALL and *XALL have been added to facilitate output of all hydrocarbon component mole fractions to the Simulation Results /output files.

*OUTSRF *SPECIAL is now accepted as an alias for *OUTSRF *WELL. See template data set gmtrg011.dat.

Enhancements to Existing Keywords

Water Vaporization Improvements

The water vaporization model has been improved by: reformulating the model for 2-phase hydrocarbon system with water vaporization, reformulating the H2O component equilibrium equations, and improving the stability test for Aqueous-Liquid-Vapor systems.

Improved Adaptive Implicit Switching

The threshold-based AIM switching criterion has been extended to include aqueous components. This gives better performance for

Improved Upstream Weighting for Dispersion Control

The two-point upstream weighting method (*TWOPTFLUX) has been parallelized for fast run time, and enhanced to work without restriction with all of the GEM simulation options.

Shale Gas Initialization Speed-Up

For models with very large numbers of refinements (multi-level LGR) grid processing speed has been improved significantly, and property assignment code in GEM has been made more efficient, resulting in substantially reduced initialization times.

Improved Parallel Performance

The code for transmissibility updating in models using rock compaction tables has been parallelized, as well as the sector and material balance calculations and the velocity calculations.

An implicitness change criterion to decide load-balancing via keyword *JLOAD-CHIMP has been implemented.

Additional *MOD Keyword Feature (*ARRAY)

The *MOD *ARRAY option lets you replace values in a rectangular region of the grid, using an array of values. For example: *MOD *ARRAY value_array replaces the existing property values in the region by the corresponding values from value_array. The number of values in value_array must be (i2-i1+1) (j2-j1+1) (k2-k1+1) and repeat counts are allowed. A value must be specified for each block in the I-J-K range, even if the block is null. See template data set gmgro020.dat.

Template Dataset Changes

The following Table lists new or significantly modified template files.

GMSMO044	Velocity Dependent Relative Permeabilities Pope's model.
GMSMO045	Mixed initialization for CBM models.
GMSMO046	Usage of EOSSET-wise independent viscosity correlations.

GMSMO047	Demonstrate use of *FINE-INTG subkeyword in DEPTH-AVE initialization.
GMSMO048	Companion template to GMSMO047 for comparison.
GMSMO049	Test implicit switching based on composition threshold of aqueous components.
GMGMC049	Use of GEOGRID in a dual perm model with LGR in the reservoir grid.
GMGMC050	Use of a combination of dual grids LGR in reservoir grid and Barton-Bandis model.
GMGMC051	Use of *GNULL - Null Blocks in *GEOGRID.
GMGMC052	Geomech template demonstrating extending geogrid feature.
GMGMC053	Geomech template demonstrating extending/contacting geogrid feature.
GMGRO020	Use of *ARRAY qualifier to modify a property in a set of grid-blocks.
GMSPR025	Demonstrate usage of relative permeability interpolation keywords.
GMSPR026	Asphaltene modelling with solid deposition dependent rock-fluid data.
GMSPR027	Thermal + deposition + Rel Perm Interpolation.
GMTRG011	WAG process with cycle duration based on cum. injection/time targets.
GMTRG012	Use of special symbol '$' to link the value of the trigger test condition with a well action line.
GMTRG013	Template with triggers on reporting groups.
GMWWM097	Demonstrate usage of *SHUTLAYER1.
GMWWM098	Test BHF initialization with TURB/QUAD.

WELL MANAGEMENT CHANGES

Use of On-Time Averaged Rates in Layer Lump Calculations

New keyword *LAYERLUMP_ACTR selects actual or on-time averaged rates and ratios based on the rates in all calculations involving lumps. The alternative (default) is to use instantaneous rates in layer lump calculations.

Removing Targets from Group Control

The keywords *GCONP ‘group_name’ *TARGET *NONE will remove any production target specified previously (including *VREP, *RECYCLE and *PMAINT) for group ‘group_name’. Existing maximum / minimum constraints are not affected.

The keywords *GCONI ‘group_name’ *TARGET ‘stream_name’ *NONE will remove any previously specified injection target (including *VREP, *VREFP, *RECYCLE and *PMAINT) for the indicated ‘stream_name’ (gas or water) and group ‘group_name’. Existing maximum constraints are not affected.

Trigger on Reporting Group

The keyword *TRIGGER has a new subkeyword for trigger definition: *ON_REPGRP ‘reporting_group_name’ ‘reporting_group_condition’. This subkeyword indicates that the trigger test condition is to be applied to a reporting group ‘reporting_group_name’. The

reporting group must be previously defined to enable reporting group based triggers to be used. See the *TRIGGER documentation for the choices available for 'reporting_group_condition'.

New *MONITOR Capability

The keyword *MONITOR *SHUTLAYER1 will plug the most offending layer (which, when shut, causes the well to have the lowest/highest value of the monitored variable) and continue the simulation. *SHUTLAYER1 has the same effect as *SHUTLAYER for a rate type monitor (e.g. STO), but will be different for a ratio type monitor (e.g. GOR).

The keyword *MONITOR *AUTOLAYER1 is similar to *AUTOLAYER but corresponding to *SHUTLAYER1.

GRID CHANGES

Improved Corner-point Efficiency

New keywords *NNODES, *NODES, *NODEX, *NODEY, *NODEZ and *BLOCKS allow you to specify a fundamental corner-point grid using a node-based data structure.

Other New Keywords

A new irregular connection keyword (*IRCONNECT) has been added to the grid module. It is more flexible than the existing *SCONNECT keyword in that it is direction aware and so may be used with compaction and dilation models. It will use permeabilities in the blocks specified to be connected as well as input ½ lengths, input interblock area and input connection direction to calculate block transmissibility.

Enhancements to Existing Keywords

When used with keywords *DI, *DJ and *DK, sub-keyword *RG assigns non-uniform child block sizes by generating blocks that are proportioned to the specified sizes and fit into the parent block size. Previously, an error message was issued if the specified child block sizes did not sum exactly to the parent block size. Also, this new behaviour works for LGR in a corner-point grid whereas previously it worked only for Cartesian.

Data Incompatibilities with Previous Versions of GEM

Keywords *SECTOR, *SECTORARRAY and *SECTORNAMES may not be used together. Doing so in previous versions sometimes resulted in duplicated sectors which may or may not contain information.

*SECTORNAMES must appear at most once and must appear before *ISECTOR.

Important Changes in GEM 2011.10

SIMULATOR CHANGES

Improved Dimensioning and Memory Allocation

A new methodology for initial memory allocation and, if necessary, re-allocation of all of the major arrays in GEM has been implemented.

At the first timestep, arrays are allocated based on the current degree of implicitness, allowing GEM to minimize the memory usage. If any array size needs to be increased during the course of the simulation, a step increase in allocation is performed. Step sizes are calculated using the default or user-input expected maximum degree of implicitness, however, there is no limit to the number of times that array size may be increased, so GEM runs will not stop prematurely due to under-dimensioning of arrays.

One-Way Geomechanics Calculation by Post-Processing of Prior Simulator Response

One-way geomechanics calculations require pressure and temperature arrays from the reservoir simulator to compute the stress-strain field at a given timestep. Since there is no feedback involved in one-way calculations, the computation of pressure, temperature (and other variables) in the reservoir simulator and the stress-strain field in the geomechanics module can be done independently. A new workflow has been developed to allow single or multiple geomechanics screening runs to be done using the pressure and temperature arrays generated at different timesteps from a single reservoir simulation run. Using this workflow, the effect of varying mechanical properties on the stress-strain field can be tested without re-running the reservoir simulation.

The new keyword *GEOPOSTPRO in a GEM data set will indicate that flow calculations will not be performed, only the geomechanics module will be invoked. The pressure and temperature arrays can be read either from a restart file or from an Simulation Results file created by the original simulation run, as indicated by the *GEOPOSTINPUT *SR2 | *RST keyword.

Constraint Equation Residual Reduction

The new keyword *CER-REDUCTION *ON | *OFF enables or disables the Constraint Equation Residual reduction method. When this option is enabled, inner (Newton) iterations are performed to reduce to a specified tolerance the residuals for the following constraint equations: aqueous solubility, adsorption, solid precipitation, solid deposition, aqueous reaction equilibria and mineral rate reactions. Note that these iterations are separate from the Newton cycles performed during the timestep convergence, which can lead to significant time savings.

This method can be enabled or disabled at any time in the simulation, and the maximum number of iterations allowed can be specified with the *ITERCER keyword.

Other New Keywords

STARS-Like Endpoint Scaling Option (*SCALING-STARS)

The default scaling behavior in GEM when SWCON is specified is to consider SLCON in the gas-liquid table as the independent variable and re-calculate SOIRG.

The behavior in STARS when SWCON is specified is to consider SOIRG in the gas-liquid table as the independent variable and re-calculate SLCON.

In addition, STARS has the additional feature (which GEM did not have) to allow the user to alter both critical and connate (or irreducible) endpoint pairs by altering one of the pair, when in a relative permeability table those endpoints are identical (i.e. critical = connate).

Both of these differences have been addressed when using the *SCALING-STARS endpoint scaling option. Explicitly using the *SCALING-STARS keyword in any relative permeability table or defining the *SOIRG array keyword (standard GEM uses the *SLCON array) will turn on *SCALING-STARS and enable GEM to treat these two aspects of endpoint scaling as STARS does.

Output of Phase Compressibilities

New subkeywords for *OUTSRF *GRID and *OUTPRN *GRID have been added to allow output of oil, gas, water and total fluid compressibility. Z-Factors of oil and gas phases can also be output.

Geomechanics Enhancements

The Modified Cam Clay (MCC) constitutive model has been implemented in GEM to capture the behavior of soft geomaterials such as soil or sand. The model and associated subkeywords are described under the keyword *MCCMODEL.

New *OUTSRF *GRID subkeywords *STRESINVF, *STRESINVS and *VOIDRATIO are available with *GEOMECH.

Output of Over/Underburden Heat Loss Information

The subkeyword *OBHLOSS has been added to *OUTPRN *GRID and *OUTSRF *GRID to output the heat loss rate computed from the analytical heat loss model on a block-by-block basis. The subkeywords *OBHLOSSRATE and *OBHLOSSCUM have been added to *OUTSRF *WELL to output the heat loss rate and cumulative heat loss for the entire model.

Reduction of Simulation Results System Output File Size

The default for writing information to the Simulation Results system has been changed from double precision to single precision. This results in reduction of the MRF file size of 30 to 50%. Since GEM uses a separate restart file, restarts are not affected by this change. This behavior can be controlled with the *SR2PREC keyword.

Enhancements to Existing Keywords

Capillary Pressure Stabilized Initial Water Saturation Input (*SWINIT) Compatible With the *DEPTH_AVE Initialization Option

The *SWINIT option has been made compatible with the *DEPTH_AVE initialization option, previously it was only implemented for the *BLOCK_CENTER option. The *DEPTH_AVE implementation does not scale the oil water capillary pressure to maintain equilibrium; it adds the appropriate amount of capillary pressure to the block to maintain equilibrium.

This is akin to the simulator determining the correct entry capillary pressure for each block based on the desired block saturation (*SWINIT).

This has the benefit of allowing the option to function even if Pcow = 0.0 and usually results in capillary pressures which are less altered than by scaling (*BLOCK_CENTER).

Change in Default Numerical Control Settings

A number of default numerical control settings have been modified to improve the solver behavior, provide more useful minimum timestep sizes, prevent large restart files from being generated by default, and improve convergence. Details are given in the table below.

Description	Keyword	New Default	Old Default
Maximum number of GMRES iterations in solver	ITERMAX	80	20
Maximum number of GMRES steps taken in solver before resetting	NORTH	40	10
Minimum timestep size	DTMIN	1.0E-05 days	1.0E-02 days
Timestep size after well change	DTWELL	1.0E-02 days	(Equal to DTMIN)
Frequency of writing restart records	WRST	100,000	TIME (every DATE or TIME)
Frequency of initialization of BHP	WELLINIT	ITER (every Newtonian iteration)	CHANGE (after changes to operating conditions)

Use of Water Density Correlation at Initialization

Previous versions of GEM used the linear model for water density in the mass-consistent initialization. The new version will use whichever correlation is specified by the user in the Fluid Model section.

Thermal Model Enhancement

Treatment of rock heat capacity has been modified to be consistent with rock compressibility handling.

Template Dataset Changes

A large number of data sets were modified to remove dimensioning keywords, as these are no longer required due to changes in the dimensioning and memory allocation algorithms.

Also, due to the change in Implicit/Explicit map printing, *OUTPRN *GRID *IMPL has been added to many of the template datasets to maintain the same output obtained with previous versions of GEM.

The following Table lists new or significantly modified template files.

gmgmc037	An example of 2D radial grid with 2D geogrid of different dimensions.
gmgmc038	Generate simulator response for geo-post-processing in gmgmc039.
gmgmc039	One-way geomechanics with geo-post-processing of Simulation Results output from gmgmc038.
gmgmc040	Generate simulator response for geo-post-processing in gmgmc041.

gmgmc041	One-way geomechanics with geo-post-processing of restart from gmgmc040.
gmgmc042	Generate simulator response for geo-post-processing in gmgmc045.
gmgmc043	Generate simulator response for geo-post-processing in gmgmc045.
gmgmc044	Generate simulator response for geo-post-processing in gmgmc045.
gmgmc045	One-way geomechanics with geo-post-processing of Simulation Results output from gmgmc042, gmgmc043, and gmgmc044. Using parent seeking feature.
gmgmc046	One-way geomechanics with geo-post-processing of Simulation Results output from gmgmc044. Without using parent seeking feature.
gmgmc047	A combination of hardening cohesion and softening friction angle in a generalized plastic model under draining condition.
gmgmc048	Use of Modified Cam Clay model.
gmsmo042	Coalbed Methane using Palmer-Mansoori model and SETPI.
gmsmo043	Compressibility output in a variable salinity model
gmthr001	Modified for over/underburden heat loss and special history temperature output
gmtrg009	Use of OR logical connector in constructing a trigger condition.
gmtrg010	WAG Process Model with Group Cyclic Controls illustrating use of GTARGET
gmwwm095	Pressure maintenance with reference volume replacement (VREFP).
gmwwm096	Output of well PI's for plotting.

Data Incompatibilities with Previous Versions of GEM

Behavior of Implicit/Explicit map printing with *OUTPRN *GRID has changed. In previous versions of GEM, the Implicit/Explicit map was printed to the .out file whenever any array data was requested with *OUTPRN *GRID. Beginning with GEM 2011.10, *OUTPRN *GRID *IMPL must be specified to see this information. This is consistent with *OUTSRF *GRID.

For geochemistry related models, the default behavior is changed to always calculate initial molality of secondary aqueous components. This will ensure that the secondary aqueous component concentration is always in equilibrium with user input concentration of primary aqueous components. The original behavior, if required, can be restored by using the keyword, *CALC-MOLALITYAQ-SECONDARY *OFF in the Initial Conditions section.

WELL MANAGEMENT CHANGES

*FRAC Added to *WELL Keyword

Well fractions can now be defined per well and is useful for defining fractional (½, ¼, etc.) wells when simulating elements of symmetry. Rates and well indices for full well are input, *FRAC determines the fraction of the rate and index to use. See the *WELL keyword for more information.

*AND and *OR Conditions in Triggers

The trigger facility allows for conditional execution of certain user specified actions related to well and group control. Whereas previously only a single condition per trigger could be entered by the user, it is now possible for the user to specify multiple conditions for a single trigger. This essentially constitutes a compound logical statement with connecting logical operators AND/OR. The user also needs to assign a priority to each logical operator as this influences how the overall compound logical statement is evaluated. For more information refer to the explanation section of the *TRIGGER keyword in the well and recurrent data section of the user manual.

Voidage Replacement Injection Group

The new keyword *VRI_GROUP is used to specify the injection group from which injection rates will be taken to calculate a production recycling or voidage replacement target for a given group or groups. Please see keyword *VRI_GROUP for more information.

Voidage Replacement by Pattern Capability

A group named as a source for recycling or voidage replacement (in a keyword like *GPRODGROUP) can now be a reporting group.

Since reporting groups allow for fractional well membership, use of a reporting group as a contributing or nominating group for recycling or voidage replacement enables imposition of injection targets based on a pattern type flood (using a reporting group which represents an element of symmetry). For example, the source group could be made up of the four ¼ producing wells surrounding an injector.

The injection group defined using *VRI_GROUP may be a reporting group. This allows fractional well membership, allowing production targets based on pattern injection.

Output of Well Productivity Index

The keyword *OUTSRF *WELL *WELLPI can be used to specify that well productivity index information be sent to Results for plotting.

GRID CHANGES

*VAMOD and *VATYPE Keywords Added

The *VAMOD and *VATYPE keywords were enabled in the grid module to more easily create elements of symmetry. This feature can be used in conjunction with *WELL *FRAC (see above).

Improved Corner-point Efficiency

These changes were made to the Simulation Results file format to improve efficiency and capacity for large corner-point grids. Results and GEM can read both the new and previous Simulation Results file formats.

Increase binary record index from 32 bits to 64 bits, effectively removing the limitation of number of blocks with *GRID *CORNERS (previously 22 million).

Eliminate redundant corner-point coordinates for (1) most grids using *GRID *CORNER and (2) all grids generated by *WSRF *GRIDDEFORM. This saves space in the binary MRF file and reduces significantly the resources needed by Results to load the file.

Data Incompatibilities with Previous Versions of GEM

The behavior of *PVCUTOFF has changed for thermal runs. A cell's type is changed to "zero porosity" if its pore volume falls below *PVCUTOFF and it has a small non-zero porosity or net-to-gross ratio. Previously that cell's type was changed to null, which generally is not appropriate in a thermal context. Manual application of keyword *NULL can reproduce the previous resulting grid, but this is not recommended and is not practical for large grids.

Important Changes in GEM 2010.10

SIMULATOR CHANGES

Thermal Modelling in GEM

Beginning with version 2010.10, GEM supports use of a thermal option wherein the temperature equation is solved along with the other equations.

The model includes an energy balance with convection, conduction and heat loss terms. The convection terms include phase enthalpies. Aqueous phase enthalpy is determined from a steam table look-up while oil and gas phase enthalpies are determined from the equation of state. This allows modelling of non-ideal fluid effects and heating or cooling due to fluid expansion or compression, including the Joule-Thomson effect. Conduction is modelled with a term for thermal conductivity of the rock and fluids, and heat losses to overburden, underburden and edge boundaries are calculated with an analytical formulation.

Temperature of injected fluids may be specified with the tubing data format (*IWELLBORE) or simply by specifying the keyword INJ-TEMP. See the Recurrent Data section for details. The thermal model and thermal fluid properties are specified in the Component Properties section (see keywords *THERMAL, ENTHCOEFF and ENTHALPY-MIXING), and thermal rock properties are entered in the Reservoir Description section (see keywords CP-ROCK, THCONR0 and HEAT-LOSS).

Information on setting up a thermal GEM run may be found in the Tutorial section of this User's Guide. Also, the "tpl\thr" template folder includes a number of example data sets. Details of the equations used in the thermal model are given in Appendix F.

Independent Geomechanics Grid

It is now possible to specify a geomechanics grid that is independent of the host simulator's grid.

The geomechanics grid can be specified with much larger grid blocks than the fluid flow simulation grid, and can encompass as much of the surrounding rock as desired. This allows for much faster computation compared to the case where the geomechanics module uses the same grid as the fluid flow computation.

There are also options for translating and rotating the geomechanics grid to control the position relative to the host grid. Deformation controls may be applied to force the host grid to have no deformation on specified boundaries.

See the *GEOGRID keyword description in the Geomechanics section for more information.

Improved Parallel Performance

Improvements to the implementation of parallelization in GEM have allowed the Simulator to run more efficiently on larger number of threads on Shared Memory Processors.

Array initialization, well term calculation in Jacobian building, large integer arithmetic and memory allocation have all been enhanced to allow faster parallel runs of large problems.

Full Well Solve

A method has been implemented to minimize solver-pointer re-generation that treats shut in or wells on BHP constraint as solver wells. The work-saving is significant for models with a large number of well layers. See the *INACTIVE-WELL-SOLVE keyword in the Numerical Methods Control section for more details. The method is turned ON by default.

Entry of Permeability and Non Darcy Correction Factor in Recurrent Data

This option allows the input/redefinition of the permeability arrays *PERMI, *PERMJ and *PERMK and the non-Darcy correction factor *NDARCYCOR in recurrent data. This feature can be used for modelling hydraulic fracturing events in the simulator.

GEM-Surface Network Models

In previous versions of GEM, the IPR table passed to the surface network program was generated with equally spaced intervals. Beginning with GEM 2009.13 a logarithmic spacing is used. This is expected to be helpful in reducing oscillations for high-productivity and backflowing wells.

Resistance Factor Calculations

Resistance to flow due to solid deposition has now been generalized so that power-law or Kozeny-Carman relations with adjustable exponents can be used for permeability reduction calculations due to asphaltene or mineral deposition. See the *PERM-VS-POR keyword in the Component Properties section for more details.

WELL MANAGEMENT CHANGES

Completion Lumping Option

*LAYERCLUMP provides a way for the user to define a control lump - a set of layers (perforations, completions) which possesses a name and a set of computed rate, cumulative, and stream ratio values. To each control lump there is also an associated setting value.

*LAYERRLUMP provides a way for the user to lump well layers in a group called a report lump. For report lumps, stream rates, cumulatives and production rate ratios are calculated and the user can design triggers based on these quantities. A given layer can be assigned a partial membership to report lump and the same layer can belong to more than one group.

*RM-LAYERCLUMP removes membership of a layer from the named control lump.

*RM-LAYERRLUMP removes membership of a layer from the named report lump.

*CLUMPSETTING assigns a control setting or flow restriction parameter to the named control lumps. The flow restriction parameter is used as a multiplication factor in adjusting the value of the well index for all the layers belonging to a given control lump.

There is also a set of computed rate, cumulative, and stream ratio values for each control lump. Triggers may be defined using these quantities specific to named control lumps; actions can be any valid well data lines, including lines which alter the control lump setting values. The syntax of the control lump based trigger statement is:

*TRIGGER ‘trigger_name’ *ON_CTRLLUMP ‘control_lump_name’ quantity > value

The syntax of the report lump based trigger statement is:

*TRIGGER ‘trigger_name’ *ON_RPTLUMP ‘report_lump_name’ quantity > value

See *LAYERCLUMP keyword for a list of quantities which can be used in control and report lump triggers.

***WHYSTAB *AUTOWELL Monitor**

“*MONITOR WHYSTAB *AUTOWELL frequency” can be used to monitor production well wellbore hydraulics stability using the *AUTOWELL option, which shuts the well for a specified time.

***MXCNRPT Maximum Number of Continue Repeats**

Every well having *CONT *REPEAT (continue-repeat) specified through *OPERATE is entitled to one repeat of the current timestep if one of its operating constraints is violated. During convergence of a timestep, it is possible that different wells violate and then switch their operating constraints consecutively after each repeat. Although the maximum number of possible repeat cannot exceed the number of wells, it may lead to a significant increase of run time if there are many wells. Quantity nmxrpt limits the number of such repeats per timestep, and keyword *MXCNRPT lets the user override its default value (of 1).

Important Changes in GEM 2009.10

SIMULATOR CHANGES

Geomechanics Coupling

One- and two-way (explicit as well as iterative) coupling with CMG's Geomechanics module has been implemented into GEM and is being released as a Beta feature.

The geomechanics module handles different constitutive models like, linear and non-linear elastic, generalized plasticity, elasto-plastic, single surface, and creep models. For the elasto-plastic model the yield criteria can be specified via Mohr-Columb or Drucker-Prager that are suitable for the description of geologic material. There is also an isotropic strain hardening and softening option. The behavior of cyclic loading and unloading that occurs during cyclical injection and production can be modeled. A cap model is also available for yield criteria of elasto-plastic material. For Non-linear elasticity, hypoelastic and hyperelastic constitutive models are available. Associated or non-associated flow rules can be used for Mohr-Columb, Drucker-Prager, single surface, or generalized plasticity models.

The above constitutive models can be used in 2D or 3D problems in the Cartesian, corner-point, or axissymmetric grid. Finite element method is used in the geomechanics module to compute the geomechanical responses. The user can specify a variety of boundary conditions in 2D and 3D on nodal points of the finite element.

The geomechanics module solves for the force equilibrium of the formation and calculates volumetric dilation/compression as a result of both elastic and plastic straining.

The coupling options allow porosity to be optionally dependent on deformation. It can be a function of pressure temperature or volumetric strain; or a function of pressure temperature and total mean stress.

The change in matrix and fracture permeability due to geomechanical effects can be computed by a number of algorithms including the ones by Li and Chalaturnyk; and Barton-Bandis.

A large number of possible geomechanics responses are available for viewing in CMG's graphical post-processor "Results", and also available to the text ".out" and ".geo" files. The user can view grids in "Results" that deform with time, but there are some restrictions for displaying 2D radial grids.

A new data section GEOMECHANICS has been added which precedes WELLS AND RECURRENT DATA section in the GEM input data file to facilitate entry of geomechanics related keywords. Please see the chapter on geomechanical features for specific keywords.

The coupled geomechanic modeling may be useful for a variety of situations, like determination of leakage through points of weakness in the cap rock during GHG sequestration, effect of rock deformation and porosity change during CBM/ECBM simulation, solid deposition in asphaltene modeling, and mineral dissolution and precipitation during geochemical processes, and others.

Improved Parallel Performance

A new method has been implemented for parallel equation and residual building using a method of domain decomposition based on Parasol classes. This method is expected to give better scalability on computers with larger number of processors.

The method is activated by using *JACPAR *ON in the Numerical Methods section, or by specifying –jacpar on the command line. In order to use *JACPAR, the parallel computing licensing feature must be active.

Oil-Wet Rock-Fluid Model

The oil-wet rock fluid model modifies both the initialization and flow calculations in GEM. See the OILWET subkeyword of the RPT keyword for details.

The oil wet option initialization assumes that the reservoir was initially water wet and has undergone a wettability alteration to become an oil wet reservoir. This assumption has a significant impact on how fluids are placed during reservoir initialization.

In the oil wet model, three phase flow is calculated assuming water is the intermediate phase. Therefore K_{ro} is a function of oil saturation, K_{rg} is a function of gas saturation, while K_{rw} is a function of both oil and gas saturation (through K_{rwo} and K_{rwg}). The normal three phase flow models (e.g. Stone 1, Stone 2, segregated, etc.) are used to determine water relative permeability, not oil permeability.

Non-Darcy Flow Model Enhancements

Direct input of the parameters used in the calculation of the non-Darcy Beta factor can now be specified with the NONDARCY GENERAL keyword. A new array keyword NDARCYCOR has been implemented to allow input of a multiplier for the Beta parameter used in non-darcy flow calculations. These features are useful for explicit modelling of non-Darcy flow in fractures.

In previous releases, the non-Darcy model specification was done in the Fluid model section. Beginning with this release, the NONDARCY keyword will be read in the Rock-Fluid section along with the NDARCYCOR keyword. The old format for the NONDARCY keyword in the Fluid model section is obsolescent but will continue to be read for backward compatibility.

Constraint Equation Residual Convergence

The constraint equation residual checking has been implemented to reduce the sum of square residuals of constraint equations based on user specified tolerance and user specified maximum number of Newton iteration for which the checking is required. It can be invoked via *CONVERGE sub-keywords, *CEQVLE, *CEQAQU, *CEQADS, …, etc. The number of iterations for checking can be specified via, *NCHEK-CEQ.

The option is especially useful for GHG sequestration, CBM modeling, etc., where use of keywords, *ITERMIN or *DTMAX could be minimized.

Combined Stability and Threshold Criterion of Adaptive Implicit Switching

A combined stability (CFL) and threshold criterion has been implemented for adaptive implicit switching. While switching based on CFL is applied at the timestep level, the switching with saturation/composition threshold can be done at timestep on Newton iteration level. The option is available via *AIM *STAB *AND-THRESH in the NUMERICAL METHODS section, and is being released as a Beta feature.

Diary Output Redesign

The Gem log file has been redesigned to provide more useful and flexible information to the user:

- The average material balance error, and cumulative solver failures are printed at the end of each timestep for *CHANGES format.
- The width of user block address is dynamically expanded for large size models to accommodate i,j,k triplets for large models for the *CHANGES format.
- Convergence status flags can be optionally printed with changes format to detect convergence problems.
- Material balance error and status of wells is printed at the end of each timestep for *WELL-INFO format.
- The header is repeated after a page-full of information for easy identification of the printed information on the log file.

***PTUBE1/*ITUBE1 Table Generation in Builder**

The new PTUBE1/ITUBE1 well head pressure tables are supported by the well head pressure calculator program. This now includes generation of injection tables as well as production tables which incorporate gas lift.

GHG and Aqueous Phase Modelling Improvements

A number of improvements to the models for gas solubility and aqueous phase property modelling have been implemented. These included fixes to the *INCOMP *AQUEOUS feature, improvements to water vaporization modelling, allowing use of SORM with the OGW flash, improved reporting of aqueous and mineral component, improved salting-out calculations, and other bug fixes.

WELL MANAGEMENT CHANGES

***ITUBE1/*PTUBE1**

PTUBE has been replaced by the more flexible PTUBE1, in addition ITUBE1 allows for the definition of injection tables using CMG standard well head pressure input table format. *PTUBE tables are still supported.

***GCONI 'Stream' *PMAINT *PMMAXR d1 d2**

Introduces the maximum surface rate of the specified injection stream that is available for the pressure maintenance. It is the summation of two sources: a fraction (d1) of the total produced (from the same group) and a make-up rate (d2). The values must be non-negative real numbers. The fraction d1 is nondimensional and defaulted to 0. The make-up rate d2 is (m3/day | bbl/day | cm3/min) for water and (m3/day | scf/day | cm3/min) for gas or solvent, and is defaulted to 1.0e+20.

***GCONPMULT *RECYCLE**

This subkeyword specifies a recycling production target. This indicates that the production wells connected to this group produce such that the phase injected by the injection wells connected to this group as specified by *GAS, *WATER or *SOLVENT is reproduced (recycled) out of the reservoir.

*TARGET *WTG / *BHW

The *WTG subkeyword for surface wet gas rate and *BHW subkeyword for reservoir water rate are now available for setting well constraints with the *TARGET keyword.

*PERFRG keyword removed

This keyword is no longer required as the *PERF keyword can be used instead.

Important Changes in GEM 2008.10

SIMULATOR CHANGES

SUBDOMAIN, MINC and SUBDOMAIN-Dual Permeability-Reinfiltration

New features have been added to GEM for modelling dual-continua (matrix/fracture) processes:

The standard SUBDOMAIN model is an extension of the DUALPOR model which divides the matrix vertically to allow simulation of fluid segregation due to gravity within each matrix block.

The MINC (multiple-interacting-continua) model is an extension of the DUALPOR model which divides the matrix into multiple nested rings. This allows simulation of fluid transport from the center of the matrix block through each nested ring before reaching the fracture system.

The SUBDOMAIN model can be extended to include gravity drainage between subdomains via the TRANSD keyword. This allows fluid transport from the bottom of one subdomain into the top subdomain of the block below it without having to flow into the fracture first. This provides a hybrid of subdomain and dual-permeability modelling capabilities. This behavior can be further modified with the SD_REINF keyword, which will scale the capillary pressure in the bottom subdomain of a block and allow the flow to be modelled as a combination of gravity drainage and reinfiltration.

See the keywords *MINC, *SUBDOMAIN, *FRACVOL and *TRANSD in the Reservoir Description section and *SD_REINF in the Rock-Fluid section.

Templates gmfrr004 through gmfrr007 show example applications of these new features.

Hysteresis for WAG Floods

Special hysteresis options have been implemented for modelling important rock-fluid interactions occurring during Water Alternating Gas injection processes.

The WAG hysteresis functions may be applied to water and or gas phases. In this model, the relative permeability of a phase is dependent on its own saturation history, as well as the saturation history of the other phase. This allows modelling the decrease in injectivity observed with repeated WAG cycles. The Land and Carlson model is used for gas phase hysteresis, and a 3-phase relative permeability table is entered for water.

See the HYS_MODEL keyword in the Rock-Fluid section, and template file gmsmo034.dat.

GEM-GHG (Greenhouse Gas) Modelling

Prior to the 2008.10 release of GEM, a standalone special executable called GEM-GHG was available for modelling aqueous phase chemical reactions and mineral precipitation/dissolution. In this release the GHG calculation modules are available with the standard version of GEM. As a result, the options for aqueous property modelling in GEM have been enhanced substantially, even when GHG reactions are not being used, as described in the following sections. The models with GHG options can now be run using parallel processing feature of GEM which was not supported by the standalone GEM-GHG executable.

There are many new keywords associated with the GEM-GHG modules. In the Component Properties section, see the headings: GHG Option – Aqueous Components, GHG Option -

Mineral Component, GHG Option – Reactions, GHG Option – Reaction Equilibrium and Rate Parameters, GHG Option - Aqueous Phase Model and GHG Option – Factors for CO_2 Inventory.

In the Initial Conditions section, see the entry titled Initial Aqueous/Mineral Concentrations.

In Recurrent Data, the composition of a water injection can now be specified with the keywords *INCOMP *AQUEOUS.

In addition, new subkeywords of *OUTPRN and *OUTSRF are available in the I/O Control section, along with the new keyword *INVENTORY-CO2, to display the results of the GHG calculations.

Template data sets are available in the "tpl/ghg" folder.

Aqueous Phase Properties

Keywords DENWS and RHOWS have been implemented to allow specification of mass or molar water density for surface calculations. Previously, GEM did not accept user input for these quantities. Mass density will be converted to molar density for use in GEM. Molar density is used to convert between molar and volumetric rates for surface water injection and production calculations.

Aqueous density can now be calculated from the Rowe and Chou correlation, which is a function of pressure, temperature and salinity, in addition to the usual linear model. See the keyword *AQUEOUS-DENSITY in the Component Properties section.

Aqueous viscosity can now be calculated as a function of pressure, temperature and salinity using the Kestin correlation, activated with the *AQUEOUS-VISCOSITY keyword.

A constant brine salinity can be entered with the *SALINITY keyword. More detailed modelling of aqueous ion concentrations is available with the Greenhouse Gas (GEM-GHG) options.

An example is given in template file gmsmo037.

Solubility of Gases in the Aqueous Phase

Accurate models for the Henry's constants of CO_2, N_2, H_2S and CH_4 have been implemented, taking into account pressure, temperature and salinity (salting-out coefficient). These models are activated with the keywords *HENRY-CORR-CO2, *HENRY-CORR-N2, *HENRY-CORR-H2S and *HENRY-CORR-C1. The existing aqueous phase solubility models are still available in GEM.

Water Vaporization

Vaporization of water may be modelled using the *OGW_FLASH keyword. In this case, H2O must be included as a component in the EOS set, as indicated by the H2O_INCLUDED keyword. These keywords are described in the Component Properties section.

Saline Water Injection

Gem version 2008.10 allows injection of aqueous components into the reservoir through use of keyword *INCOMP *AQUEOUS in the Well and Recurrent data section provided the model is set up to handle aqueous components. The definition of aqueous components is done in the Component Properties section and the specification of concentration of aqueous components is done in the Initialization section. Please refer to keywords *NC-AQUEOUS, *COMPNAME-

AQUEOUS etc. in the Component Properties section and keyword *MOLALITY-AQUEOUS etc. in the Initialization section.

There is a description in the Tutorial section for Saline Water injection.

Temperature-Dependent Asphaltene Precipitation

The asphaltene precipitation model has been enhanced to include temperature-dependent terms. This allows the asphaltene onset pressure to vary with temperature, and the amount of asphaltene precipitate may be a function of temperature, pressure and composition. WinProp version 2008.10 may be used to generate the temperature-dependent parameters, given one or more onset pressures at temperatures different than the reference temperature. See Component Properties section keyword *SOLIDTHERMAL and *SOLID_TREF.

An example is shown in template file gmspr018.dat.

Single-Phase Identification

Component Properties keyword *PHASEID has a new subkeyword *TCMIX, indicating that single phase identities will be determined by comparing the block temperature to the pseudo-critical temperature of the mixture calculated using the Li critical temperature mixing rule (Li, 1971): If block temperature is greater than or equal to the pseudo-critical temperature, the phase is identified as a gas, otherwise it is identified as an oil.

In the Initial Conditions section, the keyword TCMULT can be used to “tune” the critical temperature correlation on a region by region basis.

In the Rock-Fluid section, the keyword TRLIMS can be used to introduce reduced temperature limits which will be used together with the pseudo-critical temperature of a single hydrocarbon phase to generate a relative permeability interpolating function. This function is used to calculate a water-hydrocarbon relative permeability by interpolating between the water-oil and gas-liquid relative permeability curves.

An example is shown in template file gmspr019.dat.

Modelling of Cap Rock Leakage via Special Well Controls

A method has been implemented in GEM to model the flow of fluids, primarily gas, from the reservoir through the cap rock once the pressure difference across the seal exceeds a specified threshold value. This cap rock leakage occurs through wells defined with special controls, situated in blocks where the cap rock properties have been defined using the CROCKTYPE keyword.

See the keywords WEL_OPN_THR and WEL_OPN_MLT in the Reservoir Description section and CRL_WELLS in the Recurrent Data section.

An example is shown in template file gmsmo035.dat.

Improvements in GEM-Surface Network linked models

A number of enhancements have been implemented, including: passing phase gravities and non-hydrocarbon concentrations to allow GEM to run with the surface network in black-oil mode (compositional tracking mode), on-time fraction exchange between GEM and surface network, and the ability to submit jobs using SSH.

Appendix D of the User’s Guide has detailed documentation on all aspects of GEM-surface network modelling.

Well Layer Output

Unlike versions prior to 2008, well layer output is not written to Simulation Results file unless requested via *OUTSRF *WELL *LAYER and associated subkeywords. The default is *OUTSRF *WELL *LAYER *NONE. If no subkeyword follows after *OUTSRF *WELL *LAYER, then *NONE is assumed. Well and block pressures are written under both *OUTSRF *WELL *LAYER *ALL and *OUTSRF *WELL *LAYER *ALL *DOWNHOLE.

Command Line Options

Two new command line options are described in the I/O Control section: -log *log_path* is used to identify a path to where the log file should be written, and –file_no_replace will cause the simulation to stop with a fatal error if an attempt is made to overwrite any existing output file, including the log file.

WELL MANAGEMENT CHANGES

Cycling Groups

Allow the user to specify data for EOR/IOR processes which rely on alternating production and/or injection cycles. Generally multiple cycles are required and each cycle consists of a number of well-defined parts. A typical example is water alternating gas (WAG) injection.

*GCONCYCLE_START allows the user to begin to specify data for EOR/IOR processes which rely on alternating production and/or injection cycles.

*GCONCYCR_START allows the user to revise data specified earlier with *GCONCYCLE_START. The revisions will be applied immediately.

*GCONCYCLE allows the user to turn on or off group cycling control specified earlier with *GCONCYCLE_START. The action indicated with GCONCYCLE takes effect immediately.

*CYCPRT_START allows the user to select the cycle starting part. CYCPRT_END allows the user to select the cycle ending part. Group cycling control must be specified earlier with *GCONCYCLE_START. The action indicated with CYCPRT_START and/or CYCPRT_END take effect immediately.

GAPPOR Stream Specification

The group control may direct a group to switch target (value and stream) if there is a maximum constraint (*GCONP *MAX) being violated. *AUTODRILL followed by the stream identifier ensures that auto-drillable wells can be opened only when the group is apportioned for the desired stream.

Group Constraint Monitor MINREC and Action SHUTCYCINJ

The keyword MINREC specifies that if the total group gas rate available for recycling falls below a specified value, then a remedial action will be taken. This remedial action may be one of *STOP, *SHUTALL or the new action *SHUTCYCINJ. *SHUTCYCINJ indicates that all currently open gas cycling injectors in the group should be shut.

Important Changes in GEM 2007.10

SIMULATOR CHANGES

Improvements to the GEM-GAP Interface

For surface-network linked runs, GEM can now be made to control certain aspects of a linked well's operating condition.

Specifically for linked wells:

1. Any well monitor defined via the *MONITOR keyword is allowed.
2. Keywords that control well and layer operations within a *TRIGGER are allowed.
3. GEM or the surface-network can shut-in a well.

The constraint handling capability of surface-network linked runs in GEM has also been upgraded to permit STO, STG, STW, or STL constraints. Please see Appendix-D of this guide for details.

For GEM runs coupled with GAP, a portion of the GEM simulation results are now available for scripting. This allows RESOLVE to access variables other than IPR tables and can be used by GAP to control the network. A Resolve script can be written to do calculations using results from both GEM and GAP. Accessible GEM results are also reported in the Resolve Reporting window. Examples of processes that can be handled with scripting include voidage replacement on a localized basis, conditional gas recycling, and preferential withdrawal from certain areas of the reservoir.

Leverett J Function

A Leverett J Function option has been implemented in GEM. This option allows the user to replace water-oil and/or gas-liquid capillary pressures with Leverett J functions. Capillary pressures are calculated from the J function tables, input surface tension, and account for the varying permeability and porosity of each block. See the keyword *JFUNC in the Rock-Fluid section.

Functionality identical to IMEX has been implemented, including input of surface tension arrays (*SRFTNW and SRFTNG) and endpoint scaling (*JFWMAX and *JFGMAX). Also, a technique for modeling tilted water oil contacts is available using PCOW_SHIFT or JFW_SHIFT.

Initial Temperature Array

Reservoir temperature can be specified for each block using the array keyword *TEMPER in the Initial Conditions section. This provides complete flexibility in the specification of temperature variation throughout the reservoir.

Specification of Cyclic Group Controls

A new system has been developed to allow the user to specify data for EOR/IOR processes which rely on alternating production and/or injection cycles. Generally multiple cycles are required and each cycle consists of a number of well-defined parts. Examples include water alternating gas injection (WAG) or cyclic steam injection (SSI). A number of criteria are available to stipulate the duration of each cycle part.

As the keywords for these controls are not yet supported in the Builder interface, this feature is released on a “Beta” basis. The documentation for the keywords and a template data set are found in the …\CMG\GEM\2007.10\doc folder.

Speed-up

Some additional code parallelization and code modification for work-saving has resulted in significant speed-up of GEM runs, particularly for models with large numbers of well completions.

Change of Default for Relative Permeability Scaling Algorithm

Beginning with the 2005.XX releases of GEM, an 8-point scaling method was available by specifying *RPT *SCALING-NEW. However, the older 4-point scaling method was still the default. In the 2007.10 release, the 8-point scaling method becomes the default. The 4-point method is available by specifying *RPT *SCALING-OLD.

Matrix-Fracture Transmissibility Multiplier

New keyword *TRANSMF allows you to modify the transmissibility between a fracture block and its co-located matrix block.

Binary Data Reading

New grid_array read_option *BINARY_DATA allows you to acquire, store and transfer data in binary form. This subkeyword was in v2006.10 but did not appear in the User’s Guide.

WELL MANAGEMENT CHANGES

Shut in Wells above Formation

*MODELSHUT indicates that the listed wells identified by well_list will be closed above the formation with crossflow modelling of the constituent (open) layers, whenever the wells are shut in due to a well operation.

Two New Drawdown Constraint Types

Well control keyword *OPERATE has two new subkeywords:

*DWA defines the maximum drawdown within all open layers,

$$\Delta P_d = \max_{l,\,open} \pm \left(Pblock_l - Pwell_l\right)$$

which corresponds to the notion of imposing the maximum drawdown constraint to avoid formation damage.

*DWB specifies the average drawdown for all open layers:

$$\Delta P_d = \pm \sum_{l,\,open} PI_l\left(Pblock_l - Pwell_l\right) \Big/ \sum_{l,open} PI_l$$

weighted by the total product/injectivity (PI) at the reservoir condition.

Monitoring Well Head Pressure Stability

A real well can operate only above a certain rate at which the bottomhole pressure remains in the stable region of the well-head pressure curve (a region in which the curve of WHP vs.

BHP has a positive slope). New *MONITOR subkeyword *WHYSTAB allows you to monitor hydraulic stability for a well.

Voidage Replacement Production Target (*GCONP *VREP)

This indicates that the production wells connected to this group produce an amount of the bottomhole fluid in proportion to the total bottomhole fluid injected into the reservoir by the injection wells connected to this group.

Recycling Production Target (*GCONP *RECYCLE)

This indicates that the production wells connected to this group produce such that the phase injected by the injection wells connected to this group as specified by *GAS or *WATER is reproduced (recycled) out of the reservoir.

Sector Pressure Maintenance Production or Injection Target

New *GCONP and *GCONI subkeyword *PMAINT specifies that the group production / injection rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector (*PMSECT) at a desired level (*PMTARG).

Shutting in Multiple Wells to Meet Production Targets/Monitors (Subkeyword *SHUTMOWS)

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then a list of prioritized most offending wells (MOWS – the wells with the higher rates of the named surface stream) should be shut.

Voidage Replacement Injection Targets including Makeup of Gas or Water (GCONI *VREP *GMKUP/*WMKUP)

A make-up stream can be added with *GMKUP or *WMKUP to meet a total voidage replacement fraction for each group under voidage replacement.

New Apportionment Methods for Meeting Group Targets

*APPOR-METHOD defines how to distribute a group target within its pool – the collection of all the contributing wells or groups. The pool for a targeted group consists of all its child groups and the wells directly attached. . CMG's multi-level well management module currently supports the following methods:

Instantaneous potentials (*IP, default)

Guide rates (*GUIDE)

Internally generated guide rates (*INGUIDE)

Priority ranking (*PRIOR)

*PRIOR-FORM defines the priority formulae and numerical control parameters for the priority ranking apportionment method set by *APPOR-METHOD *PRIOR to meet group targets.

On-Time Factors Applying to Groups

*ON-TIME specifies the fraction of time during which a well or group operates. Please see the *ON-TIME keyword for a detailed description of how *ON-TIME operates with Groups.

Important Changes in GEM 2006.10

1. It is now possible to define conditions (triggers) under which a sequence of recurrent data lines (actions) will be read and executed. In this way it is possible to get a very wide range of actions taken under specified conditions. Please see the manual page for the TRIGGER keyword in the Well and Recurrent Data section of this manual.
2. Some restrictions upon the ordering and timing of well data have been relaxed. Please see the introductory pages of the Well and Recurrent Data section of this manual for more information.
3. A set of wells can be shut and reopened with a single statement in a way that does not disturb the open/shut pattern of wells that existed when the collective shut command was issued. Please see the manual pages for the WLISTSHUT and WLISTOPEN keywords in the Well and Recurrent Data section of this manual.
4. GEM can now be coupled to Petroleum Experts' RESOLVE/GAP or Neotechnology Consultants' FGI/FORGAS surface network models. The link to GAP is through an interface dll (GemGapLink.dll) written by CMG, while FORGAS is linked via an interface (FGI) written by Neotechnology Consultants. These dll's take care of the coupling between GEM and the surface network programs and no additional keyword is required in the GEM input file. The communication between the simulator and surface network model is made through ASCII signal and data files. These files are automatically created and continuously updated by the coupled system.
5. Special care is required while preparing the GEM data file for coupling with the network programs. All the linked wells should be defined at the first *DATE/*TIME keyword in the Wells and Recurrent section of the GEM data file. The constraints and operation of these wells are controlled by the network program and not by GEM. Group production and injection controls (*GCONP and *GCONI) should not be used for groups containing wells controlled by RESOLVE or FGI. Please see the manual pages in Appendix D: Coupling GEM with Surface Network Models, of this manual for further details.
6. Streamline output is enabled for display in Results. This is enabled with the keywords OUTSRF GRID STRMLN. More details can be found in the I/O Control section of this manual.

Important Changes in GEM 2005.10

1. The group well management facility now can model undersea manifolds and the pressure drops between these manifolds and the surface. Both production and injection manifolds can be modeled; the manifolds are represented as groups of wells running with the same manifold (tubing head) pressure. For more information see the documentation of the MANIFOLD, GCONP, GCONI, PCON-MANIFOLD, GPTABLE, GPHYDDEP, GROUPALQ, WELLALQ, and IWELLBORE keywords in the Well and Recurrent Data section of this manual.
2. GEM now is able to read keywords that scale all eight relative permeability endpoints: see descriptions of *SWCRIT, *SGCRIT, *SORW, *SORG, *SWCON, *SGCON, *SOIRW, and *SLCON. GEM can also modify relative permeability table values as well; see descriptions of *KRWIRO, *KROCW, *KRGCL, *KROGCL, *PCWMAX, and *PCGMAX. Also, hysteresis effects are allowed for P_{cwo}, P_{cgo}, k_{rg} and k_{row}. *SCALING-NEW must be stated to specify the treatment of relative permeability tables using this new system (see the Rock-Fluid section of this manual), while *SCALING-OLD can be specified to use the old system (as was used in versions GEM 2004.xx and earlier). If nothing is specified, then *SCALING-OLD is the default. Any use of any of the new keywords (those other than *SWCRIT, *SGCRIT, *SORW, *SORG) must be accompanied by use of *SCALING-NEW. Also *KRPERF (see Recurrent Data section) can now be used in GEM, provided that *SCALING-NEW has been specified.
3. GEM can now be run in parallel using shared-memory multi-processor computers. To invoke the parallel solver and use other parallel features, specify "*SOLVER *PARASOL" and "*DPLANES 4" in the *NUMERICAL section of the data set. The value following *DPLANES does not affect GEM (although it does affect IMEX and STARS when used for those simulators), but the same keyword has been used to make data input similar for the three simulators. See the Numerical section of this manual for further details and additional controls.
4. The Palmer and Mansoori matrix shrinkage and swelling model for coal bed methane applications has been enhanced to take account of gas composition; see the *POISSR, *YOUNGM, … keywords in the Reservoir Description portion of this manual for further discussion. (There are related switches in the Component Properties section (see *PALMAN…), but since the default behavior is preferred, these latter keywords need not be used.)
5. A *TRANSF fault multiplier for named faults has been provided so that an entire fault's multiplier set can be altered at once. See the *TRANSF keyword in the Reservoir Description portion of this manual for a further discussion.
6. Note that any rate or cumulative quantity expressed using "Molar Mass" units in Results Graph can be also viewed as a volume at standard conditions (1 atm and 15.555 deg C / 60 deg F) assuming ideal gas properties (i.e., Z = 1). This kind of plot is useful for estimating the individual component volumes that would be obtained from separation of light gases (for instance, mixtures of CH_4, N_2 and CO_2, for which the Z = 1 assumption is a good one at these conditions) produced

from certain Coal Bed Methane fields. Unit selection can be found under the View|Properties|Units panel in Results Graph; check the “User Defined Unit System” box and then select “scf(Z=1)” or “sm3(Z=1)” for Molar Mass. Once this is done, quantities such as “Gas Mole Rate (Component_Name) SC” will appear as a volumetric rate for the selected component (calculated according to the assumptions described above). This feature offers only one set of standard conditions: 1 atm and 15.555 degC / 60 deg F.

Important Changes in GEM 2004.10

1. There are now keywords (*SOLID-CONV-RATE, …) that model forward and reverse rates for the conversion of precipitated asphaltene to flocculated asphaltene. This conversion may be made reversible, fully irreversible or partially irreversible. Asphaltene “flocs” may also deposit in the porous media under user control.
2. More work has been done fine tuning the *VELDELRP keyword for velocity dependent relative permeability effects in condensate reservoirs. Capillary number can now be output (*CAPN).
3. As well as being able to display well productivities, the user can now directly input fluid phase or total well productivity/injectivity or productivity/injectivity multipliers using the *SETPI keyword.
4. *AMALGAMATED grids can be seen at their finest scale (*RESULTFILE *FINE).
5. There have been changes to *GCONM. New subkeywords (*MAXGAS and *MAXSTW) define group production monitors that trigger upon a high gas rate or high water rate. However, when an action that involves selecting a "most offending" well or layer is specified for either of these two monitors, "most offending" is deemed to mean the well or layer with the highest GOR or water cut, rather than that with the highest gas or water rate.
6. The operation of the *GAPPOR *AUTODRILL apportionment option has been changed so that selection of wells to open automatically can proceed even with no wells initially open in the group whose target is being apportioned. Also, when the wells are ranked for opening by Instantaneous Production Potential, this potential is taken to be the well’s production rate with an assumed BHP of 1 atm. Similarly for injectors the Instantaneous Injection Potential is taken to be the injection rate at a BHP of 3500 atm. See the manual pages for the *GAPPOR and *DRILLQ keywords for more information.
7. Well names up to 40 characters in length are now accepted. The manual page for the *WELL keyword has more information.
8. There have been reductions made to GEM’s total run-time storage requirements.

Important Changes in GEM 2003.10

1. New keywords have been provided controlling Coal Bed Methane (CBM) and Enhanced Coal Bed Methane (ECBM) models.

   *LANG-DIFFUSION-COAL selects a diffusion model based on concentrations calculated from adsorption characteristics while *FREE-DIFFUSION-COAL selects concentrations based on free gas properties. The *LANG-… model better matches desorption times without model tuning when using the *COAL-DIF-TIME keyword, and is recommended for new models.

   NOTE: GEM used *FREE-DIFFUSION-COAL as its default diffusion model for all older releases numbered GEM 2002.xx. GEM 2003.10 now uses the *LANG-DIFFUSION-COAL as its default. The *FREE-DIFFUSION-COAL keyword is provided so older, tuned data sets can be run as before using new versions of GEM.

   *NO-WBLOCK-COAL, *UNI-WBLOCK-COAL and *DBL-WBLOCK-COAL control how the presence of water affects the area available for diffusion as gas comes out of the coal (matrix) into the cleat system (fractures).

   Detailed descriptions of these keywords can be found in the Component Properties portion of this manual.

2. GEM now allows the direct definition of a Palmer and Mansoori model through use of the Compaction/Dilation feature without having to generate an external table. This feature is useful for modelling coal shrinkage and swelling for CBM and ECBM settings. See *CROCKTYPE, and the subsequent section referring to *POISSR, *YOUNGM, *STRINF, *PRESLN, and *EXPPM, in the Reservoir Description portion of this manual for a discussion of how to implement this option.

3. An additional pair of keywords for defining sectors in a model has been implemented. Use *SECTORNAME to define sector names and *ISECTOR to position these sectors in the reservoir. See the Reservoir Description portion of this manual for a discussion of these keywords.

4. *TORPSDIFF, *TORNOPSDIFF and *TORNOPS2DIFF have been provided to select different options controlling the tortuosity model and the selective inclusion of the term converting bulk velocities to pore velocities when molecular diffusion is being modelled. See the section referring to *DIFCOR-OIL/-GAS/-AQU and *DIFFC-OIL/-GAS/-AQU in the Component Properties portion of this manual.

5. Two new keywords *HEADITER and *BHPHEADINIT have been added to the Well and Recurrent Data section. These allow iterative calculation of well layer heads so that wellbore pressures in the completions are consistent with the flow rates which result from these pressures.

Important Changes in GEM 2002.10

1. GEM now allows the modelling of Compaction/Dilation for reservoir rock. The implementation of this option is similar to that found in IMEX, with similar functionality. Besides (again) providing a linear compressibility option, the Compaction/Dilation option allows modelling of various rock mechanic processes through the input of tables of porosity and permeability multipliers vs. pressure. In addition, various options for modelling reversibly and hysteresis can be specified. See the Reservoir Description portion of this manual.
2. GEM now supports Coal bed Methane (CBM) and Enhanced Coal Bed Methane (ECBM) features. See the Coal Bed Methane Features section in the Rock-Fluid portion of this Manual. Compaction/Dilation can be used with CBM.
3. Multicomponent Adsorption from the Gas Phase through tabular adsorption data entry is now available for entering tables of adsorption data in multicomponent systems. This provides an alternative to the extended Langmuir isotherm modelling technique for multicomponent systems, which GEM retains. A special format for binary system data is provided. See the Rock-Fluid portion of this Manual.
4. A Velocity Dependent Relative Permeability Option is now available that makes relative permeabilities dependent on phase velocity and interfacial tension. This option is designed to model gas condensate systems where the reduction in relative permeability due to condensate dropout is mitigated by low interfacial tension and/or the high fluid velocities, especially in the near well region. See the Rock-Fluid portion of this Manual.
5. It is now possible for a user to specify that an internally generated table based on GEM's WHP correlation be used for computing tubing head pressure. The keyword which invokes this technique is *WHPCRTB, the purpose of which is to allow faster simulations using the WHP correlation with nearly identical results.
6. Reporting groups have been introduced to allow reporting of results for well aggregates which are very flexibly defined, without the restrictions which apply to hierarchical groups through which group controls are imposed. The *REPORTING-GROUP keyword allows the user to define the reporting groups.
7. A bottomhole fluid (BHF) group production rate target and monitor has been introduced (see the entries for the *GCONM, *GCONP, and *GUIDEP keywords). The BHF group target allows group production strategies to be set during prediction runs which may better allow reservoir pressure to be maintained while respecting maximum rate limits upon injectors. This would be done by using the BHF production target with the injection voidage replacement target.

Important Changes in GEM 2001.10

1. Enhancements to well management introduced in this release are in two areas: group recycling injection targets (keywords *GCONI, *GPRODGROUP, *WPRODGROUP, *GSALESFRAC, *GSALESMAXR,*GFUELFRAC, *GFUELMAXR, *WMKMAXR, *WMKUPTO, *GMKMAXR, *GMKUPTO, *GMKCOMP, *WRECYMAXR, *GRECYMAXR, *GSALESMASK, *GFUELMASK, and *GRECYMASK), and wellbore and bottomhole pressures (keywords *BHPDEPTH, *BHPGRAD, and *LAYERGRAD).
2. The enhancements in recycling allow fuel and sales rates to diminish the amount of gas available for re-injection (keywords *GSALESFRAC, *GSALESMAXR, *GFUELFRAC, *GFUELMAXR, *GSALESMASK, and *GFUELMASK). Also, the production fixing the injection target can come from a named group (keywords *GPRODGROUP and *WPRODGROUP). Gas and water make-up injection can run in either a constant-rate mode or a mode in which the make-up rate is adjusted to fix total injection at a given rate (if possible) using the given well constraints (keywords *WMKMAXR, *WMKUPTO, *GMKMAXR, and *GMKUPTO). It is also possible to cap recycling injection rates at a given maximum value (keywords *WRECYMAXR and *GRECYMAXR).
3. The enhancements in the handling of wellbore and bottomhole pressures allow a well’s bottomhole pressure to be referred to a user-specified depth rather than being the wellbore pressure in the reference completion (keyword *BHPDEPTH). It is also possible to specify the pressure gradient value which will be used to compute the difference between the wellbore pressure in the reference completion and the bottomhole reference depth (keyword *BHPGRAD). In addition the user can now specify that certain layers (or all layers) of a well should use a specified pressure gradient for calculating the wellbore pressure differences between adjacent completions (keyword *LAYERGRAD).
4. The anisotropic Peaceman well model has been added to the simulator. This model accounts automatically for permeability anisotropy and grid block aspect ratio.
5. Some changes have been made to GEM's strategies regarding pinched out cells. The *PINCHOUTARRAY keyword cannot be combined in the same data set with the *PINCHOUT-TOL keyword. Also, default tolerances for *PINCHOUT-TOL have been changed: 0.0010 (m|ft) for corner-point grids on the fundamental grid only, if *PINCHOUTARRAY does not appear, 0.0002 (m|ft) for non-corner-point grids on the fundamental grid only, if *PINCHOUTARRAY does not appear, with 0.0 (m|ft) otherwise. Pinched out (and null) cells can be marked as refined in data, without terminating the simulation, although these cells will remain inactive. *PINCHOUT should no longer be used.
6. Refined grids are permitted in Recurrent data.
7. GEM has a numerical dispersion control option that is activated with the *TWOPTFLUX keyword. This option blends the usual up-streamed fluxes with higher-order, averaged, two-point fluxes under the control of a Total Variation Limiting (TVD) flux limiter. The TVD limiter ensures that the overall numerical

scheme maintains numerical stability and eliminates over- and under-shoot in the calculations. The limiter allows the higher-order flux calculation to predominate away from the edges of saturation fronts and fluid banks, and reverts to the more stable upstream scheme near frontal regions where compositional variable and/or saturations are changing rapidly. Much better resolution over fronts (over a few grid cells) can be seen.

8. GEM now reads strings for special history output (*OUTPRN and *OUTSRF). GEM dimensioning is more robust and accounts better for null cells and well dimensions. The Grid Module requires less storage and is faster for many cases.

Important Changes in GEM 2000.10

1. GEM can model the formation of asphaltene and its subsequent adsorption on the reservoir rock. (CMG's WINPROP can be used to construct the fluid characterization for this type of simulation.) A new oil-gas-water flash is available when aqueous phase solubility is modelled. Multicomponent gas phase adsorption is available, as is phase velocity-dependent dispersion (isotropic, or with separate specifications of longitudinal and transverse dispersivities). GEM also has a new viscosity correlation for heavier fluids due to Pedersen and correlations for modelling molecular diffusion. GEM can also model some additional gas phase flow mechanisms such as non-Darcy flow (which includes some inertial effects) and a Klinkenberg flow modification.
2. GEM now accepts the *TURB sub-keyword for the *PERF, *PERFV, *PERFRG, and *PERFVHY keywords to activate the turbulent skin modification of the layer reservoir gas phase productivity/injectivity calculation. An additional parameter value (the D-factor) must be entered for each layer on the perforation line when this option in is force. GEM now also supports two different ways of accounting for the layer gas phase productivity/injectivity changes due to the property changes which the reservoir gas undergoes as the pressure changes from the grid block value to the wellbore value. The first way, activated by the *QUAD sub-keyword of *PERF, *PERFRG, *PERFV, and *PERFVHY, is the quadratic inflow technique, in which the gas viscosity is assumed to remain at the grid block value and the gas density is assumed to vary linearly with pressure. The second, activated with the *PSEUDOP sub-keyword, includes a full recalculation of the gas density and viscosity at the wellbore pressure. No additional data are necessary for the *QUAD and *PSEUDOP modifications; the two cannot be applied simultaneously, as they represent two different ways of treating the same effect. Please consult the manual pages in this chapter describing the *PERF, *PERFRG, *PERFV, and *PERFVHY keywords for more information.

   In addition, the *PERF keyword now accepts the general i1 j1 k1 / i2 j2 k2 / … format for specifying the grid block in which a layer is perforated. This means that all perforations into refined grids, including hybrid grids, can now be done through *PERF. The *PERFRG and *PERFVHY keywords are still supported but are no longer necessary. Please see the manual page for the *PERF keyword for more information.

Introduction to GEM

INTRODUCTION

In enhanced recovery schemes involving gas or solvent injection, the process may be immiscible or miscible depending on the composition of the injected fluid and the reservoir oil, and on the reservoir pressure and temperature. Examples of such processes are enriched gas drive, high pressure gas drive, CO_2 flooding, and the cycling of a gas condensate reservoir. The simulation of these processes requires special handling of both the thermodynamic and the fluid flow aspects of the reservoir.

GEM is an efficient, multidimensional, equation-of- state (EOS) compositional simulator which can simulate all the important mechanisms of a miscible gas injection process, i.e. vaporization and swelling of oil, condensation of gas, viscosity and interfacial tension reduction, and the formation of a miscible solvent bank through multiple contacts.

Some of the additional features of GEM are listed in the following.

ADAPTIVE IMPLICIT FORMULATION

GEM can be run in explicit, fully implicit and adaptive implicit modes. In many cases, only a small number of grid blocks need to be solved fully implicitly; most blocks can be solved explicitly. The adaptive implicit option selects a block's implicitness dynamically during the computation and is useful for coning problems where high flow rates occur near the wellbore, or in stratified reservoirs with very thin layers. Several options are provided for selecting implicit treatment.

PROPERTIES

GEM utilizes either the Peng-Robinson or the Soave- Redlich-Kwong equation of state to predict the phase equilibrium compositions and densities of the oil and gas phases, and supports various schemes for computing related properties such as oil and gas viscosities.

The quasi-Newton successive substitution method, QNSS, as developed at CMG, is used to solve the nonlinear equations associated with the flash calculations. A robust stability test based on a Gibbs energy analysis is used to detect single phase situations. GEM can align the flash equations with the reservoir flow equations to obtain an efficient solution of the equations at each timestep.

CMG's WINPROP equation of state software can be used to prepare EOS data for GEM.

COMPLEX RESERVOIRS

GEM uses CMG's Grid Module for interpreting the Reservoir definition keywords used to describe a complex reservoir. Grids can be of Variable Thickness - Variable Depth type, or be of corner-point type, either with or without user-controlled Faulting. Other types of grids, such as Cartesian and Cylindrical, are supported as well as locally Refined Grids of both Cartesian and Hybrid type. Note that Hybrid refined grids are of a locally cylindrical or elliptical nature that may prove useful for near-well computations.

Regional definitions for rock-fluid types, initialization parameters, EOS parameter types, sector reporting, aquifers, ... are available. Initial reservoir conditions can be established with given gas-oil and oil-water contact depths. Given proper data (such as from WINPROP), fluid

composition can be initialized such that it varies with depth. A linear reservoir temperature gradient may also be specified.

Aquifers are modelled by either adding boundary cells which contain only water or by the use of the analytical aquifer model proposed by Carter and Tracy.

Dual porosity modelling can be done with GEM. Each cell is assigned separate matrix and fracture pore spaces. Shape factors describing flow between porosities are implemented based on the work of Gilman and Kazemi. Additional transfer enhancements are available to account for fluid placement in the fractures. The GEM user can also specify a dual permeability model which allows fluid flow between adjacent matrix blocks. This option is useful when matrix-matrix mass transfer processes are important, such as in situations dominated by gas-oil gravity drainage processes.

GEOMECHANICAL MODEL

Several production practices depend critically on the fact that the producing formation responds dynamically to changes in applied stresses. These include plastic deformation, shear dilatancy, and compaction drive in cyclic injection/production strategies, injection induced fracturing, as well as near-well formation failure and sand co-production. A geomechanical model consisting of three submodules is available for treating aspects of the above problems. The coupling between the geomechanical model and the simulator is done in a modular and explicit fashion. This increases the flexibility and portability of the model, and decreases computational costs.

WELLS

Bottomhole pressure and the block variables for the blocks where wells are completed are solved fully implicitly. If a well is completed in more than one layer, its bottomhole pressure is solved in a fully coupled manner; i.e., all completions are accounted for. This eliminates convergence problems for wells with multiple completions in highly stratified reservoirs.

A comprehensive well control facility is available. An extensive list of constraints (maximum/minimum bottomhole or wellhead pressures, rates, WCUTs, GORs, ...) can be entered. As constraints are violated, new constraints can be selected according to the user's specifications. Various actions and apportionments are available.

Up to three hydrocarbon streams can be controlled on the surface: Oil, Intermediate Liquid and Gas. Various types of surface separation facilities can be used to generate these streams, including the modelling of EOS and plant separator stages, where the latter are described using key-component tables.

The gas cycling option in GEM allows for the preferential stripping of components and the addition of a make-up gas stream to the recycling gas stream.

MATRIX SOLUTION METHOD

GEM uses AIMSOL, which is a state-of-the-art linear solution routine based on incomplete Gaussian Elimination as a preconditioning step to a GMRES iteration. AIMSOL has been developed especially for adaptive implicit Jacobian matrices.

For almost all applications, the default control values selected by GEM will enable AIMSOL to perform efficiently. Thus, GEM users do not require detailed knowledge of the matrix solution methods.

GEM uses run-time dimensioning as well to make the most efficient use of computer resources.

SIMULATION RESULTS FILES

Various types of Simulation Results Files can be written while GEM is running, including files for CMG's RESULTS. RESULTS is CMG's visualization software that can be used to examine 2-D and 3-D reservoir displays, as well as XY plots of important dynamic data.

PORTABILITY

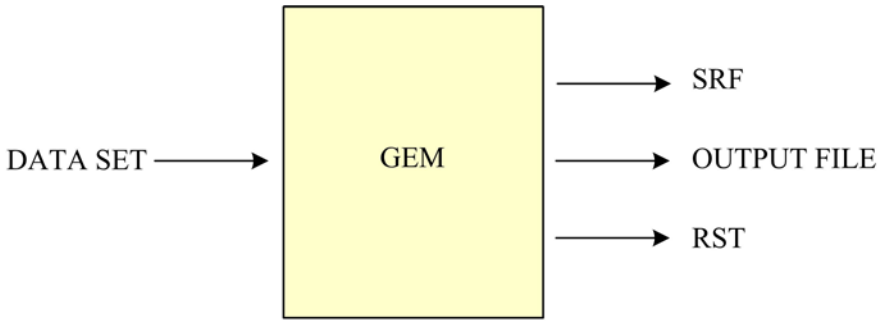
GEM has been run on many computers from many manufacturers, such as IBM, SGI, and SUN, as well as PCs. Currently supported chips and operating systems are given in the Installation Guide.

Tutorial

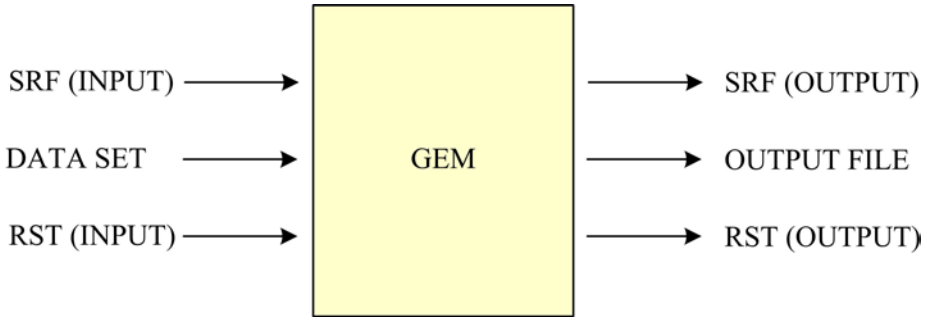
Introduction

The tutorial section is a guide for the novice user of the keyword input system. It does not replace the reference user manual in this document. Only specific keywords and topics are discussed in this tutorial section. The user manual contains a detailed description of each keyword, while the tutorial section tackles "how-to" questions you may have when building a data set.

GEM uses the data set that you create initially and then creates three other files. Each GEM run may create an output restart file (RST), an output Simulation Results File (SRF), and an output file:



If a restart run is desired, then several existing files are needed and another three are generated. This is illustrated in the diagram:



Data Groups in the Keyword Input System

There are several points to remember when you build a data set using the keyword input system:

a) There are seven different data groups in the keyword input system.

b) The groups must follow a certain input order:

   I/O Control
   Reservoir Description
   Component Properties
   Rock-fluid Data
   Initial Conditions
   Numerical Methods Control
   Geomechanics Data
   Well Data and Recurrent Data

c) The keywords belonging to each group cannot appear in other groups, unless it is specifically written. Usually, this happens with recurrent data from other sections which may be changed in the Well Data section.

d) Also pay attention to the order that some keywords, within a group, are entered.

How to Document Your Data Set

Documenting your data set is done with the keywords:

a) *TITLE1

b) *TITLE2

c) *TITLE3 and

d) *CASEID

They must be located in the Input/Output Control section.

These keywords are optional and may be excluded from the data set; however they are very useful for documenting files and distinguishing similar data sets from each other. At least one title should be used. All titles and the case identification must be enclosed within single quotes.

*TITLE1 and *CASEID are both used in the Simulation Results File which is used to create graphics of the simulation. *TITLE1 may be as long as 40 characters, but both *TITLE2 and *TITLE3 are allowed up to 80 characters each. The case identification is limited to 8 characters.

You may also use two keyword indicators or '**' to insert comments throughout your data set. Comments may appear anywhere in your data set.

Example:

```
*TITLE1
'Simulation Run #1 - 1989-01-23'
*TITLE2
'Dual Porosity Problem using the MINC option'
*TITLE3
'This is a 12 x 12 x 10 Cartesian grid system'
*CASEID 'RUN1'
** You may add additional information here or
** anywhere if the title lines did not allow
** enough room for documenting this data set.
** You may also use the comments to describe
** your data as you enter it.
```

How to Do a Restart

WHAT IS A RESTART FILE?

A restart (RST) file in GEM is a binary file with extension .rst. Information about the initial data and the primary variables is written to the file at the frequency you have chosen. Writing a restart file is optional.

WHY WOULD YOU NEED TO DO RESTART?

You may want to do restarts for the following reasons:

a) To do sensitivity studies or history matching,

b) To change well specifications,

c) To perform a short simulation run to see if the results are satisfactory, before running bigger, longer jobs, and

d) To save execution time in subsequent runs. For instance, you have completed a simulation run and the preliminary results look good. Now you want to do prediction runs.

   Because you have created restart records with the initial or parent run, you may select a time step, date, or time from the middle of your run and 'restart' the simulation. The simulator does not need to start at the beginning; it continues execution from the time you have chosen.

CREATING RESTART RECORDS

If you do plan on doing restarts, you need to create one with your parent run.

Use keyword *WRST to create your restart file. It must be located in the Input/Output Control section of your data set. The *WRST keyword may also appear in the Well and Recurrent Data section when well changes are done.

*WRST indicates the frequency of writing the restart records. In the Input / Output Control section, if the frequency is set to zero (i.e. *WRST 0), no restart file is created and no restart records can subsequently be written in Well and Recurrent Data section. However if in the Input / Output Control section, *WRST keyword is not specified or if the frequency is set to a very large number (say, *WRST 100000) the restart file is generated and *WRST keyword can be modified later in the Wells and Recurrent Data section to write restart(s) at desired time.

Example:

```
*WRST    10
```

While writing restart records insures that you have the ability to re-start the run from the time where the restart record is written, caution needs to be exercised while writing the restart records at a pre-determined frequency. Since large amount of information is written to restart file to exactly re-create the condition in the restarted run, the size of the restart file could be overwhelmingly large especially for models with a large number of grid blocks if the restart records are written very frequently. Apart from this, the simulation run time could increase for the run wherein too many restarts are being written. Note that writing of restarts is done in

serial mode. As such the run-time of large models could increase significantly, if too many restarts are being written, even though the simulation is being done in a parallel mode.

USING RESTART RECORDS

Keyword *RESTART, *RESDATE, or *RESTIME are used in the Input / Output Control section of the data file to re-start a run from a specified timestep, data, or time, at which restart record is available in the parent run. Presence of these keywords indicates that the current simulation run is a restart run. If you wish to start at the last timestep, then leave the rest of the line after *RESTART empty.

To do a restart run:

a) Do not change any of the original reservoir data, but do add either of *RESTART, *RESDATE or *RESTIME to the Input / Output Control section of your data set.

b) Increase the maximum number of timesteps, if necessary, or leave out *MAXSTEPS altogether.

c) You will need input files (which were created during the initial run), including an input restart file and an input simulation results file. The input file can be specified using keyword, say, *FILENAME *INDEX-IN ‘file.irf’ in the Input / Output Control section of the current simulation run. Note that ‘file.dat’ is the parent file’s name in this example.

d) You will also need names for the new set of output files. Be sure that existing files resulted from the same job.

Example 1:

```
*FILENAME *INDEX-IN ‘sim1.irf’
*RESTART 575   ** restart the run from time-step 575 of the
               ** parent run sim1.
```

Example 2:

```
*FILENAME *INDEX-IN ‘sim1.irf’
*RESDATE 2013 04 10   ** restart the run from date 2013 04 10
                      ** of the parent run sim1.
```

Example 3:

```
*FILENAME *INDEX-IN ‘sim1.irf’
*RESTIME 3650.   ** restart the run from 3650 days of the
                 ** parent run sim1.
```

Controlling Contents of the Output Print File

To control the contents of the output print file, use:

a) *WPRN and

b) *OUTPRN

These keywords may appear in the Input/Output Control section or the parameters may be changed later on in the data set in the Well Data section.

*WPRN indicates how often to write grid block data, well data, and numerical method control data, such as Newton iterations and timestep convergence behavior.

If no grid, sector or well information is desired in the output file, then the frequency may be set to zero.

Example:

```
*WPRN *WELL   0
*WPRN *GRID   0
```

If either of these is left out of the data set, then the default is to print the information at every timestep. This can produce a very large output print file, which can fill up the available space on your computer very quickly.

*OUTPRN limits what reservoir information, well information are printed.

Detailed grid block information and fluid property data could be produced by using *OUTPRN *RES.

Controlling Contents of the Simulation Results File

Simulation Results File (SRF) is a data file generated by GEM during simulation runs. The SRF file is intended to be read by CMG's Graphic and Control System to generate graphics outputs. To control the contents of the SRF, use:

a) *WSRF and

b) *OUTSRF

These keywords normally first appear in the Input/Output Control section.

*WSRF indicates how often to write grid block data and well data.

If no grid or well information is desired in the output print file, then the frequency is set to zero. This may be used to cut down the size of a very large file. You may, however, change this in subsequent well changes.

If *WSRF is left out of the data set, then the default action taken is to print well information at each time step and print grid information at every well change.

*OUTSRF limits what well data, grid data, and reservoir data are printed. You may also print special variables at specified grid block locations. Separate lists of variables are available for grid information and well information. The well information list cannot be changed during simulation, nor can it be changed in restart runs.

Describing Your Grid System

To describe your grid system, use:

a) *GRID and

b) *KDIR

To describe the grid dimensions, use:

c) *DI

d) *DJ

e) *DK

To describe the reservoir depth and tilt use either:

f) *DEPTH and

g) *DIP

-or-

h) *DTOP

Among these keywords, only *KDIR and *DIP are totally optional and may be left out of the data set.

The keywords listed above must appear in the Reservoir Description section. *GRID must be the first keyword in this section.

*GRID describes the type of grid system that is being used. There are four choices: regular Cartesian, variable depth/variable thickness, radial-angular cylindrical, and corner point. Each of these choices requires the number of grid blocks in the I (x or r) direction, in the J (y or theta) direction, and in the K (z or vertical) direction.

Example:

```
*GRID *CART 10 10 6
*GRID *VARI 10 10 6
*GRID *RADIAL 10 1 15
*GRID *CORNER 10 10 6
```

The first describes a regular Cartesian grid that is 10x10x6. The second describes a variable depth/ variable thickness grid that is also 10x10x6. The third example describes a radial-angular cylindrical system of size 10x1x15 for a coning study. The last sets up a corner point grid.

The keywords *DI, *DJ, and *DK are usually used (except for *CORNER). You enter the dimensions of the grid blocks using these three keywords. Note the use of array reading options:

Example:

```
*GRID *CART 10 10 12
*DI *CON 100.0
*DJ *CON 100.0
*DK *KVAR
25.0 2*50.0 3*40.0 75.0 3*40 2*50
```

Here the grid system is a regular Cartesian grid system. Each of the 10 grid blocks in the I direction is 100.00 meters wide. Each of the 10 grid block in the J direction is 100.0 meters wide and each layer in the K direction has the same thickness but the thicknesses differ between layers.

*KDIR is used to define the direction of the vertical axis. The default for *KDIR is *UP and layer 1 is the bottommost layer (see FIGURE 1 in Appendix B).

Describing Refined Grid

To describe the location of a refined grid, use *REFINE. You may also use the hybrid grid option with *REFINE. Hybrid grids are cylindrical refinements within a Cartesian grid system. *REFINE must appear in the Reservoir Description section.

*REFINE requires the number of refined blocks each fundamental grid blocks will be split up into for each direction where refinement is desired. *HYBRID allows you to split a Cartesian well block into cylindrical cells. You must use *RANGE to indicate the location of the fundamental grid blocks where refinement will occur.

For example, if you want to split block (1,1,3) in a 10x10x3 regular Cartesian grid system into 2 refined grid blocks in the I direction, 3 blocks in the J direction and 2 in the K direction, the keywords should look like this:

Example:

```
*REFINE 2 3 2
*RANGE 1:1 1:1 3:3
```

If you want to split different areas into different configurations, then you may use subsequent *REFINE keywords, being sure to keep track of the correct locations of those fundamental blocks.

Note that *HYBRID areas that are split into different configurations adjacent to each other have certain restrictions (See the *REFINE keyword for details).

Now that you have stated that you want to use refined grid, you must use *RG for any properties that may differ from the properties of the corresponding fundamental blocks. Otherwise, the properties of the fundamental blocks are attributed to the refined grid blocks.

Entering Null Blocks

There are two ways to indicate the presence of null (nonparticipating flow) blocks within a given grid system:

a) *NULL and

b) *POR

Both must appear in the Reservoir Description section. *POR must appear in any data set, but *NULL is optional.

Null blocks can be designated by assigning a block a porosity of 0.0. Thus, blocks with nonzero porosities are considered active unless over-ridden by *NULL as described below.

With *NULL, null blocks are indicated by the number 0; active blocks are indicated by the number 1. In the example below, all blocks except blocks 1 to 4 in the I direction, 1 to 3 in J direction and blocks 1 to 3 in the K direction, are active. The *IJK array reading option is being used for this example:

Example:

```
*NULL *IJK
1:10 1:10 1:3 1
1:4  1:3  1:3 0
```

Observe that the second line overrides the first line. *NULL is optional and if it is not present, it is assumed that all blocks are active.

The designation by *NULL always overrides any designations by *POR. Even if a nonzero porosity is assigned to a block that has been designated a null block by *NULL (*NULL assigned a 0 to the block), the block will not be considered in flow calculations.

The aforementioned example may also be obtained by use of *POR:

Example:

```
*POR *IJK
1:10 1:10 1:3 0.3
1:4  1:3  1:3 0.0
```

Using Dual Porosity/Dual Permeability

To invoke the dual porosity/dual permeability options you may use (one, and only one of):

a) *DUALPOR or

b) *DUALPERM

Dual porosity/permeability reservoir models assume that it takes two porosity systems to make up a grid block: a matrix system and a fracture system.

In addition to the regular grid information for fractured media, you also need:

c) DIFRAC

d) DJFRAC

e) DKFRAC

f) TRANSFER

These keywords must appear in the Reservoir Description section.

In the component properties section of the data set, you may also specify

g) *DIFFUSION

to include a gas-phase matrix/fracture diffusion mass transfer mechanism.

The input of porosity values requires input of values for both the matrix and the fracture.

Example:

```
*POR *MATRIX *IJK
1:10 1:10 1:3 0.3
1:4  1:3  1:3 0.0

*POR *FRACTURE *IJK
1:10 1:10 1:3 0.0
8    7:9  1:2 0.4
*MOD
8    7:8  1 = 0.45
```

The example also illustrates the use of *MOD, which modifies the grid property of some blocks, from a porosity of 0.40 to a new porosity of 0.45.

With dual porosity, null blocks imply that both the matrix and the fracture have zero porosity. Blocks can also have each individual porosity equal to 0.

Using the Aquifer Option

To use the aquifer model, you need:

a) *AQUIFER

b) *AQPROP and

c) *AQFUNC

These keywords are expected in the Reservoir Description section and are completely optional.

Problems with Small Timesteps or Long Execution Times

Before calling CMG, it is extremely helpful to rerun the problem with

*WPRN *ITER *MATRIX

which turns on the matrix convergence as well as the Newtonian iteration convergence diagnostics.

You can also print the convergence status flags to log file by using keyword *DIARY *CHANGES-UNCONV in the INPUT OUTPUT CONTROL section or in WELLS AND RECURRENT DATA section. The status flags are printed for primary variables (var), primary equation residual (res), well equation residual (wel), constraint equation residual (cer) and geomech coupling convergence (gmc). A value of 0 indicates convergence, while a value of 1 indicates no convergence. A blank in the column indicates that particular convergence is not being checked. GEM considers the timestep to have converged when either all of the primary variables or all of the primary equation residuals have converged to their tolerance values. However you may want to converge on constraint equation residuals as well by using keyword *NCHECK-CEQ and/or *CONVERGE *CEQ… in the NUMERICAL CONTROL section to obtain a better physical behavior in certain problems like CO2 sequestration, CBM/ECBM and asphaltene modeling. Note that it is normal to get occasional constraint equation residual convergence failure when default values (1.E-10) are in use. Also note that use of *CONVERGE *CEQ… may in certain instances increase the number of Newtonian iterations and thus the CPU time in the simulation.

Convergence failure may result due to:

a) Inner iteration convergence failure,

b) Newtonian convergence failure resulting in time step cuts, or

c) Material balance error

If you find in the output file that the "iteration routine fails to converge" frequently, then try these remedies:

1. Take smaller time steps. This is done by setting a smaller maximum time step size with *DTMAX or reducing the desired changes per time step with *NORM *PRESS and/or *NORM *SATUR.
2. Increase the number of specified iterations by using the keyword *ITERMAX, or
3. Increase the degree of factorization by using *SDEGREE. Please note that this remedy increases storage requirements.

   If the iteration routine fails to converge on the first or second Newton iterations, but converges on at least the last one then it is not a serious problem.

Newtonian iteration convergence failure results in time-step cuts and are caused by maximum changes, in the primary variables, which exceed the specified nominal changes by a factor more than two per time step. Nonphysical values such as negative pressures and/or saturations may be encountered or the specified maximum number of Newtonian iterations is exceeded.

If the problem is caused by maximum changes, it is not a major problem IF it does not occur often.

If large numbers of time steps are repeated, then you should try the following remedies:

4. Check the rock and PVT curves for nonlinearities; the curves should be smooth.
5. Check that grid and other properties are properly specified.
6. Check the well constraint specification. It is good practice to always specify a maximum bottomhole pressure for each injector and a minimum bottomhole pressure for each producer.
7. Increase the specified number of Newton's cycle using the keyword *NEWTONCYC if nonconvergence is caused by the maximum number of Newtonian iterations being exceeded.

   If the maximum number of iterations is reached due to an oscillation in values as exhibited by an oscillation in maximum changes, then smoothing nonlinearities (4) or reducing the time step size (1) are better solutions.
8. Check that the reservoir is not flipping between single phase oil and single phase gas between iterations. If so, set the proper *PHASEID or *REFDEN.
9. Set some regions of the reservoir or the entire reservoir to fully implicit. The default switching criterion, *AIM *STAB checks for switching from explicit to implicit only if the grid block is a neighbour of an implicit block. Thus if there are regions of the reservoir where there are dramatic changes taking place and these regions are not adjacent to wells, or a fluid front move so rapidly that it tries to sweep through two grid blocks at one time step; then set these problem regions to implicit.

   Examples of such situations include:
   a) When vertical equilibrium initialization is not used. In some cases, this may result in large changes in pressure and saturation initially, even if all the wells are shut in. Run fully implicit when this happens.
   b) When there is gas cap. The bottom layer of the gas cap can be set implicit if there is strong cusping - at least in the regions where the cusping occurs.
   c) Where blocks have extremely high permeability. Small changes in pressure make very large changes in saturation. Set blocks to fully implicit in these regions.

Material balance errors can be caused by convergence tolerances being too large compared to the nominal change at each time step. Check to make sure that the tolerances are about one order of magnitude less than the nominal values.

In most cases, the default values for desired changes, *NORM, and the tolerances, *CONVERGE, are adequate. However, in cases where you are trying to simulate fractured reservoirs or use fractures to represent horizontal wells, it is recommended to use smaller values. For coning problems, smaller values of desired changes are also recommended.

Definitions of some iteration variables found in the output file (when *WPRN *ITER *MATRIX is used) include:

jconv	-	number of equations not converged
dp	-	maximum change in pressure over a time step
ds	-	maximum change in saturation over a time step
dz	-	maximum change in molar volume over a time step
max V	-	maximum change in the primary variables over a Newtonian iteration
nitr	-	iteration number in subroutine jsolv
omega	-	if first omega is small, e.g. 1.0E-3, then the iterative routine has difficulty converging
rms	-	sum of the squares of the residuals
rmsi	-	initial rms
rms/rmsi	-	linear solver has converged if rms/rmsi less than 1.0E-6 or when all current updates are less than the convergence tolerance multiply to the relative tolerance

Simulating a Single Phase Reservoir

Label the single phase fluid as either 'oil' or 'gas' using *PHASEID. Normally 'oil' is recommended such that the relative permeability of the fluid will be calculated from the water-oil relative permeability curve.

You can use the keyword *PSAT to set the saturation pressure to be a small number, e.g. -one. In this case, all reservoir fluid will be undersaturated and the expensive flash calculations in each grid block will be skipped. All fluid properties will still be calculated using the EOS. This PSAT will not affect the well stream calculations.

Horizontal Wells

Horizontal wells can be simulated with GEM as a line source (injector) or sink (producer). The wellbore frictional pressure drop as well as liquid hold-up is calculated in GEM using the flow map of Aziz et al.

This approach provides more accurate results compared to the traditional method where frictional pressure drops are ignored.

Note that the current method cannot handle fluid backflow and if the field of interest has any known backflow problems, GEM will give erroneous results. A small amount of backflow should not affect the results greatly.

Vertical Equilibrium Calculation

*VERTICAL controls the vertical equilibrium calculation.

To perform the vertical equilibrium calculation use either:

a) *VERTICAL *ON, with
   *DWOC
   *REFDEPTH
   *REFPRES
   *ZGLOBAL and
   *SWOC

   -or-

b) *VERTICAL *GASCAP, with
   *DWOC
   *DGOC
   *REFDEPTH
   *REFPRES
   *ZOIL
   *ZGAS and
   *SWOC

   -or-

c) *VERTICAL *COMP, with
   *DWOC
   *REFDEPTH
   *REFPRES
   *SWOC
   *CDEPTH and
   *ZDEPTH

These keywords must be located in the Initial Conditions section.

*VERTICAL *ON is designed mainly for undersaturated reservoirs, whereas *VERTICAL *GASCAP could be used for both saturated and undersaturated reservoirs.

For *VERTICAL *ON, gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and water saturations. No special treatment is applied when a grid block contains both the oil and gas phases. Thus, drastic fluid rearrangements may occur at the first time step.

*VERTICAL *GASCAP is a more general option where a gas cap in equilibrium with an oil zone is assumed. Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and all oil, gas and water saturations. Flash calculations are performed to determine the oil phase and gas phase compositions. Block global compositions are then calculated by mixing the oil and gas phase according to the saturations. Thus, global composition in the reservoir may vary with depth.

*VERTICAL *COMP is intended for gravity initialization in either under-saturated or saturated reservoirs with user specified variations of composition with depth. In this approach, the water-oil contact depth, together with a reference pressure at a reference depth are specified. A table of global composition vs. depth must be entered by the user. The user must also enter a critical depth; at depths below the critical depth, single hydrocarbon phase blocks are identified as oil blocks, while at depths above the critical depth, single hydrocarbon phase blocks are identified as containing gas.

Note that all fluids are flashed through the initialization separator to obtain surface volumes for reporting.

If you decide not to perform the vertical equilibrium calculation, enter:

```
*VERTICAL *OFF, with
*PRES,
*SW and
*ZGLOBAL.
```

Again, these keywords must be located in the Initial Conditions section.

The first example illustrates the Initial Conditions section of a data set where the vertical equilibrium calculation is desired for a saturated reservoir:

```
*INITIAL
*VERTICAL *GASCAP
*REFDEPTH  900.0
*REFPRES  3000.0
*DWOC     1250.0
*DGOC      788.0
*ZOIL 0.30 0.10 0.20 0.30 0.12 0.08
*ZGAS 0.78 0.19 0.03 0.0  0.0  0.0
*SWOC 1.0
```

The second example illustrates the portion of the data set when vertical equilibrium is not used

Example:

```
*INITIAL
*VERTICAL *OFF
*PRES     *CON 3400.0
*SW       *CON    0.2
*GLOBAL   *CON
          0.30  0.10  0.20  0.30  0.12  0.08
```

Defining Wells

Wells are defined using the following keywords:

*WELL
    well_number 'well_name'

Well types are defined using:

*PRODUCER or *INJECTOR or *CYCLPROD
    well_nums

Under the above well types, enter:

*PWELLBORE or *IWELLBORE
    (Wellbore data required if wellhead pressure calculation is needed)

*INCOMP
    (Required if you have defined any injectors. Keyword follows *INJECTOR)

*OPERATE
    (Optional – a BHP constraint is defaulted if omitted)

*MONITOR
    (Monitoring constraints are optional)

To define the well location and perforations, use:

*GEOMETRY
    (Optional). It must precede a well completion keyword which is followed by subkeyword *GEO)

*PERF
    (Required)

These keywords must all reside in the Well Data section of your data set. It is possible to define a well, name it, and specify its group affiliation with a *WELL keyword at one time, specify its completions with *PERF at a later time, and finally define the well's type with *PRODUCER or *INJECTOR at a still later time and have the well become active

How to Shut In a Well and Reopen It

A well may be shut in explicitly any time after it has been defined in a *WELL statement; however, *OPEN and *AUTODRILL status can only be specified after a well's type has been defined with an *INJECTOR, *PRODUCER, or *CYCLPROD keyword. When a *WELL statement is given for a well, the well is initialized to the shut in state. When *INJECTOR or *PRODUCER is given for a well, that well is automatically opened.

After being fully defined (including perforations and well type), a well may be opened or assigned autodrill status at any time using a *TIME or *DATE keyword. A well may be shut in immediately after it has been defined with *WELL.

You may open a shut-in well any time after it has been shut in. Be sure to use its well number or name when you open or shut in a well.

Example:

```
*WELL 1 'WATER INJECTOR'
*WELL 2 'PRODUCER'
*WELL 3 'SOLVENT INJECTOR'

** All wells have the same geometry.
**            rad   geofac   wfrac   skin
*GEOMETRY *K  0.25   0.34     1.0    0.0

** Initially both injectors are shut in;
** only the producer is operational. True
** for the first two years.

*INJECTOR 1
*INCOMP *WATER
*OPERATE *MAX *STW 12000.0
*OPERATE *MAX *BHP 10000.0
*PERF    *GEO 1
** if  jf  kf  ff
   1   1   3   1.0

** Shut in well #1.
*SHUTIN 1

*PRODUCER 2
*OPERATE *MAX *STO 12000.0
*OPERATE *MIN *BHP 1000.0
*MONITOR *GOR 10000.0    *STOP
*MONITOR *WCUT 0.8330002 *STOP
*PERF    *GEO 2
** if  jf  kf  ff
   7   7   1   1.0
*INJECTOR 3
*INCOMP *SOLVENT
        0.78  0.19  0.03  0.0  0.0  0.0
*OPERATE *MAX *STG 1.2E7
*OPERATE *MAX *BHP 10000.0
```

```
*PERF    *GEO 3
** if  jf  kf  ff
   1   1   3   1.0
** Well 3 and well 1 are perforated in the
** same grid block.

** Shut in well #3.
*SHUTIN 3

*TIME 730.000
*OPEN 1    ** Open water injector

*TIME 1095.0

** In this case a water injector and a
** solvent injector are defined in the same
** block; shut in the open one first before
** opening the second one, which was shut in.

*SHUTIN 1  ** Shut in water injector.
*OPEN 3    ** Open solvent injector.

*TIME 1460.0
*SHUTIN 3  ** Shut in solvent injector
*OPEN 1    ** open water injector

*TIME 1825.0
*SHUTIN 1  ** Shut in water injector.
*OPEN 3   ** Open solvent injector.
```

Using the Wellbore Model

To indicate that the wellbore model is required, use:

*PWELLBORE for producers or *IWELLBORE for injectors immediately under a well type specification. It is an optional keyword and must be located in the Well Data section.

You may want to use the wellbore model whenever the wellhead pressure is required either for

a) information only; or

b) when you want to use the wellhead pressure as an operating constraint.

Example:

```
** The wellbore model is desired for well 2
** which is an injector.
*INJECTOR 2
*IWELLBORE
** wdepth  length  rough  whtemp  bhtemp  wr
    1100.   1100. .1365    200.3  490.0   .33
```

Operating and Monitoring Constraints

*OPERATE and *MONITOR indicate the constraints on a given well. If no operating constraint is entered for a well, a bottom hole pressure constraint of one atmosphere for producers and 10,000 atmospheres for injectors is defaulted. Monitored constraints are optional.

Each well introduces a new unknown variable Pbh, the bottomhole pressure, into the simulation, and a constraint equation is required to determine this variable.

The first operating constraint in a list of operating and monitoring constraints is the primary operating constraint. The simulator at first attempts to operate on this primary constraint and monitors the others in the list at the same time. If one of the monitored constraints is violated and *CONT has been used, then this constraint becomes the operating constraint.

If more than one operating constraint is violated, then the one on the top of the list is used:

PRODUCERS

For a producer you should operate

a) on a rate constraint (the primary operating constraint) and

b) on a minimum bottom hole pressure, or

c) on a minimum tubing head pressure.

If your producer is an oil well, pick an oil rate constraint. If your well produces gas, pick a gas rate constraint.

A subsequent constraint to use with a producer may be a minimum rate constraint.

Example:

```
*PRODUCER 1
*OPERATE *MAX *STO 12000.0 *CONT
*OPERATE *MIN *BHP  1500.0 *CONT
```

This example demonstrates:

a) the use of the oil rate as the primary constraint of this oil well and

b) the subsequent constraint of bottomhole pressure.

The action to be taken if a violation occurs is to continue and switch the operating constraint to the one that has just been violated. The *CONT is the default and need not be entered.

INJECTORS

For an injector, you would pick:

a) a maximum injection rate constraint for the primary operating constraint and

b) a maximum bottom hole pressure constraint, or

c) a maximum tubing head pressure constraint.

If you are injecting gas, choose a gas rate constraint. If it's a water injector, choose a water rate, etc.

Example:

```
*INJECTOR 2
*OPERATE *MAX *STW 10000.0 *STOP
*OPERATE *MAX *BHP  2250.0 *STOP
```

This example indicates:

a) the water rate for this water injector is the primary constraint and

b) the bottomhole pressure is a secondary operating constraint which will be monitored at the same time.

In both cases, if either are violated, the simulation will stop.

MONITORING CONSTRAINTS

The format of the monitoring constraint is similar to the keyword *OPERATE. Again, the action for the constraint on the top of the list will be taken when more than one constraint is violated at the same time.

It is highly recommended that you monitor GOR and water cut in a producer; this may prevent some problems during the run of your simulation job.

Example:

```
*PRODUCER 1
*OPERATE *MAX *OIL   1200.0 *CONT
*OPERATE *MIN *BHP   2500.0 *CONT
*MONITOR *MAX *GOR  15000.0 *STOP
*MONITOR *MIN *WCUT    60.0 *SHUTIN
```

If at one time step, both the GOR and WCUT constraints are violated, the simulation will be stopped.

Entering Well Indices

The well indices can be obtained by the following two approaches:

a) From the well geometry using a well model. Use the following keywords:
   *GEOMETRY and
   *PERF *GEO

b) Specified directly by the user using the keyword
   *PERF

These keywords must reside in the Well Data section.

*GEOMETRY specifies the necessary parameters to calculate the well indices internally. The well completion keyword, *PERF, contains the location of the well completion. If *GEO subkeyword is not present in *PERF, it also contains the user specified well index.

*PERF can be used for horizontal, deviated or vertical wells.

Example:

```
*WELL 1 '12-09-18-56'
*PERF 1
**  if  jf  kf   wi   setn
    1   1   2:4  1.24  1
** well index is specified to be 1.24

-or-

*WELL 1 '12-09-18-56'
**             rad  geofac  wfrac  skin
*GEOMETRY  *K .375  .2488    1.0   0.0
** The well completion keyword must follow
** the geometry keyword pertaining to well 1.
*PERF *GEO 1
**  if  jf  kf   ff   setn
    1   1   2:4  1.    1
```

Stopping a Simulation Run

You can use the keyword *STOP to terminate the simulation run. *STOP only appears once in your data set; it normally appears as the last line of the data set in the Well Data section. A stop time or stop date is always given with it.

Example:

```
*DATE 1998 09 08
*STOP
```

Alternately, the simulation is stopped if the first time step size after a well change, specified by *DTWELL, is set to be zero.

Example:

```
*DATE 1998 09 08
*DTWELL 0.0
```

Guidelines for Setting Up Well Data

The following guide assists you with using the WELL AND RECURRENT DATA section of this document. When entering the well data for the first time in your data set, the following information must be present in this order:

1. Either *TIME or *DATE is required.
2. Define a value for *DTWELL, which is the first time step size used immediately after the well is defined.
3. Identify all new groups using *GROUP. A hierarchy may be built using the *ATTACHTO keywords. This step is optional.
4. Identify all new wells using *WELL.
5. Indicate the well locations, geometries, or the well indices using *GEOMETRY and the well completion keyword *PERF. This may be entered at a time later than the one at which *WELL was entered.

6a. Define the type of a new well or a well with the keywords *PRODUCER, *INJECTOR, or *CYCLPROD. This may be done at a *TIME later than that at which the *PERF lines were entered for the well.

6b. Indicate the need for the wellbore model. You must have it if you choose *WHP as one of the operating constraints.

6c. Define injection fluid composition for injectors.

6d. Define the operating or monitored constraints for that well.

Steps 1, 2, and 4 through 6 MUST appear in any data set, although if 6d is not done a bottom-hole pressure operating constraint will be defaulted. Step 5 may follow step 6, but if so the PERF lines must be entered at the same *TIME as the well type information.

7. A well is defaulted as shut when *WELL is first entered for it and may be explicitly shut in after *WELL has defined the well.
8. Use *OPEN to reopen a previously shut-in well. *OPEN can only be entered for a well after the type has been defined with *PRODUCER, *INJECTOR, or *CYCLPROD.
9. Be aware that different keywords are required depending on what options you are using.

Subsequent well changes at different times or dates are done with steps 10 through 15:

10. Define new wells and use steps 1,4,5, and 6 before adjusting the parameters of existing wells.
11. You may alter the primary operating constraint of any well with *ALTER, once the well's type has been defined with *PRODUCER, *INJECTOR, or *CYCLPROD. Use with *TIME or *DATE.

Only one of each of the following keywords can appear with any one *TIME or *DATE. In other words you cannot have two sets of *AIMSET, two sets of *AIMWELL, etc. under one *TIME or *DATE.

12. You may adjust the implicitness/explicitness as required (*AIMSET, *AIMWELL).
13. You may adjust the I/O controls as required.
14. The keywords *DTWELL, *DTMAX, *DTMIN may also appear in subsequent well changes.
15. You MUST always terminate the simulation run with *STOP.

Assume you have a 10 x 10 x 12 Cartesian system. A section of the well data may look like this:

Example:

```
** Well data
*RUN
*DATE 1990 1 1
*DTWELL
   0.025
*WELL 1 'OIL PRODUCER'
*PRODUCER 1
** Operating and monitoring constraints for the
** producer only.
*OPERATE *MAX *STO 1000.00
*OPERATE *MIN *BHP 3000.00
*MONITOR *MAX *WCUT   0.40
**            rad   geofac  wfrac  skin
*GEOMETRY *K  0.25   0.34    1.0   0.0
*PERF *GEO 1
** if  jf  kf  ff
   1   1   3   1
*TIME 10.0
** Change the primary constraint value at 10.0
** days from 1000. to 100.
*ALTER 1
   100.0
** At 50.0 days, change further to 50.0
*TIME 50.0
*ALTER 1
  1000.0
** Define an injector. Also modify the primary
** operating constraint for the producer again.
*TIME 720.0
*INJECTOR 1
*INCOMP *WATER
*OPERATE *MAX *STW 12000.0
*OPERATE *MAX *BHP 10000.0
**            rad   geofac  wfrac  skin
*GEOMETRY *K  0.30   0.34    1.0   0.0
*PERF *GEO 1
** if  jf  kf  ff
   7   7   3   1
*ALTER 2
   100.0
** Stop at 10 years
*DATE 1990 12 31
*STOP
```

Recurrent Data from Other Sections

There are a few keywords from the other sections whose parameters may be changed during the course of a simulation run. These keywords must be positioned after a *DATE or *TIME keyword in the Well Data section. At a given time or date the older or existent parameters will be overridden by the new ones.

From the I/O control section, these keywords may be reassigned:

a) *WRST
b) *WPRN
c) *OUTPRN
d) *WSRF and
e) *OUTSRF *GRID

Note that only *OUTSRF *GRID may be altered but NOT *OUTSRF *WELL.

From the numerical methods control section, these keywords may be changed:

a) *DTMAX
b) *DTMIN

and all numerical section keywords except *SOLVER, *DPLANES and *TWOPTFLUX.

Parallel GEM

If you are running on a pc using launcher, then to run in parallel you may modify the simulator icon and check the box “Run simulator in multi-processor mode”, and for a two processor machine, enter 2 in the box following “Number of processors to use”. This is equivalent to using the command line options “-doms –parasol 2”.

The command line option “-doms” is equivalent to using “*DPLANES” in the numerical methods section.

The command line option “–parasol n” behavior has changed to being equivalent to using:

```
*SOLVER *PARASOL
*PPATTERN *AUTOPSLAB n
*PNPROSL   n
*PNTHRDS   n
```

If an n is specified which is greater than the logical number of CPU’s on the machine, then “*PPATTERN *AUTOPSLAB n” will be used along with “PNTHRDS m” where m is the minimum of n and the number of logical CPU’s on the machine.

Thus, if you wish to use keywords for the parallel options instead of the command line switches, the base set of keywords for running the simulator in parallel is:

```
*DPLANES
*SOLVER *PARASOL
*PPATTERN *AUTOPSLAB n
*PNPROSL   n
*PNTHRDS   n
```

Where n is the desired number of logical CPU’s to be used.

There are a number of techniques for improving solver performance in PARASOL runs.

Setting “PPATTERN 2” or “PPATTERN 3” for 2 or 4 logical CPU’s may work as well (in some cases better) than the corresponding “*PPATTERN *AUTOPSLAB n”.

Setting *CHECKRB to “*CHECKRB *ON” will avoid the use of the red-black system reduction when there are too few red cells to make the reduction worthwhile.

If there is a significant number of solver failures in the run, as reported at the end of the .log file, then increasing *NORTH and *ITERMAX often helps, such as

```
*ITERMAX 80
*NORTH     80
```

Another technique to improve solver convergence is to try a higher linear solver degree such as “*SEDGREE 2”

If you are specifying *SDEGREE greater than 1 (the default) then we recommend that you also use, in the Numerical Methods section:

```
*REORDER-CBDY *ON
```

This tends to reduce interclass induced fill.

The above are the techniques we have found most frequently to improve PARASOL's performance over an initial run with defaulted settings. While the full set of PARASOL controlling keywords allows more detailed control than contained in the above, these constitute a good starting point.

In order obtain timing diagnostics, you may put in the Input/Output Control Section:

```
*DEBUG *CPUTIME *SOLVER
```

This will provide a summary at the end of the log and out file giving a breakdown of elapsed ("wall clock") and CPU times for different portions of the run. Equivalently, this option may be enabled by using the command-line argument "-cputime"

The JACPAR Option

A parallel feature having parallel partition based on parasol pattern (JACPAR) is available in GEM version 2009.10 onwards. It uses connection-based cache level localization for Jacobian building and residual calculation. The feature may make the parallel processing in GEM more efficient especially on machines with larger number of CPUs in certain cases.

To invoke this feature, command-line option –jacpar –parasol n may be used, where n denotes the number of logical CPUs to be used.

If you wish to use keywords for parallel processing using parasol patterns for Jacobian building and residual calculations, the base set of keywords will become:

```
*JACPAR *ON
*SOLVER *PARASOL
*PPATTERN *AUTOPSLAB n
*PNPROSL   n
*PNTHRDS   n
```

Where n is the desired number of logical CPU's to be used. Unlike, *DPLANES (-doms), use of *SOLVER *PARASOL (-parasol) is a necessity for *JACPAR feature. For further details and exceptions, please see the documentation for *JACPAR under Numerical Methods Control chapter.

Please note that *DPLANES and *JACPAR are mutually exclusive, i.e., -doms and –jacpar options should not be used together.

By default, Launcher offers *DPLANES (-doms) option, so to use *JACPAR (–jacpar) via Launcher you would need to manually enter –jacpar –parasol n in the Additional simulator switches area on the Submit/Run a Simulation Job dialog box. Replace n by number of logical CPUs desired, and leave the number of processors on the top of the dialog box unchanged (1).

Saline Water Injection

GEM Version 2008.10 and beyond allows injection of aqueous components into the reservoir through use of keyword *INCOMP *AQUEOUS in the Well and Recurrent data section provided the model is set up to handle aqueous components. The definition of aqueous components is done in the Component Properties section and the specification of concentration of aqueous components is done in the Initialization section. Please refer to keywords *NC-AQUEOUS, *COMPNAME-AQUEOUS etc. in the Component Properties section and keyword *MOLALITY-AQUEOUS etc. in the Initialization section.

This tutorial section is intended to provide a simplified guide to injecting saline water when no chemical equilibrium or mineral reactions are involved in the model. A few alternate keywords in Initialization; and Wells and Recurrent sections can be used to simplify the task of data setup.

1. In the Component Properties section define one and only one aqueous component NaCl with keywords as under:

   *NC-AQUEOUS 1
   *COMPNAME-AQUEOUS 'NaCl'

2. For aqueous-phase density, and viscosity, calculations use the following input; also in the Component Properties section.

   *AQUEOUS-DENSITY *ROWE-CHOU
   *AQUEOUS-VISCOSITY *KESTIN
   *SALINITY-CALC *ON

3. Also have the derivatives calculated numerically by using the following keywords in the Component Properties section.

   *DERIVATIVEMETHOD *NUMERALL

4. In the initialization section specify the initial NaCl concentration in water as under, (for simplicity, assuming only one initialization region is present in the model). Initial salinity *SALINR could be specified in a variety of units as described in manual pages for *SALINR.

   *SALINR *PPMVOL 58000.

5. In the Wells and Recurrent data section input the salinity of injected water using keyword *INJ-SALIN as under.

   *INJ-SALIN *MOLAL ‘IW-1’ ‘IW-2’
   1.5 2.5

   Note that wells ‘IW-1’ and ‘IW-2’ in the example above should have been defined as <u>water injectors</u> and perforated and should be open before the *INJ-SALIN input. For the allowed units of salinity please refer to manual pages for *INJ-SALIN.

Use *MOLALITY ‘NaCl’ in the input/output section to write out molality of the aqueous component to the SR2 system if you wish to track the NaCl concentration in the reservoir with passage of time. Template file gmghg019.dat demonstrates saline water injection data setup for GEM.

Using the Oil Wet Option

Notes:

1. The oil wet option initialization assumes that the reservoir was initially water wet and has undergone a wettability alteration to become an oil wet reservoir.

   This assumption has a significant impact on how fluids are placed during reservoir initialization.

2. Within an Initialization region, all rock types should either be oil wet or water wet. If both oil wet and water wet rock types are found, unphysical results may occur. The DEPTH_AVE initialization option will warn the user if this occurs and stop the simulation.

3. With the DEPTH_AVE initialization the user has the choice of entering a value of the capillary pressure at the water oil contact with the keyword WOC_PC. If no value is entered by the user then the default for oil wet is to assume that the DWOC (depth to the water – oil contact) corresponds to the free water level. Consequently the minimum value of the capillary pressure in the SWT table will be assigned at WOC. For the oil wet option please note that the capillary pressure values in the SWT table as well as the value entered with WOC_PC keyword are interpreted to be the non-wetting phase (water) pressure minus the wetting phase pressure (oil) or

   $P_{cwo} = P_w - P_o$

In the case of BLOCK_CENTER COMP initialization, the user is not provided with the option of entering a capillary pressure corresponding to the oil water contact depth. GEM internally assigns a capillary pressure of zero at the WOC. The user should then enter a DWOC that corresponds to zero capillary pressure.

Recall that for the water wet case typically the DWOC corresponds to the minimum depth at which a water saturation of 100% is reached. For the water wet case transition zone extends from the WOC upwards. For the oil wet case typically transition zone will extend downwards from the WOC.

Oil Wet Initialization (DEPTH_AVE and BLOCK_CENTER COMP initializations):

If the reservoir is identified as oil wet (keyword RPT OILWET) then the user can initialize the model, that is specify the initial pressure and fluid global compositions / saturations in the reservoir in one of three ways: Using DEPTH_AVE, all sub options are available, using BLOCK_CENTER, COMP sub option only is supported, or using USER_INPUT.

DEPTH_AVE and BLOCK_CENTER COMP initializations (when used with the oil wet option) assume that the reservoir is initially water wet and has had its wettability altered (to oil wet). The capillary pressure between non-wetting and wetting phases still govern the transition zones, but the gas zone contains gas and water, the oil zone contains oil and water (both mobile and immobile) and the water zone contains water.

Unless specified explicitly, the WOC_PC (the capillary pressure at the water oil contact) is internally set equal to the minimum Pcwo (see Pcwo definition below), which occurs at the largest oil (wetting phase) saturation. Therefore the transition zone occurs between this depth towards the 100% water saturation depth.

A different WOC_PC can be entered for each Initialization region for DEPTH_AVE initialization.

This type of initialization produces results which are very similar to a water wet reservoir initialization which at a later date undergoes a wettability change to oil wet, and is then allowed to re-equilibrate.

Flow in an Oil Wet Model

In an oil wet model (regardless of initialization option), three phase flow is calculated assuming water is the intermediate phase. Therefore K_{ro} is a function of oil saturation, K_{rg} is a function of gas saturation, while K_{rw} is a function of both oil and gas saturation (through K_{rwo} and K_{rwg}).

The normal three phase flow models (e.g. Stone 1, Stone 2, segregated, etc.) are used to determine water relative permeability not oil permeability.

$$K_{rg} = f(S_g) \qquad K_{ro} = f(S_o) \qquad K_{rw} = f(S_o, S_g)$$

When modeling 3 phase flow for example using Stone's 2nd model we would obtain K_{rw} from:

$$K_{rw} = K_{rwco} * \left[\left(\frac{K_{rwo}}{K_{rwco}} + K_{ro} \right) * \left(\frac{K_{rwg}}{K_{rwco}} + K_{rg} \right) - K_{ro} - K_{rg} \right]$$

Where K_{rwco} is the water relative permeability at connate oil, K_{rwo} is the water relative permeability obtained from the water-oil table (S_o+S_w=1) and K_{rwg} is the water relative permeability obtained from the liquid-gas table (S_l+S_g=1, S_o=connate oil). This approach for oil wet rock is a direct extension of Stone's model which originally developed for water wet systems.

Relative Permeability/Capillary Pressure Curves in an Oil Wet Model

When the oil wet option is used the meanings of the columns in the relative permeability tables are altered. Normally water is the wetting phase and oil is the non-wetting liquid phase.

SWT Table

When the oil wet option is active the column which normally contains water saturation (the first) should now contain the saturation of the wetting phase (oil). The K_{rw} column (2nd column) should contain the wetting phase relative permeability (oil relative permeability). The K_{row} column (3rd column) contains the non-wetting phase relative permeability (water relative permeability).

The fourth and fifth columns which normally (for water wet models) contain the P_{cow} and P_{cowi} now contain P_{cwo} and P_{cwoi}. These represent the positive capillary pressure between the non-wetting water phase and the wetting oil phase. Since the tables still tabulate non-wetting phase pressure minus wetting phase pressure versus wetting phase saturation (So), the shape of the P_{cwo} curves resembles that of the Pcow curves versus water saturation in a water wet system.

$$p_{cwo} = f(S_o) = p_w - p_o \text{ and } p_{cow} = f(S_w) = p_o - p_w$$

SLT Table

The meanings of the columns of the gas-liquid tables are not altered. However the liquid saturation in the table is made up of water and connate oil. In the DEPTH_AVE / BLOCK_CENTER COMP initialization option the P_c curve entered in the liquid gas table must be the P_{cog} curve (even though oil is not the intermediate phase.

If the capillary pressure received is from a water-gas system, $p_{cgw} = p_g - p_w$, to accommodate the GEM table input for DEPTH_AVE / BLOCK_CENTER COMP initialization, the user needs to convert p_{cgw} to p_{cgo} using $p_{cgo} = p_{cgw} + p_{cwo}$.

Endpoint Arrays in an Oil Wet Model

The use of the oil wet option also changes the meaning of the user specified grid block specific end point arrays. Unless these changes in definition are accounted for, the unscaled relative permeability curves will be incorrect.

The normal definitions of the affected arrays are.

S_{wcon}	=	Connate water saturation
S_{wcrit}	=	Critical water saturation
S_{oirw}	=	Irreducible oil saturation
S_{orw}	=	Residual oil saturation
S_{org}	=	Residual oil saturation (gas liquid table)
S_{lcon}	=	Connate liquid saturation which is equal to connate water (S_{wcon}) plus irreducible oil saturation (S_{oirg})

When using the oil wet option they are modified to be.

S_{wcon}	=	Connate wetting phase (oil) saturation
S_{wcrit}	=	Critical wetting phase (oil) saturation
S_{oirw}	=	Irreducible non-wetting phase (water) saturation
S_{orw}	=	Residual non-wetting phase (water) saturation
S_{org}	=	Residual non-wetting phase (water) saturation (gas liquid table)
S_{lcon}	=	Connate liquid saturation which is equal to connate oil (S_{wcon}) plus irreducible water saturation (S_{oirg})

The scaling ARRAYS *KRWIRO, *KROCW, *KROGCG, *PCWMAX, *PCGMAX,

*JFWMAX and *JFGMAX must also be used with altered definitions if the oil wet option are used.

The normal definitions of these arrays are:

K_{rwiro}	=	Water rel. perm. at irreducible oil (oil-water table)
K_{rocw}	=	Oil rel. perm at connate water (oil-water table)
K_{rogcg}	=	Oil rel. perm at connate gas (liq.-gas table)
Pc_{wmax}	=	Oil water capillary pressure at connate water
P_{cgmax}	=	Gas oil capillary pressure at connate liquid
J_{fwmax}	=	Oil water J function at connate water
J_{fgmax}	=	Gas oil J function at connate liquid

When using the oil wet option they are modified to mean.

K_{rwiro}	=	Wetting Phase (Oil) rel. perm. at irreducible water (oil-water table)
K_{rocw}	=	Non-wetting Phase (water) rel. perm at connate oil (oil-water table)
K_{rogcg}	=	Non-wetting phase (water) rel. perm at connate gas (liq.-gas table)
Pc_{wmax}	=	Water-oil capillary pressure at connate oil
Pc_{gmax}	=	Gas oil capillary pressure at connate liquid (DEPTH_AVE)
J_{fwmax}	=	Water oil J function at connate oil
J_{fgmax}	=	Gas oil J function at connate liquid (DEPTH_AVE)

Example: Oil Wet Rock Type Curves (Relative Permeability)

Figure 1

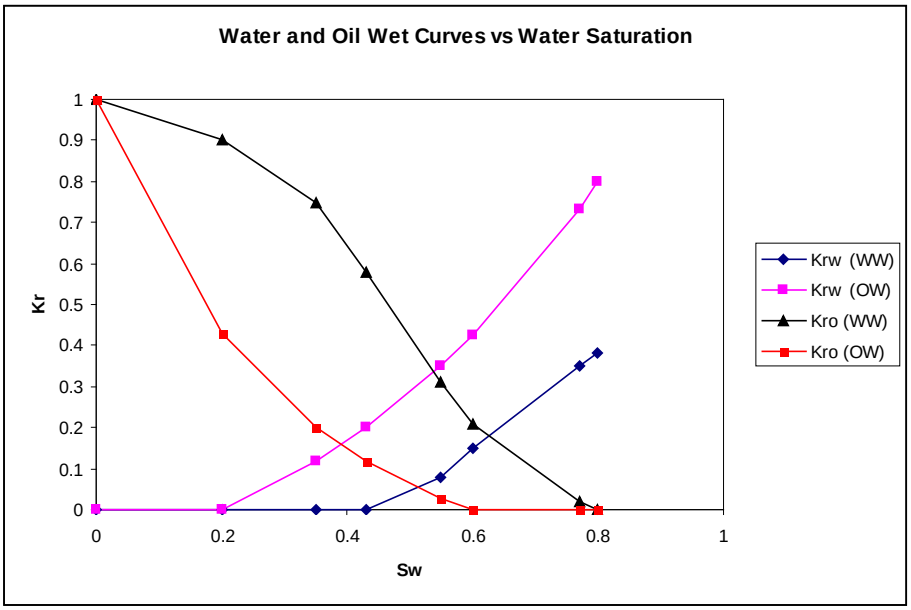
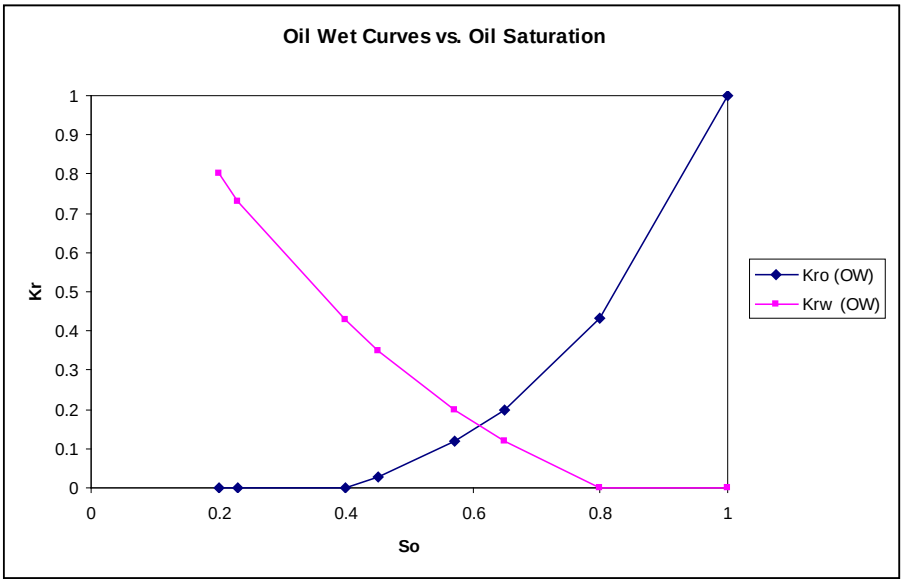


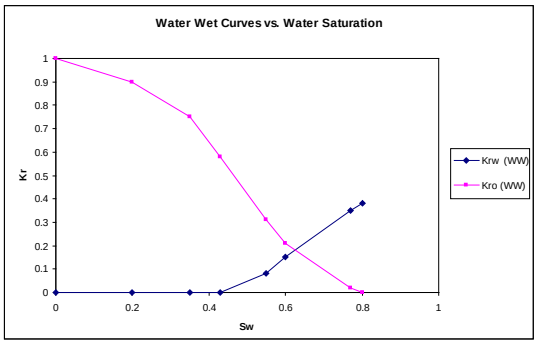
Figure 2

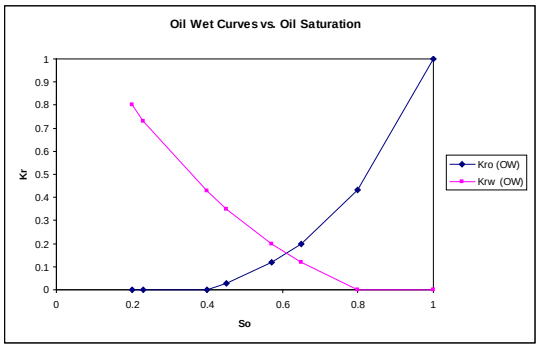


Oil wet curves are extracted from figure 1 and plotted against Oil saturation which for an oil wet case is the wetting saturation (figure 2). The curves in figure 2 must be entered as relative permeabilities in an oil wet rock type. In column 1, oil (wetting phase) saturation is entered instead of water saturation. In column 2 relative permeability to oil (wetting phase) is entered instead of relative permeability to water and in column 3, relative permeability to water (non-wetting phase) is entered instead of relative permeability to oil.

It is interesting to note that the relative permeability curves for the water wet and oil wet curves in Figure 1 are in good agreement if they are plotted vs. wetting phase (water for water wet and oil for oil wet). The figure below presents this.

Figure 3

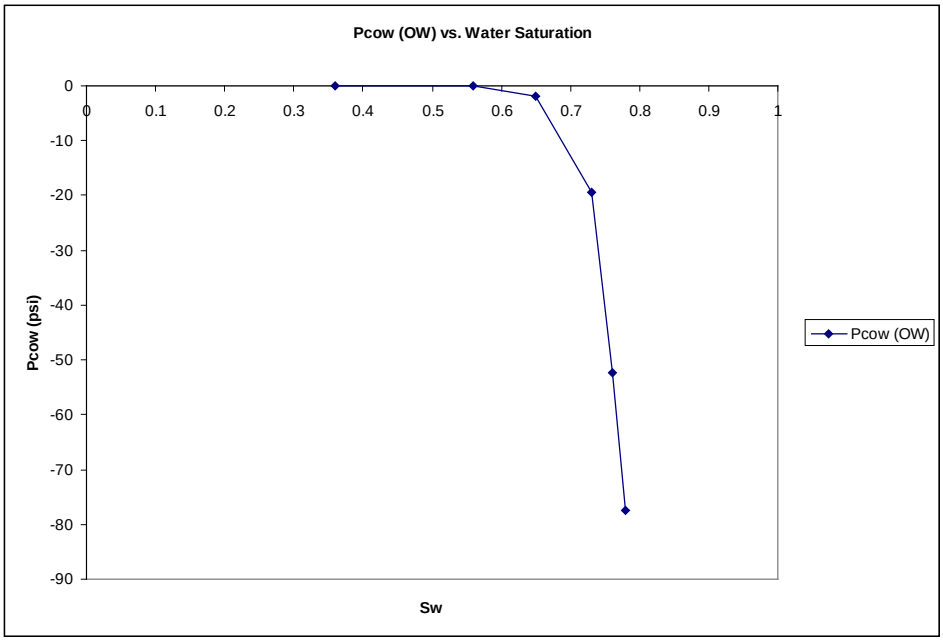




Example: Oil Wet Rock Type Curves (Capillary Pressure)

If the capillary pressure received are in terms of Pcow (water phase pressure minus oil phase pressure) in an oil wet system versus water saturation (the non-wetting phase) the expected capillary pressure curve would resemble Figure 4 below.

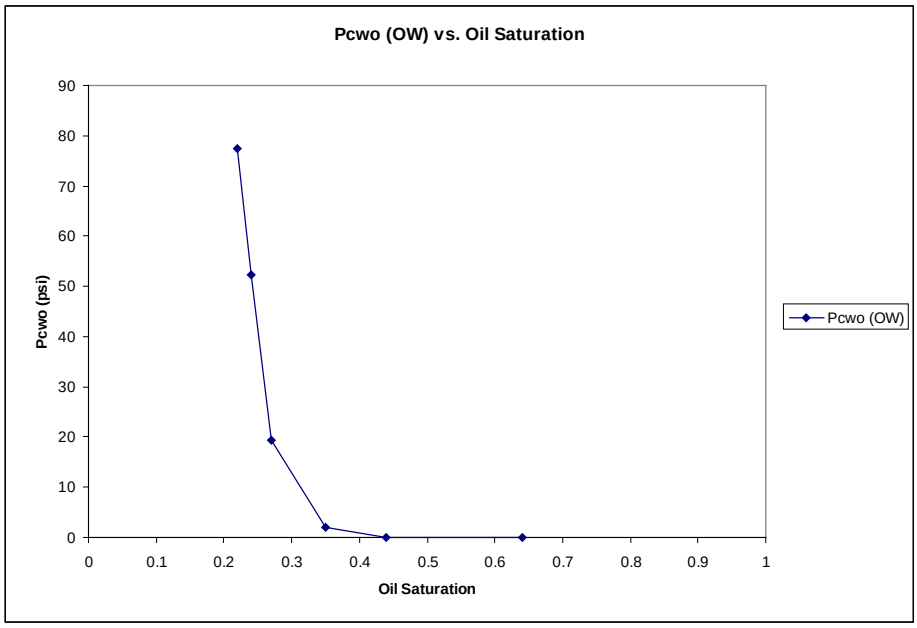
Figure 4



To use this data in GEM, it is necessary to translate the Pcow capillary pressure into the P_{cwo} curve (non-wetting phase pressure minus wetting phase pressure) that the oil wet model requires as a function of wetting (oil) phase saturation.

For example $p_{cwo} = - p_{cow}$ and $S_o = 1 - S_w$

The resulting curve is shown below. This curve is similar in shape to p_{cow} versus water saturation in a water wet system.



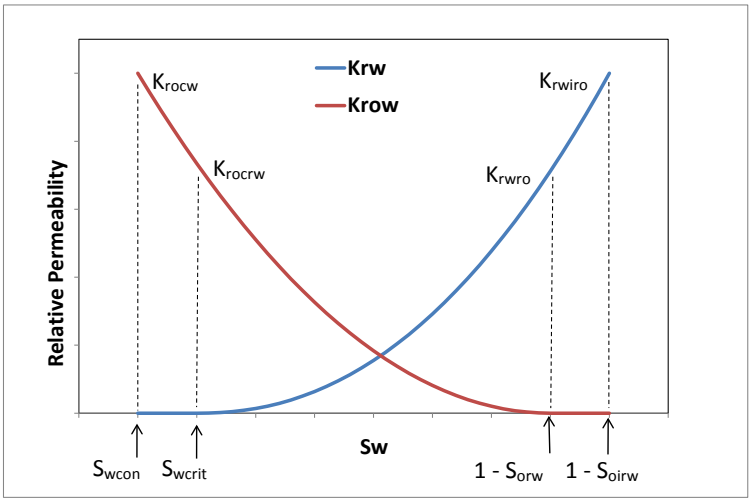
The relative permeability and capillary pressure examples are from Anderson, W.G. (JPT Oct. and Nov. 1987).

Oil Wet Option and Hysteresis

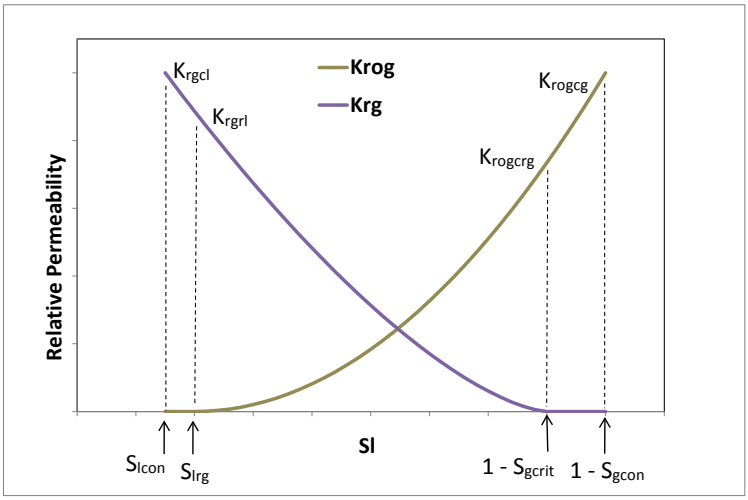
Due to the fact that the oil wet option rock type tables are tabulated against wetting phase saturation, all of the hysteresis options in GEM function. P_{cwo} hysteresis and non-wetting phase relative permeability hysteresis are modelled. It must be remembered that in the oilwet model the non-wetting liquid saturation refers to water so HYSKRO 'somax' refers to a water residual saturation and not an oil residual saturation.

Two and Three Point Scaling of Relative Permeability Values

Relative permeability values (heights) at the endpoints can optionally be assigned to grid blocks. Relative permeability values at corresponding saturation endpoints for Krw, Krow, Krog and Krg curves may be characterized by the endpoints illustrated in the next two graphs.



Oil-Water Relative Permeability



Liquid-Gas Relative Permeability

There are eight keywords available to define these Kr values at saturation endpoints for each grid block. Four of them are maximum Kr definitions, *KRWIRO, *KROCW, *KOGCG and *KROGCL. The remaining four keywords are *KRWRO, *KROCRW, *KROGCRG and *KRGRL which specify Kr at critical (or residual) saturations. These keywords can be assigned in both Rock-Fluid Data and Recurrent Data sections.

Due to the similarity of the Kr curves, in the following, we will just discuss the water relative permeability Krw to describe the scaling method. Other relative permeability values are scaled in a similar fashion.

There are two types of Krw value scaling, which depend on how endpoints are defined for a grid block.

a) Two-Point Scaling

A water relative permeability curve, Krw, always has two endpoints, one at each end of the curve, i.e. Krw at critical water saturation (Krw@Swcrit= 0) and Krw at irreducible oil (Krwiro = Krw@Soirw). Thus if a grid block is only assigned a Krwiro value (using *KRWIRO), the block Krw will be scaled using two-point scaling:

$$K_{rw}(b) = K_{rw}(t)\frac{K_{rwiro}(b)}{K_{rwiro}(t)}$$

Note that (b) and (t) refer to the block value and table value respectively, where the table value, (t), is the value read off of the Krw table for the rock type which is associated with the grid block. The block value, (b), is the value read from the *KRWIRO keyword.

b) Three-Point Scaling

If a grid block is assigned Krwro(b), using the *KRWRO keyword AND the corresponding oil-water Kr table contains a residual oil saturation endpoint, Sorw(t) which is different from Soirwt(t), the block Krw value will be scaled using three-point scaling, where:

$$K_{rw}(b) = K_{rw}(t)\frac{K_{rwro}(b)}{K_{rwro}(t)}, \text{ for } S_w \le 1 - S_{orw}$$

or

$$K_{rw}(b) = K_{rwro}(b) + \left(K_{rw}(t) - K_{rwro}(t)\right)\frac{\left(K_{rwiro}(b) - K_{rwro}(b)\right)}{\left(K_{rwiro}(t) - K_{rwro}(t)\right)}, \text{ for } S_w > 1 - S_{orw}$$

If the table's Krw curve is flat between 1-Sorw and 1-Soirw, which means the term (Krwro(t) – Krwiro(t)) is close to zero, an alternate linear interpolation will be applied when Sw > 1-Sorw:

$$K_{rw}(b) = K_{rwro}(b) + \left(S_w(t) - 1 + S_{orw}(t)\right)\frac{\left(K_{rwiro}(b) - K_{rwro}(b)\right)}{\left(S_{orw}(t) - S_{oirw}(t)\right)}$$

Where Sw(t) represents the water saturation of the table that is scaled from the block water saturation Sw.

If a grid block is assigned Krwro(b), using the keyword *KRWRO keyword AND the corresponding oil-water Kr table contains a residual oil saturation endpoint, Sorw(t) which is the SAME as the irreducible oil saturation Soirw(t), and the block residual oil saturation Sorw(b) is different from the block irreducible oil saturation, Soirw(b), the block Krw value will be scaled using three-point scaling, where:

$$K_{rw}(b) = K_{rw}(t)\frac{K_{rwro}(b)}{K_{rwro}(t)} \text{ for } S_w(b) \le 1 - S_{orw}(b)$$

or

$$K_{rw}(b) = K_{rwro}(b) + \frac{S_w(b) - (1 - S_{orw}(b))}{S_{orw}(b) - S_{oirw}(b)} * \left[K_{rwiro}(b) - K_{rwro}(b)\right]$$

For $S_w(b) > 1 - S_{orw}(b)$

The equations for calculations corresponding to Krow, Krog and Krg are developed following a similar logic to that of Krw.

If a block's maximum Kr endpoint is unassigned, the corresponding value of the table will be used as the default. However, for a grid block, an unspecified relative permeability value at a critical endpoint simply means the block does not use three-point relative permeability value scaling.

To enable the three-point Kr value scaling in a specific block for:

Krw:

*KRWRO array value for that block must be defined. In addition in the *SWT table associated with that block, the table value of Sorw(t) must be different from the table value of Soirw(t) OR the grid block value of Sorw(b) entered via keyword *SORW must be different from the grid block value of Soirw(b) entered via keyword *SOIRW.

Krow:

*KROCRW array value for that block must be defined. In addition in the *SWT table associated with that block, the table value of Swcrit(t) must be different from the table value of Swcon(t) OR the grid block value of Swcrit(b) entered via keyword *SWCRIT is different from the grid block value of Swcon(b) entered via keyword *SWCON.

Krog:

*KROGCRG array value for that block must be defined. In addition in the *SLT/SGT table associated with that block, the table value of Sgcrit(t) must be different from the table value of Sgcon(t) OR the grid block value of Sgcrit(b) entered via keyword *SGCRIT is different from the grid block value of Sgcon(b) entered via keyword *SGCON.

Krg:

*KRGRL array value for that block must be defined. In addition in the *SLT/SGT table associated with that block, the table value of Sorg(t) must be different from the table value of Soirg(t) OR the grid block value

If the above is not true for a particular curve, three-point Kr value scaling will not be performed for that curve.

Three-point relative permeability vertical scaling is independent of three-point saturation scaling as defined by the *3PTSCALING keyword. Using the three-point saturation scaling option (via keyword *3PTSCALING) does not turn on the three-point relative permeability endpoint value (vertical) scaling option. Only defining the critical relative permeability endpoints (as discussed above) activates this option. If a critical (vertical) Kr endpoint is assigned to a grid block, for example *KRWRO, then both horizontal (saturation) and vertical (relative permeability) will be scaled based on three point scaling for that particular curve, in this example Krw (water curve), even if the *3PTSCALING keyword is not specified.

Modeling Flow in Fractured Shale Gas Reservoirs

This tutorial discusses the modeling of flow in hydraulically fractured shale reservoirs by:

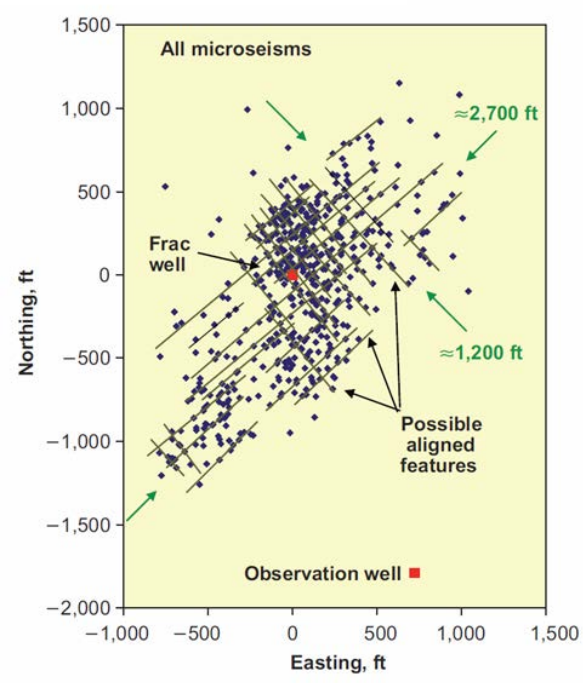
- Using a dual-permeability grid to model flow into networks of fractures connected to vertical or horizontal wells.
- Representing fractures explicitly using multi-foot-thick pseudo-fractures with corresponding pseudoization of Darcy flow and the Forchheimer number to support reasonable prediction of pressure drops in the fracture.
- Modelling the region around fractures with a finely gridded LGR to accurately reproduce the pressure drop around the fracture.

Background

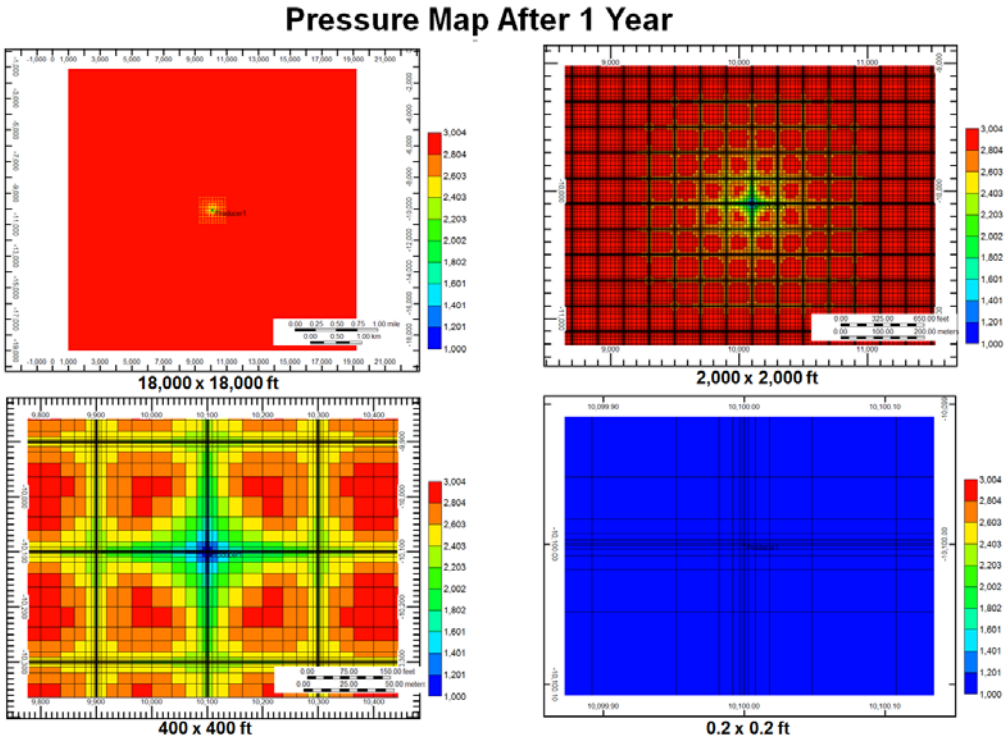
For practical exploitation, unconventional shale gas reservoirs require hydraulic fracturing of existing fracture networks to create a stimulated reservoir volume (SRV).

Papers have discussed the modeling of gas production from unconventional shale gas reservoirs using numerical simulation of explicit fracture networks within the SRV. The simplifications made to these models produced reasonable results but still relied on actual fracture widths (on the order of 0.001 ft.) to account for non-Darcy flow. This made simulation onerous because, with explicit fracture representation, models used large detailed grids and required significant execution times, due to both the large number and small size of the fracture cells.

CMG's objective was to develop a fractured shale gas simulation model that is easy to set up, reasonably accurate, and which runs in minutes rather than hours or days. We based our investigation on microseismic data from fractured wells:



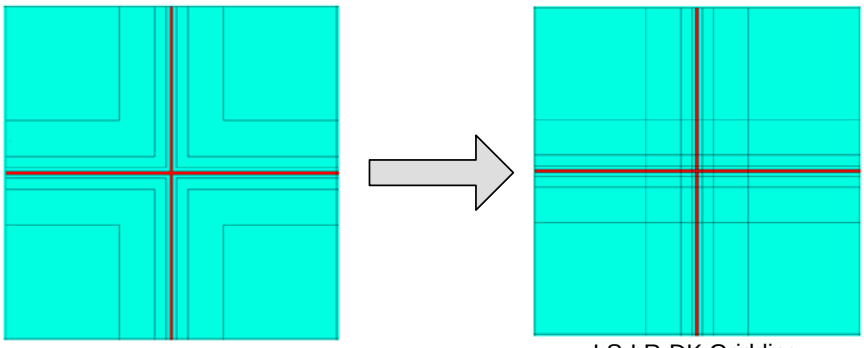
Using this data, we created a finely-gridded single-well reference solution (with 6-14 million cells) to simulate Darcy and non-Darcy flow within an explicitly modeled SRV complex fracture network. The following figure shows the grid used in the reference model, at progressively finer and finer resolutions:



Our investigation was based on 2-D models, with and without primary hydraulic fractures, and included scenarios that modeled stress-sensitive permeability and subsequent re-stimulation of horizontal wells. The reference model produced accurate flow predictions using cells that were sized small enough (0.001 ft. wide) to properly capture the large pressure drops involved.

The reference solution was compared with dual-permeability (DK, normally used when simulating naturally fractured reservoirs) and multiple interacting continua (MINC, developed to account for transient flow from matrix to fracture) dual-continua models. Our findings confirmed that dual-permeability models do not have the resolution to accurately model transient matrix-to-fracture flow in low-permeability fractured shales. Because MINC models fractures using one grid cell/block, it cannot accurately model pressure drops within fractures. Although this may be satisfactory for high-conductivity fractures, it is not acceptable for the low-conductivity fractures experienced in shale reservoirs. As well, because MINC does not support direct matrix-to-matrix flow, it cannot be used to model high-permeability shales.

As a result of these limitations, and encouraged by the MINC results, CMG proposed a new modeling technique for simulating and coupling flow inside the SRV using coarse, logarithmically spaced, locally refined, dual-permeability (LS-LR-DK) grids. Flow outside of the SRV was modeled using unrefined dual-permeability grids. The following figure compares the gridding used in the MINC models with that used in the LS-LR-DK models:

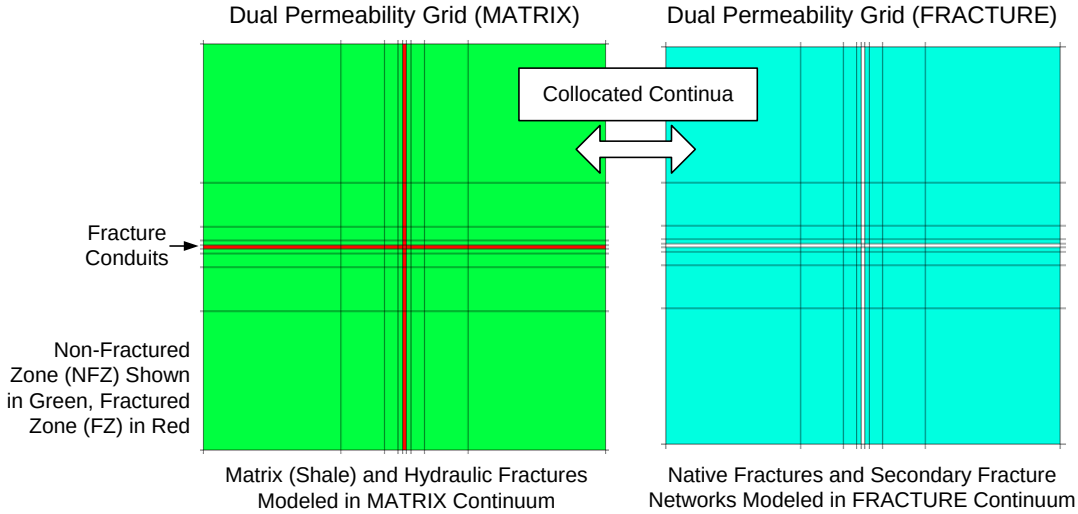


MINC Gridding LS-LR-DK Gridding

As shown above, the gridding used in the LS-LR-DK model, which is somewhat similar to that used in the MINC models, uses logarithmic cell spacing to accurately model flow within the fracture, within the shale, and from shale to fracture. Logarithmic grid spacing best represents the large pressure drops near the matrix-fracture interface, and reduces computation requirements further from the interface. Our decision to use local logarithmic spacing was based on the assumption that outside the SRV, it would not be necessary to model flow as accurately as it needs to be modeled within the SRV; in fact, outside the SRV, a simple DK grid was used.

We used the standard effective permeability calculation used by DK models; however, to reduce computation, we scaled the width of the fracture cells up to 2.0 ft. (which effectively converts them to a fracture "conduit") then scaled the permeability (K_{feff}) to maintain the conductivity of 0.001 ft. cells with permeability K_f. As well, since K_{feff} is also used in the Forchheimer equation, a scaling factor was used to correct the Forchheimer number when K_{feff} is used instead of K_f.

CMG's LS-LR-DK model is based on two collocated fracture networks with fluid transfer between them—a MATRIX continuum to capture matrix (shale) and explicit hydraulic fracture permeability, and a FRACTURE continuum to capture naturally fractured shale permeability. This is shown below:



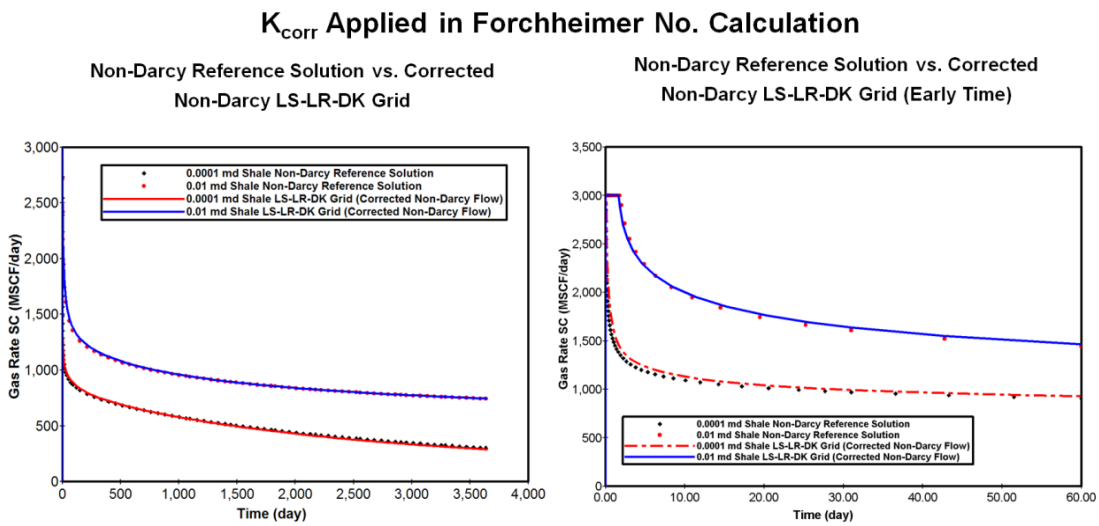
As shown above, the fracture conduit or fractured zone (FZ), is defined by the single row and column shown in red.

Dual-permeability FRACTURE cells are used to model native, un-stimulated fractures, as well as other secondary fracture networks, such as localized regions of intense fracturing. Fluid in the dual-permeability MATRIX continuum flows along the explicitly modeled fracture conduit, adjacent MATRIX (shale) cells, and collocated dual-permeability cells.

Within the MATRIX continuum, properties can be assigned to the fractured zone (FZ, which forms the fracture conduit) and the non-fractured zone (NFZ).

The CMG LS-LR-DK model used in our investigation runs in minutes using standard hardware, unlike the reference model, which can take days. The results, which are summarized in SPE 132093 *Accurate Simulation of Non-Darcy Flow in Stimulated Fractured Shale Reservoirs* by B. Rubin, Computer Modelling Group Ltd., demonstrate excellent matches with reference solutions, using a modest number of refinements to simulate flow within the SRV when the fracture permeability and the fracture Forchheimer number (for non-Darcy flow) are scaled appropriately.

To illustrate, the following simulation results compare predicted gas rate for the reference model with that for the LS-LR-DK model for extended and early times, once the Forchheimer correction was applied:



From our investigations, CMG concluded the following:

- An LGR-based, logarithmically spaced, dual-permeability (LS-LR-DK) grid approach accurately models shale-gas reservoirs using orders of magnitude less CPU time than finely gridded equivalent models.
- A flow permeability correction factor (K_{corr}) allows non-Darcy flow to be accurately modeled in LS-LR-DK grids, using pseudo-fractures a thousand times wider that the actual fracture sizes.
- Using LS-LR-DK grids and K_{corr}, the CMG model accurately matches the reference solutions for flow in fractured shale gas reservoirs for both vertical and

horizontal wells. From this, we concluded that this approach can be used with a high degree of confidence.

- An even finer 13-million-cell reference grid produced results that were not significantly different from those produced by the 6-million-cell reference solution gird, confirming that resolution is most critical in the shale near the fracture-shale interface.

The remaining sections in this tutorial demonstrate the entry and configuration of LS-LR-DK fractures in GEM datasets.

Fracture Definition

GEM supports the definition of two types of hydraulic fractures:

- Complex fractures can be irregularly distributed and defined by manually selecting the fractured blocks, or based on microseismic data.
- Planar fractures, also referred to as bi-wing fractures, typically extend from blocks containing perforations and follow I or J layers.

The examples used in this section are from GEM dataset *gmfrr008.dat*, a dual-permeability gas-water model for testing complex and planar fractures.

Defining Complex Fractures

In GEM, *BLOCKGROUP and *REFINE INNERWIDTH keywords are used to define SRV (stimulated reservoir volume) regions that contain complex fractures. *BLOCKGROUP is first used to define the blocks that belong to the complex-fracture SRV, as shown in the following example:

```
*BLOCKGROUP 'Well-1 - SRV 1_1'
*IJK 19     21    1    1
     7    22:24   1    1
     17     22    1    1
     19:20  22    1    1
     16:18  23    1    1
            . . .
            . . .
            . . .
     17     38    3    1
     12:13  39    3    1
     17     40    3    1
     21     42    3    1
```

In the above, block group 'Well-1 – SRV 1_1' is created and IJK blocks are assigned to it.

Once the block group is defined, *REFINE *INNERWIDTH is used to place LS-LR-DK grids in each cell of the block group, for example:

```
*REFINE *INNERWIDTH 0.6096
     *BG 'Well-1 - SRV 1_1' INTO 5 5 1 K2INT 500 WF2 0.0005
```

The above keywords place a 5×5×1 LS-LR-DK grid in each cell of block group 'Well-1 – SRV 1_1'. The innermost fracture conduit row/column, tagged as the fracture zone (FZ), is assigned an effective width of 0.6096 m (2 ft.), from which the effective fracture permeability $K_{fraceff}$ is calculated, as follows:

$$K_{fraceff} = K_{frac} \times \frac{Width_{frac}}{Width_{lgrblock}}$$

where:

K_{frac}	Intrinsic fracture permeability
$Width_{frac}$	Intrinsic fracture width
$Width_{lgrblock}$	Effective fracture width, defined by *INNERWIDTH

*K2INT and *WF2 specify the intrinsic permeability (500 mD) and fracture width (0.0005 m) of the secondary fracture network within the SRV, respectively. Based on these settings, the model calculates the effective dual-permeability secondary fracture properties.

Using *POR, the porosity of the non-fractured zones of the MATRIX and FRACTURE continua are defined, as shown in the following example, where we define the matrix (shale) porosity everywhere at 3%, and the fraction of the FRACTURE continuum that is void space as 0.008%:

```
POR *MATRIX CON         0.03
POR *FRACTURE CON       8E-005
```

The permeability of the MATRIX (shale) and FRACTURE continua are defined as follows:

```
PERMI *MATRIX CON       0.0001
PERMI *FRACTURE CON     4E-005
PERMJ *MATRIX CON       0.0001
PERMJ *FRACTURE CON     4E-005
PERMK *MATRIX CON       0.0001
PERMK *FRACTURE CON     4E-005

*PERMI MATRIX BG 'Well-1 - SRV 1_1' *FZ 0.410105
*PERMJ MATRIX BG 'Well-1 - SRV 1_1' *FZ 0.410105
*PERMK MATRIX BG 'Well-1 - SRV 1_1' *FZ 0.410105
```

As shown above, PERMI/J/K are defined generally for the MATRIX and FRACTURE continua then specifically for the fractured zone (FZ) in the block group defining the complex fracture.

*DIFRAC, *DJFRAC, and *DKFRAC keywords are used to define the spacing of the natural fractures in the FRACTURE continuum, and in the NFZ (non-fractured zone) of the MATRIX continuum, for example:

```
DIFRAC CON   10
DJFRAC CON   10
DKFRAC CON    0
```

In the above example, natural fracture planes are spaced 10 m perpendicular to the I and J axes, with no fracture planes perpendicular to the K axis.

In *mxfrr022.dat*, the non-Darcy correction factor is specified in the FZ zone of the MATRIX continuum, and set to zero in the FRACTURE continuum and in the NFZ zone of the MATRIX continuum, as shown below?

```
NONDARCY GENERAL 0.5
1.485e9     1.021 0
1.485e9     1.021 0
```

```
NDARCYCOR FRACTURE CON   0
NDARCYCOR MATRIX   CON   0
NDARCYCOR MATRIX BG 'Well-1 - SRV 1_1' NFZ   0.0
NDARCYCOR MATRIX BG 'Well-1 - SRV 1_1' FZ  270.30
```

The value of the non-Darcy correction factor is calculated as follows:

$$\text{NDARCYCOR} = \left(\frac{\text{Width}_{\text{lgrblock}}}{\text{Width}_{\text{frac}}}\right)^{(2-\text{N1g})}$$

where:

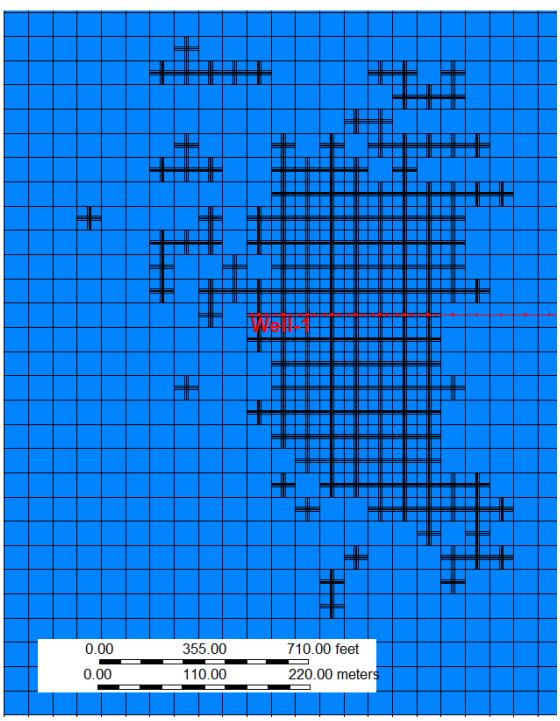
$\text{Width}_{\text{frac}}$	Intrinsic fracture width
$\text{Width}_{\text{lgrblock}}$	Effective fracture width, defined by *INNERWIDTH
N1g	Exponent of the ($K_{rg} \times K$) term in the β factor correlation for the specified model. Our example uses the *NONDARCY *GENERAL model, in which case N1g = 1.021

Substituting for the above parameters, the non-Darcy correction factor for our example is:

$$\text{NDARCYCOR} = \left(\frac{0.6096}{0.0020}\right)^{(2-1.021)} = 270.30$$

In the above formula, *0.002* is the assumed intrinsic fracture width. Once calculated, the NDARCYCOR value is manually entered in the dataset.

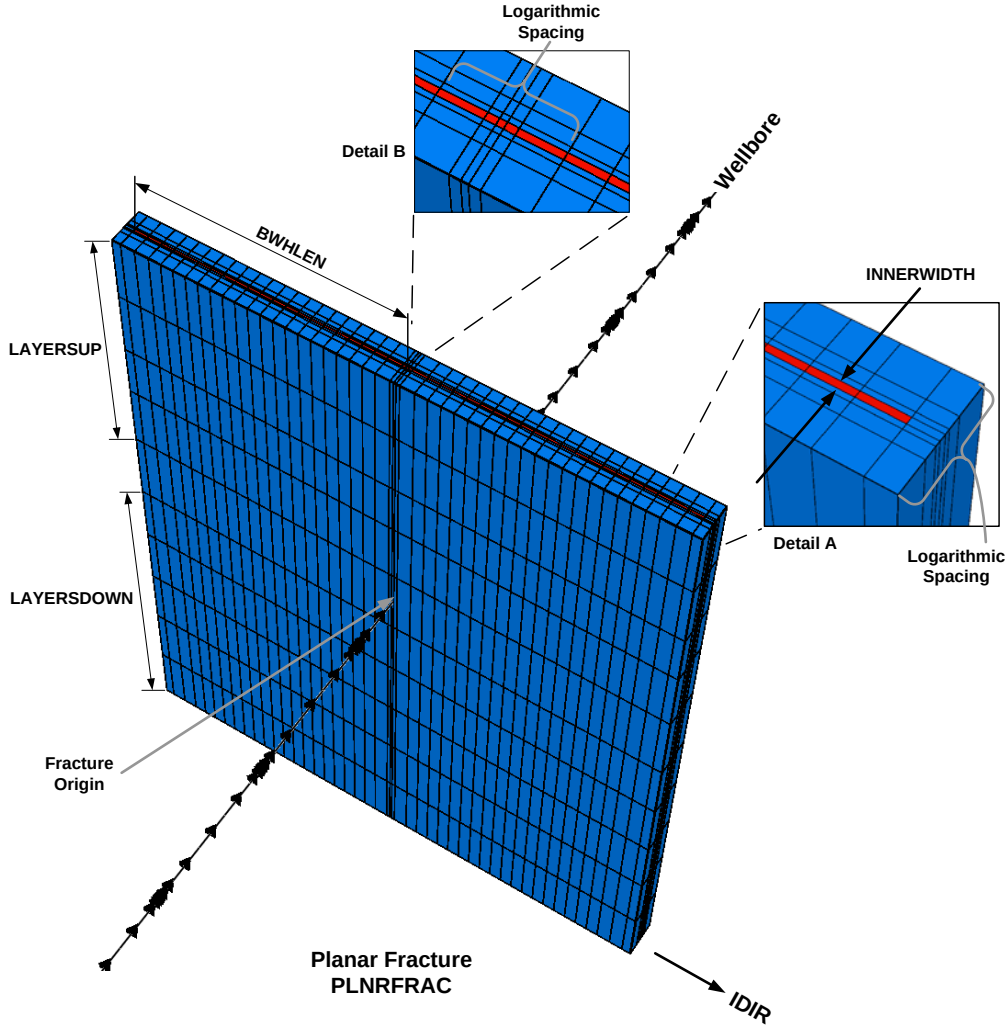
As necessary, other properties can be assigned to the block group defining the fracture. Once the complex fracture is entered, it can be viewed by opening the dataset in Builder:



Shows the layer of the complex fracture that contains the horizontal wellbore.

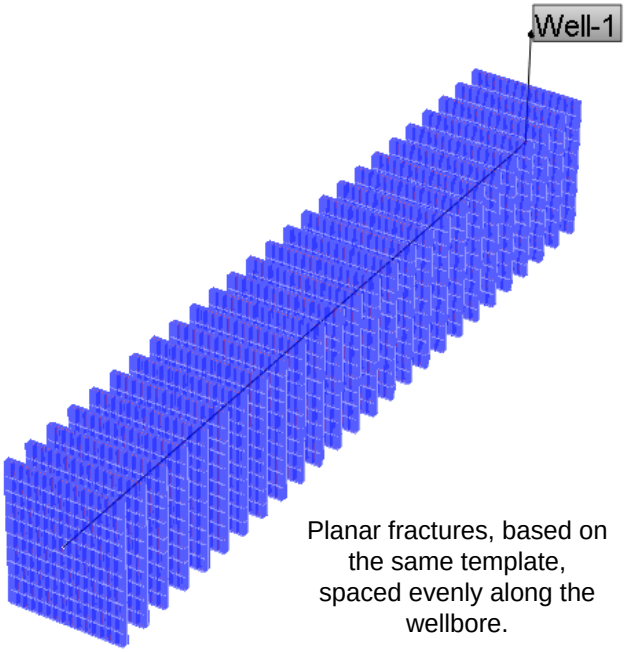
Defining Planar Fractures

A planar fracture is a slab of fractured blocks that aligns with the reservoir axes, and originates in a perforated block, as shown in the following example:



Planar fractures can be defined using *PLNR_REFINE to specify all of the fracture's properties.

Planar fractures can also be defined using templates (created using *PLNFRAC_TEMPLATE) that specify general fracture properties—primary and secondary permeability, orientation, size, and grid refinement—then applying these template (using keyword *PLNRFRAC) to multiple fractures, which may differ only with respect to their origin; for example, a series of similar fractures along a wellbore can share the same template:

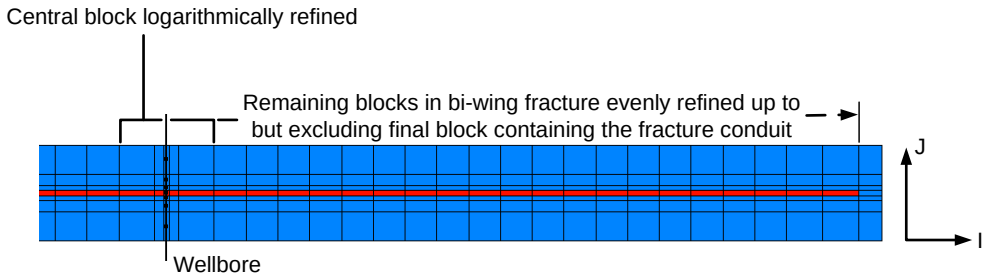


In this tutorial we will provide information about using templates to define planar fractures, however, much of the information provided is applicable to the use of *PLNR_REFINE to create fractures without using templates.

In the dataset associated with the above images, fracture template 'Template (Well-1 – Frac 1)' is defined as follows:

```
*PLNRFRAC_TEMPLATE 'Template (Well-1 - Frac 1)'
   *PLNR_REFINE *INTO  5 7 1
      *BWHLEN 72
      *IDIR
      *INNERWIDTH 0.6096
      *LAYERSUP 4
      *LAYERSDOWN 5
   *PERMI MATRIX *FZ 99.6145
   *PERMJ MATRIX *FZ 99.6145
   *PERMK MATRIX *FZ 99.6145
*END_TEMPLATE
```

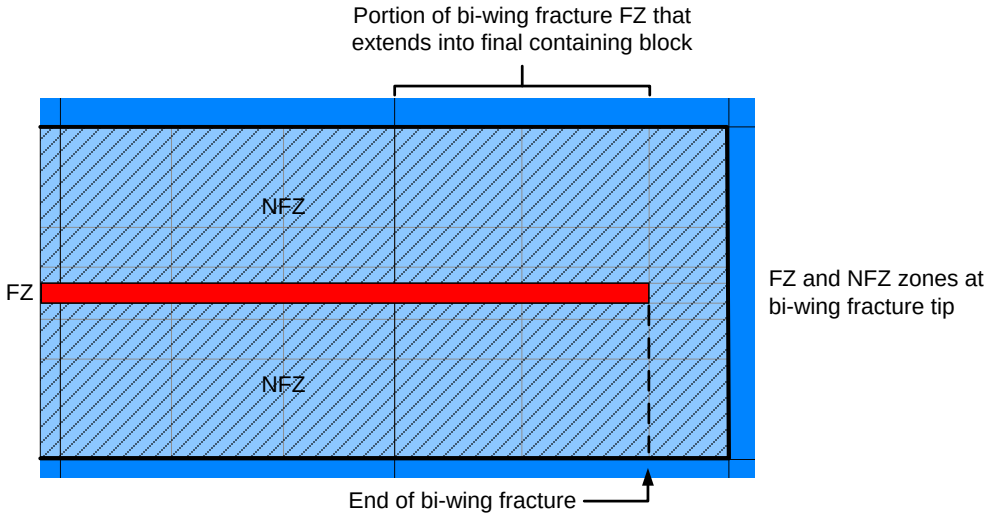
The above planar fracture template defines a bi-wing fracture that is refined (*PLNR_REFINE) into 5 blocks in the I direction, and 7 blocks in the J direction. Logarithmic refinement is applied in both I and J directions only in the central blocks nearest the wellbore. In the I direction, the remaining blocks are evenly refined, with the size approximately that of the largest (outermost) refinement in the logarithmically refined central block, as shown below. In the J direction, logarithmic refinement is maintained.



In our example, the blocks are equally sized in the I direction. If they had not been, differences in the size of the refinements in the I direction outside the central block would be visually more obvious.

There is no refinement in the K direction. The bi-wing half-length (*BWHLEN) is 72 m. The fracture template is oriented in the I direction (*IDIR).

The following shows the refinement of the block containing the tip of the bi-wing fracture. The tip point of the fracture is defined by the bi-wing half-length entered by user. As illustrated in the example, the end of the fracture will usually not align with a block boundary.



The width of the innermost fracture refinement (*INNERWIDTH) is 0.6096 m. The template extends the fracture four layers above (*LAYERSUP) the layer containing the fracture origin, and five layers below (*LAYERSDOWN). Finally, the permeability in the MATRIX continuum fracture zone (*FZ) is 99.6145 md, in the I, J, and K directions.

Once a planar fracture template has been defined, it can be applied to multiple fractures, as shown in the following example:

```
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,25,8 *BG_NAME 'Well-1 - Frac_1'
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,29,8 *BG_NAME 'Well-1 - Frac_2'
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,33,8 *BG_NAME 'Well-1 - Frac_3'
   . . .
   . . .
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,113,8 *BG_NAME 'Well-1 - Frac_23'
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,117,8 *BG_NAME 'Well-1 - Frac_24'
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,121,8 *BG_NAME 'Well-1 - Frac_25'
*PLNRFRAC 'Template (Well-1 - Frac 1)' 31,125,8 *BG_NAME 'Well-1 – Frac_26'
```

As shown in the above example, the origin is different for each fracture. As well, each fracture defines a block group with a unique name. The final fracture in the list uses the properties of planar fracture template 'Template (Well-1 – Frac 1)'. Its origin is block (31,125,8) and it defines block group 'Well-1 – Frac_26'.

Using Builder to Define Complex and Planar Fractures

Builder provides a simple user interface for entering complex and planar fractures. Refer to the "Hydraulically Fractures Wells (IMEX and GEM)" section in the "Wells and Group Control" chapter of the *Builder User Guide* for further information.

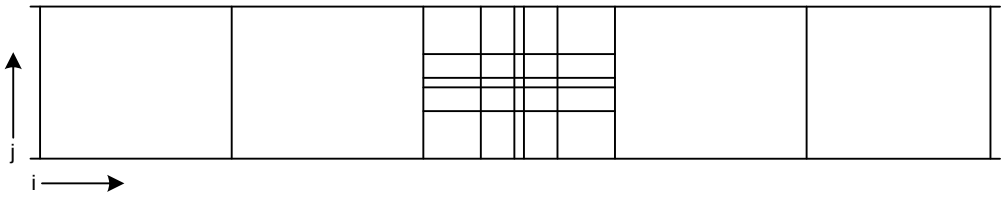
Using GOHFER Data Directly in CMG

Step 1: Define a template for a planar fracture

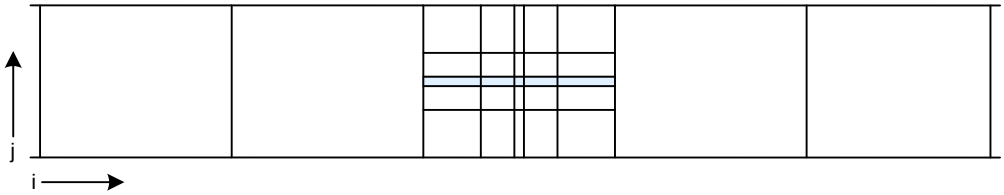
```
PLNRFRAC_TEMPLATE 'Template_Stage_3'
  PLNR_REFINE INTO 5 5 1
  IDIR
  BWHLEN 560
  FRAC_CON
  INNERWIDTH 2
  LAYERSUP 1
  LAYERSDOWN 2
END_TEMPLATE
```

a. PLNR_REFINE INTO 5 5 1

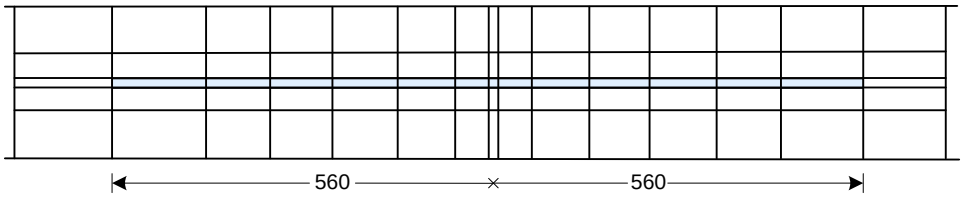
The parent gridblock in which the perforation is defined in is discretized using the LS-LR-DR technique, into a 5x5 locally refined grid:



b. IDIR sets the direction of the main "transverse" fracture.

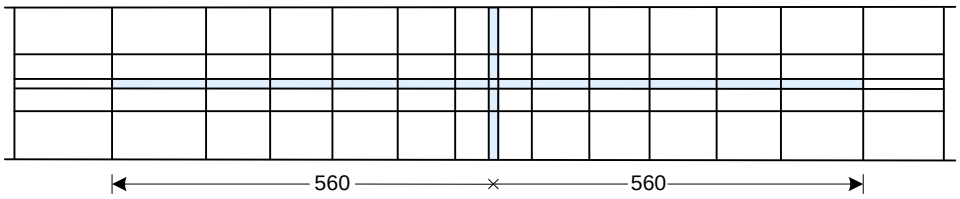


c. BWHLEN defines the half length of the transverse fracture. The LGR grid is extended to encompass the entire fracture:

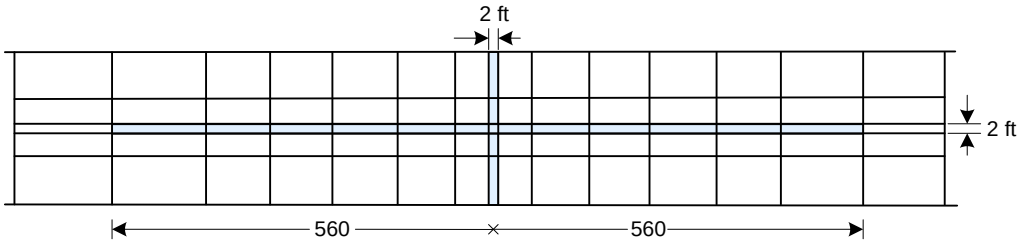


d. FRAC_CON

The fracture zone is allowed to extend longitudinally in the PERF grid block:



e. INNERWIDTH sets an *effective* fracture conduit width of 2 ft.:



f. LAYERSUP and LAYERSDOWN

The LGR is repeated for one layer above the perfed layer and for two layers below the perfed layer.

Step 2: Use the template to define a hydraulic fracture for a specific perforation (in block 13,42,5)

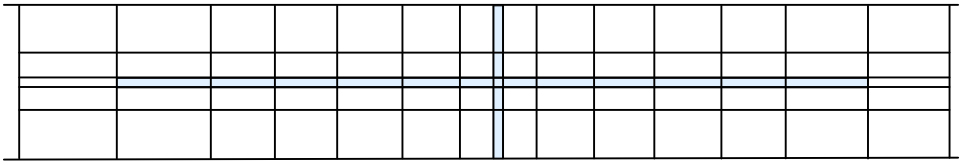
This step will also define three individual structured block groups.

```
PLNRFRAC 'Template_Stage_3' 13,42,5
SBG_NAME 'Frac1_Stage3'
SBG_FZT 'Frac1_Stage3_FZT'
SBG_FZL 'Frac1_Stage3_FZL'
```

Structured block group 'Frac1_Stage3' extends 1 layer above and 2 layers below layer 5

Both FZ and NFZ regions can be referenced:

FZ shaded, NFZ unshaded



15 x 5 x 4 grid
i=1,15, j=1,5, k=1,4 grid

'Frac1_Stage3_FZT' extends 1 layer above and 2 layers below layer 5:

Frac1_Stage3_FZT

13 x 1 x 4 grid
i=1,13, j=1, k=1,4

'Frac1_Stage3_FZL' extends 1 layer above and 2 layers below layer 5:

Frac1_Stage3_FZL

1 x 5 x 4 grid
i=1, j=1,5, k=1,4

Step 3: Define properties in the transverse fracture

```
PERM MATRIX SBG 'Frac_Stage3_FZT' ALL <all 52 (13x4) values>
PERM MATRIX SBG 'Frac_Stage3_FZL' ALL <all 20 (5x4) values>
```

The values have been re-interpolated from the original GOHFER grid to the CMG LS-LR-DK transverse fracture structured block group and the CMG LS-LR-DK longitudinal fracture structured block group.

Modelling Non-Darcy Flow in Hydraulic Fractures Accurately Using a Grid Based Approach

It is possible to use the *NONDARCY model to accurately model the non-Darcy response of a fracture within a simulation using a pseudo fracture. This requires the user to generate a reasonably fine grid representation of the fracture. However we have developed techniques which allow the 0.10 inch wide fracture to be represented by grid blocks as large as 1 foot wide.

The keys to the proper modelling of non-Darcy flow in the 1 foot wide fracture are

1. Adequate representation of flow around the fracture. By this we mean a fine grid surrounding the fracture.
2. The use of an effective fracture permeability (K_{feff}) to ensure that the flow in the fracture is properly accounted for.
3. The use of a properly calculated non-Darcy correction term (*NDARCYCOR) which corrects the Forchheimer number in the fracture for the effective fracture permeability used in (2).

This discussion will assume we are modelling a single 500 ft (½ length) vertical fracture in a 0.10 mD gas water reservoir. The fracture permeability (Kf) has been calculated as 10,000 mD and the fracture width is 0.10 inch. The fracture is assumed to extend from the top to the bottom of the pay.

As mentioned above, the fracture is going to be modeled using a number of 1 foot wide blocks whose total length is 1000 feet. When setting up this grid it is crucial that the well block pressure approximately equals the grid block pressure. In order to do this we represent the intersection of the well and the vertical fracture by a 1 foot by 1 foot grid block. The well radius is not critical, except that it must be close enough to the grid blocks effective (Peaceman) radius to allow the well pressure to be nearly the same as the grid block pressure. For a 1 foot by 1 foot block, a well radius of 1/8th of a foot will accomplish this.

Proper representation of pressure in the reservoir is important, so the blocks surrounding the fracture should vary in a logarithmic fashion away from the fracture both in the direction parallel and perpendicular to the fracture. There should be a detailed discretization of the area next to the fracture which is able to capture the pressure field surrounding the fracture in a reasonable fashion.

Fracture Pseudoization

As the fracture is being modeled as a number of 1 foot wide blocks, the flow down the fracture must be pseudoized in order produce results similar to the flow down a 0.10 inch wide fracture.

The fracture K_f x A should be identical in the simulation model using a 1 foot fracture wide to the simulation model using a 0.1 inch wide fracture.

Therefore the K_{feff} used in this model is reduced by the ratio of the model's fracture area to the actual fracture area, which in this case is equal to the ratio of model fracture width to the actual fracture width (1/0.00833). This gives us a fracture permeability of 83.33 mD.

The use of the effective fracture width corrects Darcy flow down the fracture, but does not properly correct the Forchheimer number in the fracture to account for the reduced fracture permeability. In order to accomplish this, the NDARCYCOR correction term must be defined in the blocks which represent the fracture (only). NDARCYCOR outside of the actual fracture may be set to 1.0 to model non-Darcy flow outside the fracture blocks or to 0.0 to ignore non-Darcy flow outside the fracture blocks. Outside the fracture, non-Darcy effects are normally smaller in magnitude.

In the fracture blocks:

The value of NDARCYCOR = $(K_f/K_{feff})^{(2-N1g)}$

N1g is the exponent of the (K_{rg} x K) term in the correlation for β factor. The β factor is used in the Forchheimer equation (see keyword *NONDARCY).

Assuming we are using Geertsma's model for β, which uses N1g = 0.5, we obtain:

NDARCYCOR = $10000/83.33^{(2-N1g)} = 10000/83.33^{1.5} = 1314.6$.

If another correlation were to be used, for example, where N1g = 1.5, NDARCYCOR would be $10000/83.33^{0.5}$ (= 10.954). The value of the correction is not important, as long as the value of N1g used in the calculation of NDARCYCOR is the same N1 used in the *NONDARCY model.

This example assumes the fracture width and/or fracture permeability does not vary in the fracture. If this were to be modeled, each block would have its own K_{feff} and NDARCYCOR to account for varying K_f and fracture width.

The combination of the effective fracture permeability (to preserve K_f x A in the fracture) and the NDARCYCOR factor as calculated above (to correct the Forchheimer number for K_{feff}) will produce a pressure drop in the coarse fracture model which is nearly identical to the one produced using a 0.10 inch fracture and that will run significantly faster.

GEM has two options for entering the non-darcy model parameters. Three models with fixed parameters can be chosen: Geertsma, Frederick and Graves I, or Frederick and Graves II, or the parameters in the general equation for the Beta factor can be entered for each phase via the NONDARCY GENERAL keyword. The NDARCYCOR parameter can be modified to account for differences between the parameters specified for a general implementation of the non-darcy flow model, and one of GEM's specific models.

The general form of the non-darcy Beta parameter for the gas phase is:

$$\beta = \frac{\alpha}{(K \cdot K_{rg})^{N1} (\phi \cdot S_g)^{N2}}$$

For example, the specific parameters for the Frederick and Graves II model can be inserted in this equation to give:

$$\beta_{FGII} = \frac{2.11 \times 10^{10}}{(K \cdot K_{rg})^{1.55} (\phi \cdot S_g)^{1.0}}$$

To calculate a correction factor (X_{cor}) to multiply NDARCYCOR so that the Frederick and Graves II model will give the same result as a general model, the following equation can be used:

$$X_{cor} = \frac{\alpha / 2.11 \times 10^{10}}{(K \cdot K_{rg})^{(N1-1.55)} (\phi \cdot S_g)^{(N2-1.0)}}$$

For gas problems, use $S_g = 1 - S_{w,con}$ and $K_{rg} = K_{rg}(1 - S_{w,con})$.

LGR Fracture Representation in a Variable Thickness/Depth Grid

In a variable thickness/depth grid it is very convenient to represent the fracture by LGR (Local Grid Refinement). The discussion of proper representation of pressure (above) still applies, but the use of LGR is a flexible way to model one or more hydraulic fractures without increasing overall grid dimensions significantly.

Case 1 - LGR to Model a Fracture whose Length is equal to the Parent Block Length

- Assume for this example that the fracture is parallel to the I direction.
- The fracture is represented by a single row of refined blocks in the I direction in the center of the parent block.
- Assume that the refined block in the fracture which connects the fracture to a sink (well) is a 1 ft. x 1 ft. block in the center of the refined grid and hence in the center of the fracture.
- The number of grid blocks used in the refined grid should be able to adequately represent the pressure drop away from the fracture and should be odd in both the I and J directions in order to center the fracture in the cell.

In this example we will use a 21 x 17 block refined grid to model the 200 ft long fracture within a 200 x 150 ft parent block.

In the I direction, where Di (parent block) = 200.0, Di (iwell) = 1.0

Determine the center refinement:

If NLGRi = 21, the central refinement = Int (NLGRi/2) + 1 = 11

This sets the dimension in the I direction of the well block and its location in the I direction within the LGR

Di (iwell) = Di (11)

The fracture is symmetric about block 11 in the I direction so

Di (iwell+1) = Di (iwell-1), Di (iwell+2) = Di (iwell-2),

Di (iwell+3) = Di (iwell-3), etc.

or

Di (12) = Di (10), Di(13) = Di (9), Di(14) = Di (8), etc.

The LGR block sizes are all based on the size of the well block (Di (iwell)) so that

$$Di\ (iwell+I) = ½Di\ (iwell) \times (R^{I} – R^{(I-1)}) \text{ where } I = 1, Int\ (NLGRi/2)$$

and

$$Di\ (iwell-I) = ½Di\ (iwell) \times (R^{I} - R^{(I-1)}) \text{ where } I = 1, Int\ (NLGRi/2)$$

Where $R = (Di\ (Parent\ Block) / Di\ (iwell))^{(1/\ (Int\ (NLGRi/2)))}$

For the J direction a similar calculation is done:

In the J direction, where Dj (parent block) = 150.0, Dj (jwell) = 1.0

Determine the center refinement:

If NLGRj = 17, the central refinement = Int (NLGRj/2) + 1 = 9

This sets the dimension in the j direction of the well block and its location in the j direction within the LGR

Dj (jwell) = Dj (9)

The fracture is symmetric about block 9 in the j direction so

Dj (jwell+1) = Dj (jwell-1), Dj (jwell+2) = Dj (jwell-2),

Dj (jwell+3) = Dj (jwell-3), etc.

or:

Dj (10) = Dj (8), Dj (11) = Dj (7), Dj (12) = Dj (6), etc.

The LGR block sizes are all based on the size of the well block (Dj (jwell)) so that

$$Dj\ (jwell+J) = ½Dj\ (jwell) \times (RJ – R\ (J-1)) \text{ where } J = 1, Int\ (NLGRj/2)$$

and

$$Dj\ (jwell-J) = ½Dj\ (jwell) \times (R^{J} - R^{(J-1)}) \text{ where } J = 1, Int\ (NLGRj/2)$$

Where $R = (Dj\ (Parent\ Block) / Dj\ (jwell))^{(1/\ (Int\ (NLGRj/2)))}$

An example follows:

Case 2 - LGR to Model a Fracture whose Length is longer than a Parent Blocks Length

In this case we will assume that the fracture length is exactly equal to 3 parent block's length in the I direction. The J direction LGR discretization is calculated using the same procedure for all 3 parent blocks. The central block's I direction discretization remains unchanged.

The blocks on either side of the central parent block in the I direction (as mentioned) above have the same J direction discretization as the central parent block (NLGRj = 17). However it is only necessary to ensure that the I direction discretization produces LGR blocks of equal size in the I direction whose I direction sizes are approximately equal to the largest LGR Di calculated for the central parent block. This will vary with problem size but normally should not require NLGRi to be larger than 5 for the two blocks representing the ends of the fracture.

An example follows.

Example: No LGR Grid - Partial Dataset (Fracture is Parallel to the I Direction)

```
** Reservoir Description Section
** Fracture begins at the first 70 foot DI block and ends at the last 70
foot DI block
DI IVAR 20*200 200 150 100 6*70.0 39.5 20.0 10.0 5.0 2.25 1.50 1.25 1.0 1.25
1.50 2.25 5.0 10.0 20.0 39.5 6*70.0 100 150 200 20*200
DJ JVAR 20*200 150.0 100.0 70.0 40.0 20.0 10.0 5.0 2.25 1.50 1.25 1.0 1.25
1.50 2.25 5.0 10.0 20.0 40.0 70.0 100.0 150.0 20*200

** Permeability in the fracture is set to Kfeff
PERMI CON 0.10
MOD 24:50 31 1 = 83.33
PERMJ EQUALSI
PERMK EQUALSI

** Rock Fluid Property Section
** Use the Geertsma Beta factor
NONDARCY GENERAL 0.50
48511.34 0.5 5.5   10000.0
48511.34 0.5 5.5   10000.0
48511.34 0.5 5.5   10000.0

** Set the pseudoized fracture non Darcy flow correction factor (in the
fracture blocks)
NDARCYCOR CON 1 ** (or 0.0)
MOD 24:50 31 1 = 1314.6
```

Example: LGR Grid from Case 1 - Partial Dataset (200 ft Fracture in 1 block)

```
REFINE 26 26 1 INTO 21 17 1
** In the I direction, R = (200.0/1.0)** 1/ (Int (NLGRi/2))
**                        (200.0/1.0)**0.10 = 1.69864665
**DI(12) = DI(10) = 0.5*(1.69864665-1.0) = 0.34932
**DI(13) = DI(9)  = 0.5*(1.69864665**2  - 1.69864665)     = 0.59337
**DI(14) = DI(8)  = 0.5*(1.69864665**3  - 1.69864665**2)  = 1.00793
**DI(15) = DI(7)  = 0.5*(1.69864665**4  - 1.69864665**3)  = 1.71213
**DI(16) = DI(6)  = 0.5*(1.69864665**5  - 1.69864665**4)  = 2.90830
**DI(17) = DI(5)  = 0.5*(1.69864665**6  - 1.69864665**5)  = 4.94017
**DI(18) = DI(4)  = 0.5*(1.69864665**7  - 1.69864665**6)  = 8.39161
**DI(19) = DI(3)  = 0.5*(1.69864665**8  - 1.69864665**7)  = 14.2544
**DI(20) = DI(2)  = 0.5*(1.69864665**9  - 1.69864665**8)  = 24.2132
**DI(21) = DI(1)  = 0.5*(1.69864665**10 - 1.69864665**9)  = 41.1296
DI IVAR 51*200
DI RG 26 26 1 IVAR 41.1296 24.2132 14.2544 8.39161 4.94017 2.90830
                      1.71213 1.00793 0.59337 0.34932 1.0 0.34932 0.59337
                      1.00793 1.71213 2.90830 4.94017 8.39161 14.2544
                     24.2132 41.1296
** In the J direction, R = (150.0/1.0)** 1/ (Int (NLGRj/2))
**                        (150.0/1.0)**0.125 = 1.870731277
**DJ(10) = DJ(8) = 0.5*(1.870731277-1.0)                  = 0.43536
**DJ(11) = DJ(7) = 0.5*(1.870731277**2 - 1.870731277)     = 0.81445
**DJ(12) = DJ(6) = 0.5*(1.870731277**3 - 1.870731277**2)  = 1.52362
**DJ(13) = DJ(5) = 0.5*(1.870731277**4 - 1.870731277**3)  = 2.85029
**DJ(14) = DJ(4) = 0.5*(1.870731277**5 - 1.870731277**4)  = 5.33212
**DJ(15) = DJ(3) = 0.5*(1.870731277**6 - 1.870731277**5)  = 9.97496
**DJ(16) = DJ(2) = 0.5*(1.870731277**7 - 1.870731277**6)  = 18.6605
**DJ(17) = DJ(1) = 0.5*(1.870731277**8 - 1.870731277**7)  = 34.9087
```

```
DJ JVAR 51*150
DJ RG 26 26 1 JVAR 34.9087 18.6605 9.97496 5.33212 2.85029 1.52362
                   0.81445 0.43536 1.00.43536 0.814451.52362 2.85029
                   5.33212 9.97496 18.6605 34.9087
** Permeability in the fracture is set to Kfeff
PERMI CON 0.10
PERMI RG 26 26 1 JVAR 8*0.10 83.33 8*0.10
PERMJ EQUALSI
PERMK EQUALSI
** Rock Fluid Property Section
** Use the Geertsma Beta factor
NONDARCY GENERAL 0.50
48511.34 0.5 5.5   10000.0
48511.34 0.5 5.5   10000.0
48511.34 0.5 5.5   10000.0

** Set the pseudoized fracture non Darcy flow correction factor (in the fracture
blocks)
NDARCYCOR CON      0
NDARCYCOR RG 26 26 1 JVAR 8*0 1314.6 8*0
```

Example: LGR Grid from Case 2 - Partial Dataset (600 ft Fracture in 3 Blocks)

```
REFINE 26 26 1 INTO 21 17 1
REFINE 27 26 1 INTO 5 17 1
REFINE 25 26 1 INTO 5 17 1
DI IVAR 51*200
DI RG 26 26 1 IVAR 41.1296 24.2132 14.2544 8.39161 4.94017 2.90830
                    1.71213 1.00793 0.59337 0.34932 1.0 0.34932 0.59337
                    1.00793 1.71213 2.90830 4.94017 8.39161 14.2544
                   24.2132 41.1296
DI RG 27 26 1 IVAR 40.0 40.0 40.0 40.0 40.0
DI RG 25 26 1 IVAR 40.0 40.0 40.0 40.0 40.0
DJ JVAR 51*150
DJ RG 26 26 1 JVAR 34.9087 18.6605 9.97496 5.33212 2.85029 1.52362
                            0.81445 0.43536 1.00.43536 0.814451.52362 2.85029
                            5.33212 9.97496 18.6605 34.9087
DJ RG 27 26 1 JVAR 34.9087 18.6605 9.97496 5.33212 2.85029 1.52362
                            0.81445 0.43536 1.00.43536 0.814451.52362 2.85029
                            5.33212 9.97496 18.6605 34.9087
DJ RG 25 26 1 JVAR 34.9087 18.6605 9.97496 5.33212 2.85029 1.52362
                            0.81445 0.43536 1.00.43536 0.814451.52362 2.85029
                            5.33212 9.97496 18.6605 34.9087
** Permeability in the fracture is set to Kfeff
PERMI CON 0.10
PERMI RG 26 26 1 JVAR 8*0.10 83.33 8*0.10
PERMI RG 27 26 1 JVAR 8*0.10 83.33 8*0.10
PERMI RG 25 26 1 JVAR 8*0.10 83.33 8*0.10
PERMJ EQUALSI
PERMK EQUALSI

** Rock Fluid Property Section
** Use the Geertsma Beta factor
NONDARCY GENERAL 0.50
48511.34 0.5 5.5   10000.0
48511.34 0.5 5.5   10000.0
48511.34 0.5 5.5   10000.0
```

```
** Set the pseudoized fracture non Darcy flow correction factor (in the
fracture blocks)
NDARCYCOR CON      0
NDARCYCOR RG 26 26 1 JVAR 8*0 1314.6 8*0
NDARCYCOR RG 27 26 1 JVAR 8*0 1314.6 8*0
NDARCYCOR RG 25 26 1 JVAR 8*0 1314.6 8*0
```

Introducing a Hydraulic Fracture in Recurrent Data

It is possible to model the fracturing of a well at a time other than the simulation's initial start time by introducing an LGR refinement in recurrent data, followed by the redefinition of the permeability arrays in the refinement (*PERMI, *PERMJ ,and *PERMK) and if required redefinition of the non-Darcy flow correction term *NDARCYCOR in the refinement.

It is not necessary to define the LGR refinement in recurrent data. The refinement may be defined at the start of the run and allowed to inherit the parent block properties. The definition of fracture permeability and non-Darcy correction factors in the LGR representing the fracture at a later time is sufficient.

If the LGR keywords are introduced at a later time, followed by permeability arrays which redefine the properties of the newly defined LGR, it is necessary to separate the *REFINE keywords from the *PERMI, *PERMJ, *PERMK keywords by a keyword from a section other than from the Reservoir Description Section. Any recurrent section keyword will suffice.

This is required since the LGR grid is not completely defined until the grid module is exited from (which is triggered by reading a non grid module keyword). Upon reentry to the grid module to read the permeability arrays, the LGR grids will be completely set up.

The user can separate the *REFINE keywords from the *PERM keywords by including the *NDARCYCOR keyword between the two groups.

The user may wish to separate the definition of the LGR and the redefinition of permeability and non-Darcy correction factor by a *DATE keyword and essentially refine the grid a short period of time (say one day) before the *PERMI, *PERMJ, *PERMK and *NDARCYCOR arrays are redefined.

The redefinition of permeability in recurrent data will operate properly with compaction tables. However the compaction table itself cannot be read in or redefined in recurrent data.

The permeability multipliers in the compaction tables will operate on whatever permeability is being used at the time the modified interblock transmissibility is being calculated.

When permeability is read in recurrent data and the SR2 option to output permeability is active (*OUTSRF *GRID *PERM), permeability will be sent to the simulator results file.

When non-Darcy correction factor is read in recurrent data and the non-Darcy option is active, the new non-Darcy correction factors are written to the simulation results file at the time they were defined (assuming *OUTSRF *RES *ALL is used). This allows the display of the currently active correction factors at the proper dates.

Using the Thermal Option in GEM

Beginning with version 2010.10, GEM supports use of a thermal option wherein the temperature equation is solved along with the other equations

The model includes an energy balance with convection, conduction and heat loss terms. The convection terms include phase enthalpies. Aqueous phase enthalpy is determined from a steam table look-up while oil and gas phase enthalpies are determined from the equation of state. Conduction is modelled with a term for thermal conductivity of the rock and fluids, and heat losses to overburden, underburden and edge boundaries are calculated with an analytical formulation.

GEM does not provide comprehensive support for steam injection processes which are better handled in CMG's thermal and advanced processes simulator STARS. For details of the formulation used in GEM, please see Appendix-F.

This tutorial is intended to provide a guideline for preparing and running a thermal model in GEM.

The thermal option is invoked with the keyword *THERMAL *ON in the COMPONENT PROPERTIES section. The rock heat capacity and thermal conductivity are input in the RESERVOIR DESCRIPTION section via *CP-ROCK and *THCONR0 keywords under *CROCKTYPE. The input is quite general in the sense that any selection of grid-blocks or any part of the reservoir can have its own set of rock heat capacity or thermal conductivity. Though GEM provides default values for rock heat capacity and thermal conductivity, it is recommended that reasonable values be provided by the user. If there are multiple *CROCKTYPE keywords, it is necessary to associate them with grid-blocks using the *CTYPE array, otherwise the first *CROCKTYPE is assigned to all the grid-blocks by default.

Heat loss through the reservoir boundaries is specified in the RESERVOIR DESCRIPTION section. The input is similar to that used for aquifer specifications. Heat loss can be specified to occur to the overburden, underburden, or via the reservoir edges. Most generally, heat loss parameters can be specified over any user-defined range of grid blocks. See the *HEAT-LOSS keyword description for more details.

Ideal gas enthalpies are determined from a polynomial correlation for the all components in the oil and gas phases. The coefficients in this polynomial are input via the *ENTHCOEFF keyword in the COMPONENT PROPERTIES section. These coefficients are available for library components in GEM, and are also included in the GEM fluid model file generated by WinProp. Final oil and gas phase enthalpies are calculated by adding the enthalpy departure term from the EOS to the ideal gas enthalpy.

The *TRES keyword in the COMPONENT PROPERTIES section can be used to input a constant reservoir temperature or reservoir temperatures with a constant linear gradient with depth. A generalized initial reservoir temperature input on a grid-block basis can be specified via the *TEMPER keyword in the INITIALIZATION section.

Input of rock-density is required for calculation of the accumulation term using rock heat capacity and temperature difference. Use keyword *ROCKDEN in the ROCKFLUID data section to input rock-density on a per grid-block basis.

For control of time-step size, norms and maximum changes in temperature can be input via keyword *NORM *TEMP, and *MAXCHANGE *TEMP in the NUMERICAL METHODS control section. Convergence of temperature equation can be controlled by *CONVERGE *TEMP.

In the WELL AND RECURRENT DATA section, the specification of bottom-hole injection temperature for the injection wells is mandatory for thermal runs. The specification can be done via keywords *INJ-TEMP or *IWELLBORE. It may be preferable to use *INJ-TEMP, unless well-bore hydraulic calculations are being done for the injectors in which case *IWELLBORE may include bottom-hole injection temperature along with other data. Note that *IWELLBORE and *PWELLBORE provide a more generalized way for specifying temperature at the bottom-hole or at the wellhead and are useful when wellbore pressure- or temperature-drop calculations are desired.

The reservoir temperature can be visualized by requesting an output of the temperature array via *OUTSRF *TEMP, or written to the output file via *OUTPRN *TEMP. The log file contains additional columns regarding maximum change in the temperature and corresponding grid-block location for thermal runs. Similarly additional output is provided in the main output file concerning energy balance at the end of a DATE/TIME change.

Example:

Following data-file fragment shows certain thermal related keywords in different sections. You may or may not need all of these keywords. Please refer to appropriate section of the User Guide for keyword details.

```
** === INPUT/OUTPUT SECTION ====
*INUNIT *FIELD
*OUTPRN *GRID *SO *SG *TEMP
*OUTSRF *GRID *SO *SG *TEMP

** === RESERVOIR DESCRIPTION =========
*GRID *CORNER 308 55 125

*CROCKTYPE 1
  *CP-ROCK 0.25  ** Rock Heat Capacity [BTU/(LB F)]
  *THCONR0 2.0   ** Thermal conductivity of rock
                 ** and [BTU/(HR LB F)]
*CROCKTYPE 2
  *CP-ROCK 0.37
  *THCONR0 1.75
** Assign crocktypes to grid-blocks
*CTYPE *KVAR  100*1 25*2

*HEAT-LOSS *BOTTOM-TOP-BOUNDARY
**      rockden  rockcp therm-cond
*HLPROP 165.0    0.25   2.0

** === COMPONENT PROPERTIES ==========
*MODEL PR
*NC 3 3
*TRES 100.
```

```
*ENTHCOEF
-5.5810E+00  5.6480E-01 -2.8300E-04  4.1740E-07
-1.5260E-10  1.9590E-14
-7.6010E-01  2.7310E-01 -4.2960E-05  3.1280E-07
-1.3900E-10  2.0070E-14
2.9110E+01  2.0400E-03  4.3490E-04  -8.1810E-08
7.2350E-12 -1.4560E-16

*THERMAL *ON

** === ROCK FLUID SECTION =====================
*ROCKFLUID


*ROCKDEN *CON 165.4 **Rock Density [LB/CUFT]

** INITIAL SECTION =========================
*INITIAL

*TEMPER *CON 120.

** === NUMERICAL SECTION ========================
*NORM *PRESS 145.
*NORM *SATUR 0.15
*NORM *GMOLAR 0.15
*NORM *TEMP 5.

** === WELLS AND RECURRENT SECTION =============
*RUN

 *CYCLPROD  1
 *PWELLBORE *MODEL
 **wdepth  wlength  rel_rough  whtemp  bhtemp  wradius
   7500.  7500.    0.0001      80.     120.    0.25
  *OPERATE  *MAX  *STG  6.2000E+6
  *OPERATE  *MIN  *BHP    500.000

  *INJECTOR  2
  *IWELLBORE
 **wdepth  wlength  rel_rough  whtemp  bhtemp  wradius
   7500.  7500.    0.0001      80.     270.    0.25
  *INCOMP   *CYCLING

*DATE 2015 05 10
*STOP
```

One-way Geomechanics Calculations using Post-Processing of Prior Simulator Response (GEOPOSTPRO)

Beginning with version 2010.12, GEM supports one-way geomechanics calculations based on the simulator response from a prior run.

This tutorial provides a guideline for carrying out one-way geomechanics calculations using GEM. It is important to recognize the processes where such calculations can be done to provide meaningful results. Note that the one-way calculations can provide a close estimate of the geomechanics behavior under certain situations however they are not a replacement for coupled flow- and geomechanics-simulation. Since these calculations are done by using prior simulator output as an input to the geomechanics module within the simulator, no flow-simulator time steps are necessary during these simulations.

Making a Parent GEM run

Before making a GEM run one should consider whether the output of that run is going to be used for post-processing and geomechanics calculations. If so, make sure that pressure array (using keyword *PRES) and in case of non-isothermal model temperature array (using keyword *TEMP) is written out to simulation results (SR2) file. Alternately one can write and preserve restart (RST) file that always contains pressure and temperature output in addition to many other records that are required for simulation restart. Note that since restart file contains a large amount of information apart from pressure and temperature, it could be voluminous especially for large models or if the restart information is written out frequently for models of even moderate size. Irrespective of whether pressure (and temperature) is being written to SR2 or RST file for subsequent post-processing, keyword *WRST 0 should not be used in the parent data file. Typical keyword entries in the parent run would be as under:

Example 1:

```
*TITLE1 'History Match Model for field ABC'
*TITEL2 'SR2 File to be used for Geomech Sensitivity'
*INUNIT *FIELD
*WSRF *TIME
*OUTSRF *GRID *PRES   ** Write out pres array at
                      ** well time change
```

Example 2:

```
*TITLE1 'History Match Model for field ABC'
*TITEL2 'RST File to be used for Geomech Sensitivity'
*INUNIT *FIELD
*WRST *TNEXT     ** Restart records will be written
                 ** at time steps at which Geomech calcs
                 ** will be performed in subsequent run
…
…
…
*RUN
…
…
*WRST *TNEXT
```

```
*DATE 2015 05 31
…
…
…
*WRST TNEXT
*DATE 2025 05 31
*STOP
```

Note that Geomechanics keywords are not mandatory in the parent run. It is necessary that *neither of* the SR2 (extensions .irf and .mrf) and Restart (extension .rst) files should be deleted if a geomechanics post processing run is to be performed subsequently.

Making a Geo Post Processing (GEOPOSTPRO) Run

A geo post processing run can be made simply by modifying the parent run by adding some control keywords (discussed below) and the keywords for the geomechanics section. At this time Builder support is unavailable to make this file in a customized way, however, the steps mentioned below can help you accomplish that task:

1. Copy the parent file into a separate name.
2. Add the *INDEX-IN filename in the Input/Output Section as the input to the GEOPOSTPRO run.
3. Add relevant *OUTSRF, *OUTPRN keywords requesting for geomechanics specific output and their writing frequency.
4. Add regular geomechanics section keywords starting with *GEOMECH.
5. Add *GEOPOSTPRO keyword in geomechanics section, and if required follow it by starting and ending time steps.
6. Add *GEOPSTINPUT followed by *SR2 or *RST to use pressure/temperature information written on sr2 file in the parent run or the same from the restart file.
7. Save the file and run it as a usual GEM model.

The relevant keywords from a GEOPOSTPRO data file will be similar to one shown below:

Example 3:

```
**Specify input file having prior simulation's response.
*FILENAMES *INDEX-IN 'gmgmc038.irf'
*TITLE1 'SPE3'
*TITLE2 'Perform one way geomech using '
*TITLE3 'pressure output from gmgmc038'
*INUNIT *FIELD

*WPRN *GRID 1
*WPRN *WELL 1

*OUTPRN *GRID  *PRES  *VDISPL
*OUTSRF *WELL  *PAVG

*WSRF *GRIDDEFORM  1
*OUTSRF *GRID *PRES  *YOUNG *POISSON *VDISPL
              *VERDSPLGEO *SUBSIDGEO *STRESI *TSTRESI
```

```
*OUTSRF *WELL *STRESI   3 3 1
              *STRESSM  3 3 1
              *STRESEFF 3 3 1
…
…
…
*GRID
…
…
…
*ROCKFLUID
…
…
*INITIAL
…
…
…
*NUMERICAL
…
…
…

*GEOMECH           ** geomechanics main keyword
*GCOUPLING 0
*GEOPOSTPRO 15  55  ** Perform geomech calculations
                    ** for prior simulator response
                    ** between prior simulation's
                    ** timesteps 15 through 55
*GEOPOSTINPUT *SR2
*ELASTMOD 1.428571E+06
*POISSRATIO .3
*COHESION 10000
** Initial effective stress in the reservoir
** H stress : horizontal stress on a plane
** V stress : Vertical stress on a plane
** N stress : Normal stress to a plane
**      H stress  V stress  Shear stress  N stress
*STRESS 6000       6000        0           6000
…
…
…
*RUN
*DATE 1986 1 1
…
…
…
*TIME 5478.
*STOP
```

The Log and Output file from a GEOPOSTPRO run

The primary log file shows the progress of geomechanics calculations and the source of input data along with the timestep numbers. The grid output is written to the main output as well as the new sr2 file that is created after the GEOPOSTPRO run. No secondary log file is created in such runs.

All the functionalities of geomechanics including GEOGRID are available in GEOPOSTPRO runs. However, where the coupling between flow-simulation and geomechanics is inherently implicit, GEOPOSTPRO runs will not provide any meaningful result. Typical cases are permeability modifications runs (Barton-Bandis and other such options).

Using Geomechanics Keywords in the Recurrent Section

The time stepping in the GEOPOSTPRO runs is controlled by the availability of pressure (and temperature) information in the sr2/rst file of the parent run between the specified bounds (*GEOPOSTPRO keyword). If no information is available at a particular timestep, corresponding geomechanics calculation cannot be done in the post-processing run. The recurrent geomechanics keyword, if available, is applied at a time (date) change later unlike the regular GEM runs. User is, therefore, required to be careful while using the geomechanics keyword in the recurrent section. If required, an additional time (date) change card can be inserted to facilitate intended reading.

Seeking Parent Chain of Prior Run

If a chain of restart runs has been created, GEM will, by default, try to seek all the records in the parent tree including the one specified via *INDEX-IN file, to get pressure (and temperature) information. If required this behavior can be turned off by keyword *SEEKPARENT *OFF in the Input/Output section. In that event only the SR2 (RST) set specified in the data file via *INDEX-IN will be used.

Template Files

Template files gmgmc038 through gmgmc046 illustrate the use of GEOPOSTPRO feature in GEM.

Keyword Data Entry System

Introduction to Keyword System

INTRODUCTION

In a keyword input system, each data item or group is preceded by a keyword indicating what that data item or group is. For example,

```
*MAXERROR 10
```

indicates that a maximum of 10 data entry errors are allowed before the simulator stops. Many data items have defaults, which are used if the keyword is not found in the input data file.

CHARACTER SET

There is a set of allowed characters that may be used in referring to keywords. Any character not in this set will be interpreted as a blank. Characters in quotes or comments are not checked, but will be passed along unchanged to the output.

The purpose of the character set is to detect invisible non-blank characters, such as tab, which some editors may insert in your data file.

The CMG keywords are composed of the upper and lower case alphabet, numerals 0-9, keyword indicator (*), and arithmetic operators (=, +, -, /). Extra characters are included in the set to accommodate the *TRANSLATE facility (see below).

You may increase the character set at installation time by expanding the data definition of the array CHRSET in subroutine RDLINE in the simulator source code. The only restriction is that the characters must be supported by the computer operating system.

KEYWORD INDICATOR

The optional keyword indicator * (asterisk) may appear immediately before the keyword with no blanks between.

An example of a keyword is the porosity keyword:

```
por or POR or *POR
```

In this Manual, keywords are shown in capitals with '*' so that they stand out in the text. However, mixed case, and without '*', are allowed.

Two keyword indicators or asterisks, in a row, indicate a comment line, as in:

```
** This is a comment line. The comment line may
** appear almost anywhere in the data set. It is
** very useful for documenting your data set.
```

The comment indicator may be changed by using the *COMMENT keyword described later in this section.

ORDER OF KEYWORDS

All keywords used in the keyword input system are grouped into keyword groups.

Keyword groups must appear in the data file in the same order as they appear in this document. Keywords within a keyword group may appear in any order, unless specifically stated otherwise.

There are a few keywords which may appear at any point in the data file. These keywords are *LIST, *NOLIST, *INCLUDE, *COMMENT and *TRANSLATE.

Some keywords may appear both within their keyword group, and in recurrent data.

The description of each keyword notes whether the keyword is optional or required. Some keywords are optional or required with the use of certain other keywords. Optional keywords have default values which are used if the keyword is not found in the data file.

STRINGING KEYWORDS

Keywords that are not subkeywords of a keyword or keywords that are in fact separate entities under one keyword heading cannot be strung along on one line.

The rule is one keyword, with its valid subkeyword or subkeywords, per line, unless otherwise stated or illustrated.

For example, the aquifer option requires 3 keywords. *AQUIFER has a list of subkeywords and requires the location of the aquifer. The option also needs aquifer properties (*AQPROP) and the aquifer influence function (*AQFUNC). The three keywords are separate entities and may not be strung along on the same line.

An incorrect presentation of the first two keywords is:

```
*AQUIFER *BOUNDARY *AQPROP  1000  .4  .002  230
*AQFUNC
0.01  0.112
...    ...
```

The correct presentation of these 3 keywords is:

```
** The keyword *BOUNDARY is a designated subkeyword
** of *AQUIFER and so can be strung on the same
** line as *AQUIFER.

*AQUIFER *BOUNDARY

*AQPROP
** thickness   porosity   permeability   radius
    1000.0        .4          .002       230.0

*AQFUNC
** dimensionless   influence
**     time         function
       0.01          0.112
       ...             ...
```

CASE

Keywords and alphanumerical strings may be in upper case, lower case, or any combination. Filenames must conform to the requirements of the operating system being used, for example, upper case for IBM mainframe systems.

LINE LENGTH

Only the first 512 characters in a line are processed, and any character after that is ignored.

DELIMITERS

Keywords, numbers, and character strings must be separated from each other by blanks, commas, or new line characters. Consecutive commas, with nothing except blanks between them should not occur in the data file.

CHARACTER STRINGS

Character strings ALWAYS must be enclosed in a pair of single quotes (e.g. '5-35-48-W5') or double quotes (e.g. “5-35-48-W5”). When inserting either type of quote in the string, enclose the string in the other quote type, e.g., 'This is the "right" way.’ or “Land’s End”. When a string’s maximum length is specified, characters after that maximum will be ignored.

TRANSLATION

You can use your own keyword for any main keyword if you define the translation rule using *TRANSLATE.

NUMBERS

Numbers are input in free format. Real numbers do not require decimal points. Exponentiation is indicated by 'E', 'e', 'D' or 'd'. Numbers must not contain embedded blanks.

If an integer is expected, and a number with a decimal fraction is read in, an error message will be issued, and the program will stop.

The following are examples of valid real numbers:

```
25.040
-3
1.23E+02
0.02D-4
34.e02
+2.3
 +.3
 -.3
```

The following are NOT valid real numbers:

```
34. E 02 <-- blanks in number
- 34.E02 <-- blank in number
34.E.2   <-- decimal in exponent
```

Sequences of numbers may be separated either by commas or by blank spaces.

REPEAT COUNT

There is a simple way to input multiple sequential occurrences of a number. Suppose you have five numbers in order:

```
.23   .23   .23   .41   .27
```

There are two ways to input these numbers. One is to write them as they appear directly above. However a shortcut measure is to write them using the multiple occurrence indicator ("*"). Since the first three numbers in sequence are the same you can write the numbers this way:

```
3*.23   .41   .27
```

Note that there MUST NOT be a space either before or after the "*".

INTEGER RANGE

In any instance where a sequence of INTEGER values is required, a colon must be used to indicate a range of values from one integer to another integer. Blanks cannot be present between either integer and the colon. For example:

```
1   2   3   4   6   10   11   12
and
1:4   6   10:12
```

are two equivalent ways of giving the same sequence of INTEGERS. Note that this method of input will not work if real numbers are expected.

TABLES

The keyword documentation sometimes indicates that a table of data must be entered. All the required data items are listed in a particular order. Always enter the data in the order shown in the documentation.

A value is also expected for each data item. Data items cannot be omitted except where it is explicitly shown in the documentation that certain items, located at the ends of rows, are optional. Optional items are shown enclosed within round brackets in the particular table syntax.

An example of such an event includes the water-oil relative permeability tables (*SWT keyword). Pcow is optional, and need not be entered, but in this case the user has capillary pressure data.

***SWT**

****Sw**	**krw**	**krow**	**(Pcow)**
0.2	0.0	1.0	45.0
0.2899	0.022	0.6769	19.03
0.3778	0.018	0.4153	10.07
0.4667	0.061	0.2178	4.09
0.5556	0.143	0.0835	1.80
0.6782	0.289	0.0123	.50
0.7561	0.450	0.0	.10
0.8325	0.780	0.0	.0
0.9222	1.000	0.0	.0
1.0000	1.000	0.0	.0

If the capillary pressure is not used (pcow = 0), then the table would be entered as

```
*SW
**Sw      krw       krow
0.2       0.0       1.0
0.2899    0.022     0.6769
0.3778    0.018     0.4153
0.4667    0.061     0.2178
0.5556    0.143     0.0835
0.6782    0.289     0.0123
0.7561    0.450     0.0
0.8325    0.780     0.0
0.9222    1.000     0.0
1.0000    1.000     0.0
```

The following table is unacceptable, as items anywhere within the row cannot be omitted:

```
*SWT
**Sw      krw       krow      (Pcow)
0.2       0.0       1.0         45.0
0.2899    0.022     0.6769      19.03
0.3778              0.4153      10.07
0.4667    0.061     0.2178       4.09
0.5556    0.143     0.0835       1.80
0.6782    0.289                   .50
0.7561              0.0           .10
0.8325    0.780     0.0           .0
0.9222    1.000     0.0
1.0000    1.000     0.0
```

ERROR AND WARNING MESSAGES

During data input, the lines in the data file are echoed to the print output file. If an error is detected, an error message or a warning is issued. Depending on the type of error, the message may refer to the line printed above or below the error or warning message.

If *NOLIST has been used, the data line on which the error or warning has occurred will not be printed. It is therefore recommended the *NOLIST only be used for production runs, after the data has been thoroughly debugged.

Comments (Optional)

PURPOSE:

** (two keyword indicators) may be used to add comments explaining where data came from, why options are being used, etc.

FORMAT:

** comment_text

DEFAULTS:

Optional. No defaults.

CONDITIONS:

A comment may appear at any point in the data file.

EXPLANATION:

Two consecutive keyword indicators ('**') indicate the start of comment text. The portion of the input line after the two keyword indicators is ignored. Comment lines may be used to add comments explaining where data came from, why options are being used, etc.

Comments are copied to the output print file with the rest of the data file (subject to *NOLIST and *LIST keywords). Otherwise, comment lines are ignored.

An example of a comment is:

```
*MAXERROR 14 ** Change maximum number of errors.
```

Blank Lines (Optional)

PURPOSE:

Blank lines may be used to separate sections of a data file, and generally make the data file more readable.

CONDITIONS:

Blank lines may appear at any point in the data file.

EXPLANATION:

Blank lines are copied to the output print file with the rest of the data file (subject to *NOLIST and *LIST keywords). Otherwise, blank lines are ignored.

Scan Mode for Checking Errors *CHECKONLY

PURPOSE:

*CHECKONLY indicates that the well data is to be checked for format errors, but the simulation will not be run.

FORMAT:

*CHECKONLY

DEFAULTS:

If this keyword is absent, time step calculations will be done.

CONDITIONS:

This keyword must appear in the I/O Control section.

Include Files (Optional)

PURPOSE:

The *INCLUDE keyword indicates that reading of the primary input data set is suspended. Instead, a secondary file will be read.

FORMAT:

```
*INCLUDE  'filename'
```

DEFAULTS:

Optional. No defaults.

CONDITIONS:

The *INCLUDE keyword must appear on a line by itself. Only one secondary file may be open at a time. Nesting of *INCLUDE keywords is not allowed.

NOTE:

SOME INSTALLATION SITES MAY NOT SUPPORT THE *INCLUDE KEYWORD.

EXPLANATION:

Enclose the include filename in single quotes.

When a *INCLUDE keyword is encountered, the named secondary input file is opened and data is read from the file. When the end of the secondary file is reached, the secondary file is closed and data reading continues in the primary (or original) input file.

Controlling Data File Listing (Optional) *LIST, *NOLIST

PURPOSE:

*LIST specifies listing the input data file, from this point forward, to the output print file.

*NOLIST specifies not listing the input data file to the output print file, starting immediately after the current line.

FORMAT:

*LIST
*NOLIST

DEFAULTS:

Optional keywords. Default: *LIST

CONDITIONS:

*LIST or *NOLIST may appear at any point in the data file, but must be on a line by itself.

EXPLANATION:

By default, the entire data file is listed to the output print file prior to the start of the simulation run, with the exception of a limit of 20 echoed lines for each grid-array keyword. If a *NOLIST keyword is inserted in the data file, the data file is not listed from the point of the *NOLIST keyword until a *LIST keyword or the end of data file is reached. Keyword *NOLISTLIM disables the limiting of grid-array keyword data.

Changing the Comment Indicator (Optional) *COMMENT

PURPOSE:

*COMMENT changes the two character sequence that denotes the beginning of a comment.

FORMAT:

*COMMENT 'ab'

DEFINITION:

ab

A two-character string denoting the start of a comment. The string 'ab' must be enclosed in quotes.

DEFAULTS:

Optional keyword. Default: *COMMENT '**'

CONDITIONS:

*COMMENT may appear at any point in the data file, but must be on a line by itself. All subsequent comments following the appearance of the *COMMENT keyword must be preceded by the two-character sequence 'ab'.

EXPLANATION:

By default, comments in the data file are denoted by the character string '**'. This may be changed by using the *COMMENT keyword.

Example:

```
*COMMENT '--'
*TRANSLATE 'KX' 'PERMI' -- This is a translate
                        -- rule
```

From this point on in the data file all comments should begin with '--'. In the above example the two lines beginning with '--' are comments.

Changing the Keywords by Using Translate Rules (Optional)

***TRANSLATE**

PURPOSE:

*TRANSLATE changes or translates your own favorite keyword into a CMG simulator recognizable keyword.

FORMAT:

*TRANSLATE 'your_keyword' 'CMG_keyword'

DEFINITION:

your_keyword
: A single-word keyword that you want the simulator to recognize. The allowed characters are those in the character set specifier in subroutine RDLINE in the simulator source code; no blanks, commas or asterisks are allowed. You may add any character that your computer operating system supports to this character set. Enclose the string in single quotes.

CMG_keyword
: The CMG simulator keyword (WITHOUT asterisk) that you want to replace. This must be a valid keyword recognized by the simulator. This must be enclosed in single quotes.

DEFAULTS:

Optional keyword. Default: Use the internal simulator keywords.

CONDITIONS:

*TRANSLATE may appear at any point in the data file, but must be on a line by itself. Subsequently, a simulator keyword may be referred to by using either 'your_keyword' (defined by a *TRANSLATE keyword) definition or the internal simulator keyword 'CMG_keyword'.

EXPLANATION:

If you need to redefine a keyword because you want to make the keyword more meaningful to yourself, or simply for convenience, the *TRANSLATE keyword will accomplish this task.

Example:

```
*TRANSLATE  'KX'  'PERMI'
```

This translate rule translates the *KX or KX keyword such that the simulator recognizes this to mean *PERMI. Subsequent to this keyword *KX, KX, *PERMI, or PERMI may be used to refer to the *PERMI keyword.

A keyword may have more than one translate rule,

Example:

```
*TRANSLATE 'KX' 'PERMI'
*TRANSLATE 'x_permeability' 'PERMI'
*TRANSLATE 'permx'          'PERMI'
```

User Block Address *UBA

PURPOSE:

Specify address for any block in the grid.

EXPLANATION:

The User Block Address (UBA) is a natural extension of the familiar I-J-K notation for addressing blocks, but allows the user to refer to any block in the advanced grids available in CMG simulators. UBA is based on the idea that, even for the most complex grids, the relationships between grids and blocks can be represented as a tree diagram with the fundamental grid at the base node. The UBA starts by giving the fundamental grid block in I-J-K notation, e.g., “3,6,2” or “3 6 2”. Delimiting blanks or commas must separate the block indices.

Refined Grids

To refer to a block at a finer grid level, the fundamental I-J-K is followed by a slash (/) and the next level of refined grid block in I-J-K notation, and so on, to the block of interest. Delimiting blanks or commas must surround a slash.

For example, if fundamental block (3,6,2) contains a 3x3x1 refined grid, then the UBA for one of the fine blocks is “3,6,2 / 2,3,1”. UBA “3 6 2” still refers to the fundamental “parent” block.

Naturally Fractured Grids

In the naturally fractured grid options, a fundamental “spatial” block is split into two “storage” blocks: fracture and (optionally refined) matrix. Indicate which of the two with UBA qualifiers MT or FR after the indices, e.g., “10,14,3 FR” and “10,14,3 MT”. If the block is fractured and neither FR nor MT is present, FR is assumed.

UBA Ranges

Some keywords support the entry of ranges of blocks in the UBA format, for example, *PERF. In this case, any index may appear as an integer range (e.g., “3 6 1:4 / 2 2 1:3 FT”).

UBA for Output

User block addresses appear in many places in the simulator output. For output purposes only, a descriptive 2-letter abbreviation is appended to the UBA of each block that is not a normal block in single porosity system.

NL	null block
PN	pinch out block
ZP	zero porosity block
FR	fracture block in dual porosity system
MT	matrix block in dual porosity or dual permeability

Where an output field does not have sufficient length for the entire address, the presence of additional refinement levels will be indicated by "+", e.g., “23,13,12+ FR”.

Input of Grid Property Arrays

ARRAY

Grid properties which are input are, in fact, arrays of data with one array element for each grid block. Grid properties are indicated by 'ARRAY:' in the left column immediately following the title on the manual page which describes them.

ARRAY READING OPTIONS

An array assignment consists of five parts, two of which are optional. The syntax is:

grid_array (array_qualifier) read_option data (array_modifier)

DEFINITIONS:

grid_array

The property being assigned, such as *POR. In the manual this is denoted as ARRAY: *POR

array_qualifier

This is used to assign data to different block elements and different grids. block (e.g., matrix and fracture). The array_qualifier is optional and has following form: Optional element qualifier

*MATRIX

*FRACTURE

*ALLELEM

followed by optional grid qualifier

*RG uba_range (*FZ | *NFZ)

*BG *'block_group'* (*FZ | *NFZ)

*SBG *'str_block_group'* (*FZ | *NFZ)

These keywords are described separately.

If an element qualifier is absent then *ALLELEM is assumed, except in chapter ‘Reservoir Description’ where *MATRIX is assumed.

If grid qualifier is absent then the fundamental grid is assumed.

Each of the above array reading qualifiers will access only the indicated grid and element in that grid. The user must ensure that all elements of each grid block have been assigned required data.

uba_range

A User Block Address (UBA) range or single UBA, without UBA qualifiers MT and FR. A UBA range is not allowed with read_option *EQUALSI or array_qualifier *MOD.

‘block_group’

Quoted block group name defined by keyword *BLOCKGROUP. See manual page for *BG in this chapter. See *read_options* *FZ and *NFZ.

‘*str_block_group*’

Quoted structured block group name defined by keyword *SBLOCKGROUP. See manual page for *SBG in this chapter. See *read_options* *FZ and *NFZ.

read_option

The read options are:

*CON

*IVAR

*JVAR

*KVAR

*ALL

*IJK

*BINARY_DATA

*EQUALSI

*FZ

*NFZ

These *read_option* keywords are described separately.

All these *read_option*s except *IJK, *FZ and *NFZ ensure definition of each block in the specified grid. *IJK, *FZ and *NFZ must be used with care to ensure that the grid is covered completely; this restriction is lifted in the RECURRENT DATA section where select blocks may be defined.

data

These are the actual values for the grid_array. The amount of data depends on the read_option.

array_modifier

Once an array has been input, it can be modified immediately using *MOD. This allows modification of blocks or regions after the read_option is done. The *MOD keyword is described separately.

PURPOSE:

*MATRIX is used immediately after a grid property keyword to indicate that a matrix property is being input.

KEYWORD:

*MATRIX

EXPLANATION:

Note that *MATRIX is the default for entering all grid properties if no array input qualifier is present. Therefore, you need to use *MATRIX only when the natural fracture option has been enabled.

Any of the array reading options can be used with *MATRIX. The read_option keyword must follow the *MATRIX keyword.

Example: To input the matrix porosity in a dual porosity system:

```
*POR  *MATRIX  *ALL
.12  5*.16  .18  .22  .21  8*.20
.19  10*.18  3*.21  .19  .16
```

Entering Fracture Grid Properties *FRACTURE

PURPOSE:

*FRACTURE is used immediately after a grid property keyword in a dual porosity system to indicate that a fracture property is being input.

KEYWORD:

*FRACTURE

EXPLANATION:

Any of the array reading options can be used with *FRACTURE. The array reading option keyword must follow the *FRACTURE keyword.

Example: Suppose the planes of grid blocks with J = 2 and J = 3 are fractured. You want to input the fracture porosities of these blocks.

```
*POR  *FRACTURE  *IJK
1:10  2:3  1:3  .08
```

Entering Refined Grid Properties *RG

PURPOSE:

*RG is used to assign values of an array to refined grid blocks.

KEYWORD:

*RG i1(:i2) j1(:j2) k1(:k2)

DEFINITIONS:

i1(:i2)
: I direction index range of the fundamental grid block containing the refined grid.

j1(:j2)
: J direction index range of the fundamental grid block containing the refined grid.

k1(:k2)
: K direction index range of the fundamental grid block containing the refined grid.

EXPLANATION:

Refined grids are initially defined using the *REFINE keyword in the RESERVOIR DESCRIPTION section.

By default, all refined grid blocks are assigned the values assigned to the fundamental grid block. The *RG keyword allows input of different values for each refined grid block.

Any of the array reading options may be used with *RG. The array of properties input is that of the refined grid, and corresponds to the number of blocks in the refined grid, not the fundamental grid. The array reading option keyword must follow the *RG keyword.

Example: Suppose block (1,1,1) contains a refined grid. It is divided up into ni=3, nj=2 and nk=1. You want to input the porosity of each of the refined grid blocks.

```
*POR *RG 1 1 1 *ALL
.08  .079  .078  .081  .08  .076
```

Assign Data to Block Groups *BG

PURPOSE:

*BG assigns values or tasks to a group of blocks.

KEYWORD:

*BG ‘block_group’

DEFINITIONS:

‘block_group’
A quoted character string containing the name of a block group defined by keyword *BLOCKGROUP in the “Reservoir Description” chapter.

CONDITIONS:

Block group block_group must have been defined before *BG *'block_group'* appears.

The read_option after *BG ‘block_group’ must be *CON, *EQUALSI, *FZ or *NFZ.

EXPLANATION:

Block groups are defined using keyword *BLOCKGROUP in the “Reservoir Description” chapter.

The only other array qualifier options allowed with *BG are:

array_qualifier	**Action**
*MATRIX	Assign only to matrix blocks in the block group
*FRACTURE	Assign only to fracture blocks in the block group
*ALLELEM	Assign to all blocks in the block group

The only array reading options allowed with *BG are:

read_option	**Data**	**Action**
*CON	One number	Assign to each specified block
*EQUALSI	None	Apply *EQUALSI action to each specified block; not available in Reservoir Description data section
*FZ, *NFZ	See *FZ manual page	Assign data to fracture or non-fracture zone.

Any *MOD action applied to the entire group is allowed, but any *MOD action involving a UBA is not allowed.

Since *BG may assign data to only a fraction of the grid, *BG should be used only after the whole-grid assignment is well established via defaults or whole-grid keyword specification.

For a mandatory assignment (e.g., *POR), use a data form that assigns the entire grid, then optionally use *MOD with block groups to overwrite desired regions.

Since a block may be found in multiple block groups, it is possible to attempt multiple assignments of the same quantity or task using *BG with multiple groups. The action of each block-group assignment is fully processed before the next one is started. The usual rules for multiple assignments to a block are used: value assignments are overwritten, value modifications are accumulated or overwritten, and multiple tasks are allowed or not depending on the nature of the task (e.g., *REFINE may be applied at most once).

Example: Assume a fundamental 3x2x2 grid, part of which is specified as block group 'Sand'. All blocks in the sand region are refined locally. Porosity and I-direction permeability are assigned to the entire grid and then overridden in the sand region. The J-direction permeability *EQUALSI applies to all blocks, including the sand region. The K-direction permeability *EQUALSI applies ¼ the I-direction value in all blocks.

```
*GRID *CART 3 2 2
. . .
** Define 4-block group for sand region
*BLOCKGROUP 'Sand' 1:2 2 1:2 *CON 1
. . .
*REFINE *BG 'Sand' *INTO 3 3 2
. . .
*POR *CON 0.22
   *MOD
      *BG 'Sand' = 0.34
      *BG 'Sand' * 1.1
. . .
*PERMI *KVAR 2500 3300
   *MOD *BG 'Sand' = 4000
*PERMJ *EQUALSI
*PERMK *EQUALSI / 4
```

Assign Data to Structured Block Groups *SBG

PURPOSE:

*SBG assigns values or tasks to a group of blocks defined by *SBLOCKGROUP.

KEYWORD:

*SBG *'str_block_group'*

DEFINITIONS:

'str_block_group'
A quoted character string containing the name of a structured block group defined by keyword *SBLOCKGROUP in the "Reservoir Description" chapter.

CONDITIONS:

Structured block group *str_block_group* must have been defined before *SBG *'str_block_group'* appears.

EXPLANATION:

A structured block group is a local I-J-K grid that is superposed over a rectangular array of blocks and is used for referencing one or more of those blocks. Use keyword *SBLOCKGROUP in the "Reservoir Description" chapter to define a structured block group.

The only other array qualifier options allowed with *SBG are:

array_qualifier	**Action**
*MATRIX	Assign only to matrix blocks in the block group
*FRACTURE	Assign only to fracture blocks in the block group
*ALLELEM	Assign to all blocks in the block group

All the usual array reading options (*IVAR, etc.) are allowed with *SBG, but the data corresponds to the local I-J-K grid defined for *'str_block_group'* by *SBLOCKGROUP.

Any *MOD action applied to the entire group is allowed. For any *MOD action, any UBA and *ARRAY corresponds to the local I-J-K grid.

Since *SBG may assign data to only a fraction of the total grid, *SBG should be used only after the whole-grid assignment is well established via defaults or whole-grid keyword specification.

Since a block may be found in multiple block groups, it is possible to attempt multiple assignments of the same quantity or task using *SBG with multiple groups. The action of each block-group assignment is fully processed before the next one is started. The usual rules for multiple assignments to a block are used: value assignments are overwritten, value modifications are accumulated or overwritten, and multiple tasks are allowed or not depending on the nature of the task (e.g., *REFINE may be applied at most once).

Example: Assume a fundamental 110x170x54 grid, part of which is specified as structured block group 'Sand'. All blocks in the sand region are refined locally. Porosity and I-direction permeability are assigned to the entire grid and then overridden in the sand region. The J-direction permeability *EQUALSI applies to all blocks, including the sand region. The K-direction permeability *EQUALSI applies ¼ the I-direction value in all blocks.

```
*GRID *CART 110 170 54
. . .
** Define a local block group for sand region
*SBLOCKGROUP 'Sand' 22 32 5
   40:61 115:146 23:27
*END_SBLOCKGROUP
. . .
*REFINE *SBG 'Sand' *INTO 3 3 2
. . .
*POR *CON 0.22
   *MOD
      *SBG 'Sand' = 0.34
      *SBG 'Sand' * 1.1
. . .
*PERMI *KVAR 2500 3300
   *MOD *SBG 'Sand' = 4000
*PERMJ *EQUALSI
*PERMK *EQUALSI / 4
```

Assign Property to Fracture Zone (Conditional) *FZ, *NFZ

PURPOSE:

*FZ and *NFZ specify assignment of an input property to a type of fracture zone.

FORMAT:

*FZ *fz_all*
-or-
*FZ *fz_centre fz_tip*

*NFZ *nfz_all*

DEFINITIONS:

*FZ *fz_all*
Assign property value *fz_all* to all blocks that are in the fracture zone.

*FZ *fz_centre fz_tip*
Specify the values to be assigned to the fracture zone as a function of the value *fz_centre* at the fracture centre and the value *fz_tip* at the fracture tips.

*NFZ *nfz_all*
Assign property value *nfz_all* to all refined blocks that are not in the fracture zone.

CONDITIONS:

*FZ *fz_all* and *NFZ *nfz_all* may be used only by a local grid created with *REFINE *INNERWIDTH or *PLNR_REFINE (which also uses *INNERWIDTH).

*FZ *fz_centre fz_tip* may be used only by a fracture grid created with *PLNR_REFINE or *PLNRFRAC.

EXPLANATION:

For a fracture-type local grid created by *REFINE *INNERWIDTH, *PLNR_REFINE or *PLNRFRAC, the fracture zone is defined. This in turn allows usage of *FZ to target the local grid's fracture zone and *NFZ to target the remaining refined blocks. *FZ *fz_all* applies the same input value to all blocks in the fracture zone, and *NFZ *nfz_all* does the same for the remaining blocks. Usage of both *FZ and *NFZ for a property ensures assignment of that property for all blocks in the local grid.

The fracture zone created by *PLNR_REFINE or *PLNRFRAC has additional structure (tips versus centre) which is used by *FZ *fz_centre fz_tip* to assign *fz_centre* to the fracture centre, *fz_tip* to the fracture tips, and some function to interpolate between them.

Examples:

Suppose we have a planar fracture called ′HydFrac_21′ which has fracture-block permeabilities of 1000 md but tip values of 15 md.

```
...
  *INNERWIDTH 1.3 *BWHLEN 120. *JDIR
  *BG_NAME 'HydFrac_21'
...
*PERMI *MATRIX   *KVAR *CON 0.01 ** md
*PERMI *FRACTURE *KVAR *CON 70   ** md
*PERMI *MATRIX   *BG 'HydFrac_21' *FZ 1.5 0.15
*PERMI *FRACTURE *BG 'HydFrac_21' *FZ 1000 15
*PERMI *MATRIX   *BG 'HydFrac_21' *NFZ 0.1
*PERMI *FRACTURE *BG 'HydFrac_21' *NFZ 10
```

Assigning Grid Properties to all Elements *ALLELEM

PURPOSE:

Assign data to all elements of grid blocks.

KEYWORD:

*ALLELEM

EXPLANATION:

This array input qualifier indicates the data is to assigned to all elements of a grid block. This qualifier is necessary only if a natural fracture option is used. Since this is the default, this keyword is not needed explicitly.

Example: To specify the same initial temperature of 40.5 degrees in both fracture and matrix:

```
*TEMPER *ALLELEM *CON 40.5
-or-
*TEMPER *CON 40.5
```

Constant Value Arrays *CON

PURPOSE:

*CON indicates that a constant value is entered for all array elements. The value may be entered on the same line or the next line.

KEYWORD:

*CON value

EXPLANATION:

Example: Assume you have a reservoir with a constant value of porosity of 0.16, and a constant permeability in the I direction of 100 md.

```
*POR  *CON
0.16
*PERMI  *CON  100.
```

Array Input in IJK Notation *IJK

PURPOSE:

*IJK assigns a constant value of a grid property within the region defined by the minimum and maximum block number in each of the three directions.

KEYWORD:

```
*IJK
 i1:i2  j1:j2  k1:k2  value
```

DEFINITIONS:

i1
: Initial I direction grid block index.

i2
: Final I direction grid block index.

j1
: Initial J direction grid block index.

j2
: Final J direction grid block index.

k1
: Initial K direction grid block index.

k2
: Final K direction grid block index.

value
: Constant value of the array for the defined region.

EXPLANATION:

The *IJK array reading option assigns a constant value of a grid property within the region defined by the minimum and maximum block number in each of the three directions. Later lines will overwrite previous lines if they refer to the same grid blocks.

You must input enough lines so that every grid block has been assigned a value.

If both the initial and final grid block indices in any direction are the same, the colon and final grid block index may be omitted. For example, a porosity of .2 exists in a grid region given by i1=1, i2=3, j1=3, j2=4, and k1=k2=2. Input may be either:

```
1:3    3:4    2:2    .2
-or-
1:3    3:4    2      .2
```

Example: Suppose you have a reservoir with a porosity of 0.16 everywhere except for an interior region where the porosity is 0.22. You are using a grid where ni=20, nj=25, and nk=6 and the region of higher porosity is between grid block indices I=5 and I=9, J=14 and J=23 and for K=4. First, assign the entire grid with a porosity of 0.16. Then overwrite the interior region with its higher porosity.

```
*POR  *IJK
1:20  1:25  1:6  0.16
5:9  14:23  4    0.22
```

Array Input of Values that Vary in the I Direction *IVAR

PURPOSE:

*IVAR is used to indicate values that vary in the I direction, but which are constant in the other two directions.

KEYWORD:

*IVAR value(1) ... value(ni)

DEFINITIONS:

value(1)
: Value assigned to all grid blocks with an I direction index of 1.

ni
: Number of grid blocks in the I direction.

EXPLANATION:

Enter nj values separated by spaces or commas.

Example: I direction block sizes where ni = 10:

```
*DI  *IVAR
2*1000  1100  1050  3*800  860  1010  1100
```

Note that the structure '2*1000' indicates the value '1000' occurs twice.

Example: I direction block sizes where ni = 3:

```
*DI  *IVAR  3000.0  4000.0  5000.0
```

Array Input of Values that Vary in the J Direction *JVAR

PURPOSE:

*JVAR is used to indicate values that vary in the J direction, but which are constant in the other two directions.

KEYWORD:

*JVAR value(1) ... value(nj)

DEFINITIONS:

value(1)
: Value assigned to all grid blocks with a J direction index of 1.

nj
: Number of grid blocks in the J direction.

EXPLANATION:

Enter nj values separated by spaces or commas.

Example: The J direction increments for a problem where nj=10 are: 755, 755, 755, 825, 825, 1000, 1000, 1100, 800, 800.

```
*DJ  *JVAR  3*755  2*825  2*1000  1100  2*800
```

Example: The J direction has just 3 blocks:

```
*DJ  *JVAR  3000.0  4000.0  3000.0
```

Array Input of Values that Vary in the K Direction *KVAR

PURPOSE:

*KVAR is used to indicate values that vary in the K direction, but which are constant in the other two directions.

KEYWORD:

*KVAR value(1) ... value(nk)

DEFINITIONS:

value(1)
Value assigned to all grid blocks with a K direction index of 1.

nk
Number of grid blocks in the K direction.

EXPLANATION:

Enter nk values separated by spaces or commas. This is convenient for entering properties vary only by layer.

Example: Porosity varies for each of the layers of a system where nk=5, but is constant within each layer. The layer porosities are: .0810, .210, .180, .157, and .200.

```
*POR *KVAR .081 .21 .18 .157 .2
```

Example:

```
** Each of the I, J, and K permeabilities
** are constant within each layer of the
** reservoir but vary from layer to layer.
** Hence use *KVAR to input them layer
** by layer.

*PERMI *KVAR 200.0 50.0 500.0
*PERMJ *KVAR 200.0 50.0 500.0
*PERMK *KVAR  20.0 40.0  60.0
```

Values that Vary for Most or All Grid Blocks *ALL

PURPOSE:

*ALL is used to indicate that values vary in most or all the grid blocks. The number of values expected is the number of grid blocks in the grid, including all null or zero-porosity blocks.

KEYWORD:

*ALL value(1) ... value(ni*nj*nk)

EXPLANATION:

Values are entered starting with block (1,1,1) and in increasing block order where the I direction block index increases fastest and then the J direction block index second fastest and the K direction block index the slowest.

See FIGURE 1 in Appendix D for the ordering of grid blocks.

Example: Porosities for each grid block in a three-dimensional system vary in almost every grid block: ni=10, nj=3, nk=2

```
*POR *ALL
.08  .08  .081 .09  .12  .15  .09  .097 .087 .011
.15  .134 .08  .087 .157 .145 .12  .135 .18  .092
.074 .12  .12  .154 .167 .187 .121 .122 .08  .08

.095 .13  .12  .157 .17  .18  .184 .122 .084 .09
.11  .12  .134 .157 .157 .18  .18  .098 .09  .09
.08  .09  .144 .143 .123 .16  .165 .102 .10  .10
```

Values Stored in Binary Form *BINARY_DATA

PURPOSE:

Builder uses this keyword to indicate that grid-array data is stored in binary form.

KEYWORD:

*BINARY_DATA

EXPLANATION:

In Builder

Normally Builder writes data in text-format files. However, Builder is able to write some grid definition and property data in binary form to a separate binary-format file. This option is invoked in Builder via menu "File/Save As.../Array Saving Method/Binary File Format (*.cmgbin)". The binary file is saved in the same folder as the main dataset file and given the same root name but extension ".cmgbin". Unlike the *INCLUDE facility which can involve multiple include files, there is at most one binary format file associated with a main data file.

The grid definition and property data that can be written in binary form are (1) corner-point definition keywords *XCORN, *YCORN, *ZCORN, *COORD and *CORNERS, and (2) all grid-array keywords using read option *ALL in the initialization (non-recurrent) data sections. In any given data set, all such data is written in the same form (text or binary) according to the selected Array Saving Method. You may switch between the binary and text-only writing formats whenever you wish.

For grid property keywords, only that data associated directly with subkeyword *ALL is written in binary form. Specifically, *MOD data lines are preserved as text after the *BINARY_DATA subkeyword. Therefore, you may add or modify *MOD data lines after the *BINARY_DATA subkeyword, just as you would after *ALL and its data.

Non-uniform grid property data from sources like maps or existing simulator results from Simulation Results file (sets) usually are written in *ALL format in the text file. Such data can be written directly to the binary file, avoiding *ALL text writing altogether.

Writing non-uniform data to a binary file has some distinct advantages. First, the reading of binary data is much faster than text and so for large models can speed up significantly the transfer of data to and from Builder. Second, binary format occupies less space than a comparative text representation (e.g., 8 bytes versus 20 to 30 bytes). Third, the original precision of data obtained from existing Simulation Results file(s) can be preserved by not passing the data through a text-writing step.

In the Simulator

When the simulator detects subkeyword *BINARY_DATA in its initial data scan, it opens the associated binary file which is assumed to have the same root name as the main data file but extension ".cmgbin". Each time the simulator encounters *BINARY_DATA during the data loading pass, it locates that property in the binary file and reads one value for each block in the grid, similar to the *ALL option. A mismatch between the text and binary parts of the data set will result in an error.

The reading of binary data is much faster than text and so for large models can speed up significantly data reading in the simulator. Text formatted data can differ slightly in value from its associated binary data, so text and binary versions of the same data may give slightly different simulation results.

Examples:

These are examples of data fragments you might see in the text data file written by Builder in binary file format.

```
** Null block distribution from map
*NULL *BINARY_DATA

** Permeability from map, with matching adjustments
*PERMI *BINARY_DATA
   *MOD 1:5 1:10 1:5 * 0.9
        1:3 1:4  1:2 * 1.2
*PERMJ *EQUALSI
*PERMK *EQUALSI / 5.

** Natural fracture:  matrix porosity from map
*POR *MATRIX *BINARY_DATA
*POR *FRACTURE *CON 0.008

** Initial saturation, natural fracture system
*SW *MATRIX *BINARY_DATA
*SW *FRACTURE *BINARY_DATA
```

J and K Direction Data from I Direction *EQUALSI

PURPOSE:

*EQUALSI indicates that values in the J and K directions are the same as those in the I direction, or that the values given for the I direction may be modified by division, multiplication, etc.

KEYWORD:

*EQUALSI

EXPLANATION:

*EQUALSI is used with direction-dependent keywords, such as the transmissibility, permeability and dispersion coefficients. The I direction keyword must have all of its elements (matrix, fracture, wellbore, etc.) assigned before *EQUALSI can be used with the J and K direction keywords.

Example: Permeabilities in a single-porosity system. J-direction values are equal to the I direction, but the K-direction values are twice the I-direction values

```
*PERMI *CON 100.0
*PERMJ *EQUALSI
*PERMK *EQUALSI * 2.
```

Example: The same as above, only with a natural fracture option in effect.

```
*PERMI *MATRIX *CON 100.0
*PERMI *FRACTURE *CON 10000
*PERMJ *EQUALSI
*PERMK *EQUALSI * 2.
```

Note that if *EQUALSI and *MOD appear together, then *EQUALSI is processes first and then the *MOD values are processed.

Modifying Array Data (Conditional) *MOD

PURPOSE:

*MOD indicates the modification of an input grid property.

FORMAT:

*MOD
  { *region_mod* }
-or-
*MOD *operator value*
where
region_mod = (*i1*(:*i2*) *j1*(:*j2*) *k1*(:*k2*) *operator value*
| *i1*(:*i2*) *j1*(:*j2*) *k1*(:*k2*) *ARRAY *value_array*
| *BG *'block_group' operator value*)
| *SBG *'str_block_group' operator value*
| *SBG *'str_block_group'* *ARRAY *value_array*)
operator = (+ | – | * | / | =)

DEFINITIONS:

operator value

The operation described below is applied to each non-null referenced block in the specified region. This region is specified in one of two ways according to the syntax used.

1. ***region_mod***: all the blocks specified by the I-J-K range or block group on that data line only.
2. ***MOD operator value***: all the blocks specified by the main keyword's array qualifiers (e.g., *RG, *BG, *SBG, *MATRIX, *FRACTURE).

operator	**Action**
+	Add *value* to the existing property value
–	Subtract *value* from the existing property value
*	Multiply the existing property value by *value*
/	Divide the existing property value by *value*
=	Replace the existing property value by *value*

i1(:*i2*) *j1*(:*j2*) *k1*(:*k2*)

Do the specified data modification in the region indicated by this I-J-K range. Null blocks will be skipped. These I-J-K indices correspond to the grid context, either a refined grid specified by *RG, a structured grid specified by *SBG or the fundamental grid. The I-J-K range option is not available for the *BG context.

*ARRAY *value_array*

Replace the existing property values in the region by the corresponding values from *value_array*. The number of values in *value_array* must be (*i2-i1*+1) (*j2-j1*+1) (*k2-k1*+1) and repeat counts are allowed. A value must be specified for each block in the I-J-K range, even if the block is null. The *value_array* entries appear in the order used by grid-array read option *ALL, that is, I index changes fastest and K index changes slowest.

*BG '*block_group*'

Apply data modification to each block in the region defined by this block group and possibly array qualifiers *MATRIX, *FRACTURE or *ALLELEM. This region can be of arbitrary shape and on multiple grid levels. This *region_mod* option is allowed only when the main keyword refers to the fundamental grid, (array qualifiers *RG, etc., are absent).

*SBG '*str_block_group*'

Apply data modification to each block in the region defined by this structured block group and possibly array qualifiers *MATRIX, *FRACTURE or *ALLELEM. This *region_mod* option is allowed only when the main keyword refers to the fundamental grid (array qualifiers *RG, etc., are absent). The *ARRAY sub-option is allowed.

CONDITIONS:

The *MOD keyword must appear immediately after the array property data, and may appear at most once for each array property keyword.

When *RG *uba_range* is used together with *MOD, *uba_range* must be only a single UBA (e.g., 1 2 3), not a UBA range (e.g., 1:3 2 3).

When a grid-array keyword uses array qualifier *BG *'block_group'* together with *MOD, only the "*MOD *operator value*" syntax may be used. This restriction avoids attempts to modify data via *MOD outside of the specified block group.

The *region_mod* option *BG *'block_group'* is allowed only when the main keyword refers to the fundamental grid, i.e., array qualifiers *RG, etc., are absent. This restriction avoids attempts to apply the *BG option inconsistently.

EXPLANATION:

The *MOD option is used to modify the last grid property data array input by adding, subtracting, multiplying, dividing or replacing array elements by a specified value. There are two syntax variations. In the simpler variation an operator follows immediately after the *MOD keyword, in which case the modification is applied to all grid blocks referenced by that grid property keyword.

In the other syntax variation the *MOD keyword is followed by a number of sets of I-J-K index range, operator and value. Each range-operator-value set is processed in order of appearance, so that a block may experience more than one modification after all the *MOD data is processed. Histories of modifications can appear in sequence, perhaps accumulated in a matching study.

Examples:

Suppose for a 10 x 6 x 1 grid you want to modify the porosities in the region with I indices 1 through 3, J indices 1 through 4 and with K index of 1 by adding 0.01. You further wish to assign the value of .13 to the block with I=5, J=2, and K=1. Enter *MOD after the array values. The data looks like this:

```
*POR *ALL
.08  .08  .081 .09  .12  .15  .09  .097 .087 .011
.15  .134 .08  .087 .157 .145 .12  .135 .18  .092
.074 .12  .12  .154 .167 .187 .121 .122 .08  .08
.095 .13  .12  .157 .17  .18  .184 .122 .084 .09
.11  .12  .134 .157 .157 .18  .18  .098 .09  .09
.08  .09  .144 .143 .123 .16  .165 .102 .10  .10
   *MOD
     1:3 1:4 1 + .01
     5   2   1 = .13
```

To modify the entire grid to reduce the porosity of each grid block to 95% of the original value:

```
*POR *ALL
.08  .08  .081 .09  .12  .15  .09  .097 .087 .011
.15  .134 .08  .087 .157 .145 .12  .135 .18  .092
.074 .12  .12  .154 .167 .187 .121 .122 .08  .08
.095 .13  .12  .157 .17  .18  .184 .122 .084 .09
.11  .12  .134 .157 .157 .18  .18  .098 .09  .09
.08  .09  .144 .143 .123 .16  .165 .102 .10  .10
   *MOD * .95
```

You may not repeat a required primary array keyword after it has been entered once. For example, the user wants to change some of the porosities to 0.22 after initially assigning 0.30 to all grid blocks. The following data entry is incorrect.

```
*POR *CON 0.3
*POR *IJK 5:8 14:23 4 0.22  ** Incorrect
```

The correct procedure is to use the *MOD keyword on the line immediately following *POR:

```
*POR *CON 0.3
   *MOD  5:8  14:23  4 = 0.22
```

Note that if *EQUALSI and *MOD appear together, then *EQUALSI is processed first and then the *MOD values are processed.

The *MOD keyword may be used together with *RG for a single UBA. If fundamental block (3,7,5) has a 2x2x1 locally refined grid, then the following is valid.

```
*POR *RG 3 7 5 *ALL 0.30 0.39 0.29 0.26
   *MOD * 1.08
```

The *ARRAY option lets you replace values in a rectangular region of the grid, using an array of values. Using the 10 x 6 x 1 example grid above, a region could be modified as follows.

```
*POR *CON 0.24
   *MOD 3:5 2:3 1 *ARRAY
                  .08  .087 .157  ** (3:5,2,1)
                  .12  .154 .167  ** (3:5,3,1)
```

The *MOD keyword may use both I-J-K and *BG region specifiers, where block groups 'Frac-12' and 'Frac-15' and structured block group 'Frac-16'.

```
*POR *CON 0.25
   *MOD
      1:3 5:7 89:92 = 0.37
      *BG 'Frac-12' = 0.32
      *BG 'Frac-12' * 1.05
      *BG 'Frac-15' = 0.29
      1:3 5:7 89:92 * 1.2
      *BG 'Frac-15' * 1.04
      *BG 'Frac-15' * 1.12
      *SBG 'Frac-16' * 1.04
      *SBG 'Frac-16' *ARRAY *1.12
```

Interpolating Table Data (Optional) *INT

PURPOSE:

*INT indicates that the corresponding table entry should be filled by interpolation.

EXPLANATION:

The *INT keyword may be used in table input. This keyword enables the calculation of the table entry by interpolation. Essentially the table entry corresponding to *INT is replaced by a linearly interpolated value. This option is useful when not all table entries are known. This feature is explained in further detail with the help of an example.

Suppose that it is required to enter a water-oil relative permeability table into the simulator. Also assume that the water and oil relative-permeabilities are known at different saturations.

***SWT**

****Sw**	**Krw**	**Krow**
0.2	0.0	1.0
0.3	0.05	*INT
0.4	*INT	0.7
0.5	0.40	*INT
0.6	*INT	0.5
0.7	0.8	*INT
0.8	1.0	0.0
1.0	1.0	0.0

In the above table values denoted by *INT will be calculated by linear interpolation by the simulator.

NOTE:

Interpolation is done with respect to the first column. Thus the *INT keyword cannot appear in the first column of the table.
At least one non *INT entry must appear in a column. If only one non *INT entry appears in the column then the entire column is assigned the same value.

Input/Output Control

Notes on Input/Output Control

GEM uses run-time dimensioning. The simulator pre-scans portions of the data file to estimate how much computer memory is required to do a simulation. It then attempts to obtain as much memory as is required from the computer, and begins the simulation if the memory is available.

GEM is able to proceed in most cases using the above information. However, there are cases where GEM's estimates are too large and some additional control is required to get better estimates. There are defaults built into the estimation process which can be overridden by the user using the *DIM keyword as described later in this section.

This run-time dimensioning facility means that GEM does not have to be custom dimensioned for various applications.

See the discussion of the *DIM keyword for dimensioning control.

GEM can produce output display files in the Simulation Results formats, which are readable by CMG's RESULTS. The XDR (eXternal Data Representation) system is the default file writing method, which allows for inter-platform inter-operability.

Restarted runs in GEM always use native binary (platform specific) .RST files.

Command-Line Arguments (Optional) *COMMAND-LINE-ARG

FORMAT:

gem.exe
(-f *input_data*)
(-log (*log_path*))
(-log2)
(-r *input_restart*)
(-restart (*nstart*))
(-restime *restime*)
(-resdate *resdate*)
(-checkonly)
(-dimsum)
(-onestep)
(-wait)
(-dd)
(-wd)
(-file_no_replace)
(-imrf)
(-sr3)
(-sr3_imrf)
(-resolve *n*)
(-forgas *n*)
(-perfscan)
(-doms)
(-jacpar)
(-parasol (*n*))
(-autop2d pn_1qn_2)
(-pnthrds *n*)
(-grthrds *n*)
(-pgthrds *n*)

DEFINITIONS:

gem.exe

GEM invocation command, usually the name of the executable file. It can be a local file, a link to a file, or a file accessible via your search path, depending on your computer.

-f *input_data*

Specifies that *input_data* is the path name to a GEM input data file.

-log (*log_path*)

Directs screen output to a file. If *log_path* is present, this file has path name *log_path* and extension “.log”. If this extension is not present in *log_path* it will be added. If *log_path* is absent this file has the root file name of the data set and the extension ".log". Screen input is read from a file which has the root name of the data set and the extension ".in". If a file name has not been

defined using the –f command line option and the –log option is used, it can be entered in the ".in" file.

-log2

Creates a secondary log file with root name of the input data file and extension “.log2”.

-r *input_restart*

Specifies that *input_restart* is the pathname to a GEM input restart file, given using a .RST extension, or an .IRF extension, which was generated by a previous run with the same executable.

-restart (*nstart*)

Equivalent to putting *RESTART in your data, with or without *nstart*. See manual entry for *RESTART. *Keyword entries *RESTART, *RESTIME or *RESDATE, if present in data file, will get precedence over this command line option.*

-restime *restime*

Equivalent to putting *RESTIME *restime* in your data. See manual entry for *RESTIME. Keyword entries *RESTART, *RESTIME or *RESDATE, if present in data file, will get precedence over this command line option.

-resdate *resdate*

Equivalent to using *RESDATE in your data. See manual entry for *RESDATE. *Keyword entries *RESTART, *RESTIME or *RESDATE, if present in data file, will get precedence over this command line option.*

Data *resdate* is the concatenation of the year, 2-digit month and 2-digit day, plus optional decimal point and day fraction, e.g., “19900805.5” (noon on Aug 5, 1990). This string can be found in the Simulation Results index file (*.irf) at the end of the TIME card of the timestep of interest. You are allowed to include slashes between the year-month-day info, in which case the month and whole day may be 1 or 2 digits each, e.g., “1990/8/5.5”.

-checkonly

Equivalent to putting *CHECKONLY in your data. See the manual entry for *CHECKONLY.

-dimsum

Equivalent to putting *DIM *DIMSUM in your data.

-onestep

Equivalent to putting *MAXSTEPS 1 in your data.

-wait

Wait for License. If all available licenses are being used, this argument keeps the process in a ‘sleep’ mode until a license is available (up to 72 hrs.).

-dd

Simulator output files are created in the directory where the data files are specified. This can be different from the current directory where the executable resides. Include files are assumed to be in the same directory as data.

-wd

Simulator output files are created in the specified working directory. Data files reside either in the current directory or in a directory specified by the ‘ -f ‘ command line argument. Include files are assumed to be in the same directory as data.

-file_no_replace

Prevents overwriting of all output files, including the log/diary file if "-log" is used. With this argument, an attempt to overwrite an existing output file will result in a fatal error message issued to the log file (if "-log" is used) or the default output device (screen).

-imrf

Equivalent to *SRFORMAT *IMRF. Write out Simulation Results Output to IRF/MRF file set (it was default prior to GEM 2016.10).

-sr3

Equivalent to *SRFORMAT *SR3. Write out Simulation Results Output to SR3 file.

-sr3_imrf

Equivalent to *SRFORMAT *SR3_IMRF. Write out Simulation Results Output to SR3 file, as well as IRF/MRF file set. This is the default behavior in GEM 2016.10.

-resolve *n*

Used to invoke a coupled GEM-GAP run.

n = 0 means Resolve controls all the GAP timesteps.

n = 1 means GEM provides feedback for the next GAP timestep.

Absence of *n* after -resolve implies *n* = 0.

-forgas *n*

Used to invoke a coupled GEM-FORGAS run.

n = 0 means FGI controls all the FORGAS timesteps.

n = 1 means GEM provides feedback for the next FORGAS timestep.

Absence of *n* after -forgas implies *n* = 0.

-perfscan

Perform pre-scanning of well and completion changes in the data file i.e., scan for these changes prior to time stepping.

-doms

Enables parallel processing for Jacobian building. Domains are equally divided amongst threads. If no other parallelization input is present number of threads is defaulted to 2. See Numerical Methods Control section for details on input data for parallelization. In GEM parallel keyword input in the data file overrides the command line options.

-jacpar

Enables parallel processing for Jacobian building. Jacobian domains are based on solver classes. See Numerical Methods Control section for details on input data for parallelization. In GEM parallel keyword input in the data file overrides the command line option.

-parasol (*n*)

Enables parallel processing for matrix solution via solver PARASOL. Optional *n* specifies the number of threads to use (default 2). See Numerical Methods Control section for details on input data for parallelization. In GEM parallel keyword input in the data file overrides the command line option.

-autop2d pn_1qn_2

Automatic parallel partitioning in two-dimensions can optionally be accomplished with this option, where p and q could be any of i, j, or k, with p≠q; and n_1 and n_2 are number of partitions desired in p and q directions. See Numerical Methods Control section for details on input data for parallelization. In GEM parallel keyword input in the data file overrides the command line option.

-pnthrds *n*

Number of threads *n*, to be used for simulator's Jacobian building and other simulator specific tasks. If no input follows command line option -parasol, then *n* is also used as number of threads for PARASOL. See Numerical Methods Control section for details on input data for parallelization. In GEM parallel keyword input in the data file overrides the command line option.

-grthrds *n*

Number of threads *n*, to be used for Grid Module's parallel processing. If option -grthrds is absent then threads for Grid Module parallel processing are the same as maximum of simulator and PARASOL's parallel processing threads.

-pgthrds *n*

Number of threads *n*, to be used for Geomechanics Module's parallel processing. If option -pgthrds is absent then threads for Geomechanics Module parallel processing are the same as the maximum of simulator and PARASOL's parallel processing threads.

-cputime

This command line option enables non-intrusive clock and cpu time profiling of GEM. A clock and cpu time statistics is printed at the end of the run in both log and output files when this option is invoked.

-compiler_info

This command line option enables printing of compiler information, at the end of the log and output files, with which the GEM executable was built.

DEFAULTS:

If an input data filename is not supplied here via argument “-f”, then GEM will prompt for it. If “-log” is absent, the prompted data is read from the default input device (standard_in, keyboard). If “-log” is present then the prompted data is read from a file whose name is (a) the same as *log_path* but with extension “.in”, or (b) “cmg.in” if *log_path* is absent.

If “-f” is absent but “-log” is present without *log_path*, the “diary” output goes to a file named "cmg.log".

If “-log” is absent, “diary” output will go to the default display (standard_out, screen).

If this is a restart run and the input restart filename is not supplied here via argument “-r” or via keywords *FILENAME *INDEX-IN, then GEM will prompt for it.

If neither –wd nor –dd is supplied, then output filenames are obtained from the *FILENAMES keyword. If *FILENAMES is absent, then the output files are written to the current working directory.

If “-nowait” is absent, the run is kept in a sleep mode (up to 72 hrs) until a license is available. This is useful when several jobs are submitted via the CMG Technology Launcher at one time (e.g., over the night or weekend) and the number of licenses is limited. An alternate way to run a series of jobs sequentially is to use a batch file.

If “-file_no_replace” is absent, an existing file is replaced with a new file of the same name generated by the current run.

If “-perfcan” is not present scanning of well and completion changes is not performed before timestepping.

CONDITIONS:

These arguments form must be supplied at the command line.

EXPLANATION:

GEM requires an input data file to start running. Control over all other file names, specified or defaulted, may be contained in it, but the input data file's name must be supplied explicitly by the user.

GEM directs the simulation results to output files but has diary-style output which comes out to the standard output device, or to a file identified using “-log”.

The secondary log file always has the root name of the data file and extension “.log2”. The command-line input “-log2” attempts to create the secondary log file such that its contents are different than that of the primary log file. For instance if *CHANGES format is opted for the primary log file, the command line input “-log2” will direct GEM to create the secondary log file with *WELL-INFO format, and vice-versa. Also see the related keyword *DIARY2.

A simulator command consists of the program file name (e.g., *gem.exe* shown above) followed by any number of command-line tokens. A command-line token consists of either (a) a series of unquoted contiguous non-blank characters or (b) a quoted character string. The quoting syntax depends in the command processor, e.g., CMD shell for Windows. Quotes are needed for a token that contains a blank, such as a file pathname.

The simulator recognizes a command-line argument as a **command token**, which must begin with a hyphen "-", optionally followed by a **data token**. For example, command-line argument "-f *input_data*" consists of command token "-f" and data token "*input_data*", whereas command-line argument "-wait" consist only of a command token.

A command token must start with a hyphen "-". Note that several other characters (e.g., N-dash "–" and M-dash "—") may appear to be a hyphen in your text editor, but the simulator will not recognize them as a hyphen. This issue is confused by the fact that different editors and command shells will render these other characters differently, possibly to look like a hyphen.

All command-line tokens are echoed immediately after the simulator banner, in both the .out file and the diary (.log file). This echo shows how these tokens are interpreted as individual command-line arguments. In this example, tokens "-parasol" and "8" are interpreted as one command-line argument.

```
Command-line Arguments:  -f gmflu001.dat
                         -log
                         -doms
                         -parasol 8
                         -logsum
```

Invalid or missing command-line tokens will be echoed as well, with a corresponding error message. In this example, the first two messages result from an incorrect leading character for "–log" and "—log". The last message results because "-f" requires a data token.

```
Command-line Arguments:  ùlog  **ERROR**  Leading hyphen missing
                         ûlog  **ERROR**  Leading hyphen missing
                         -f  **ERROR**  Mandatory data token missing
```

A command-line token that is unused often indicates an error in the construction of the argument list. At the end of the run, the simulator reports any unused command-line tokens. In this example, argument "-doms" is misspelled as "-dom" and so is unused.

```
Command-line Arguments:  -f gmflu001.dat
                         -log
                         -dom
. . .
Unused Command-line Arguments:  -dom
```

Use of "-perfscan" is intended to reduce pointer re-calculations in models with a large number wells/completion (perforation) changes occurring at different times/dates in the recurrent section , leading to reduction in simulation time. The option, however, needs to be used with caution as it may be counter-productive for models where there are only few wells / completion changes since there is some overhead associated with pre-scanning itself. In case all wells and completions are specified at time=0, pre-scanning is disabled, if it were specified via "-perfscan".

For additional command-line switches specific to parallel processing, see Numerical Methods Control section.

Input/Output File Names (Optional) *FILENAMES

PURPOSE:

Specify names for input and output files.

FORMAT:

```
*FILENAMES      file_types  name_options
-or-
*FILENAME       file_types  name_options
```

where file_types are one or more of:

*OUTPUT
*RESTARTIN
*RESTARTOUT
*SRFIN
*SRFOUT
*INDEX-OUT
*MAIN-RESULTS-OUT
*INDEX-IN
*MAIN-RESULTS-IN
*BINDATA-IN
*SR3-IN
*GEOMECHOUT
*SR2ASCOUT

The 'S' in *FILENAMES is optional.

For file_types *INDEX-OUT, *MAIN-RESULTS-OUT, *MAIN-RESULTS-IN, one of the following may be specified.

"
'filename'
*PROMPT

For file_type *OUTPUT one of the following may be specified.

"
'filename'
*PROMPT
*SCREEN

For file_types *INDEX-IN and *RESTARTIN one of the following may be specified.

'filename'
*PROMPT

DEFINITIONS:

*FILENAME(S)
File name keyword. The trailing 'S' is optional.

*OUTPUT

Indicates the main output file onto which formatted simulation results will be written.

*RESTARTIN

Indicates the file from which mandatory restarting data is read, if the restart reading option *RESTART is chosen. There is no default file name.

*RESTARTOUT

Indicates the file to which mandatory restarting data is written, if the restart writing option *WRST is chosen.

*SRFIN

Indicates the file from which SRF data is read, if the restart reading option *RESTART is chosen. Use *NONE if the file is not available.

*SRFOUT

Indicates the file to which Simulation Results File (SRF) data is written, if the SRF writing option *WSRF is chosen.

When a run is restarted, information which is needed to plot histories from the starting time is copied from the *SRFIN file to the *SRFOUT file. Other information (grid dumps and property plots) are not copied. Therefore, the user should not delete the *SRFIN file until the information it contains is no longer needed.

*INDEX-OUT

Indicates the index-results-file (irf) to which the simulation results ASCII data is written.

*MAIN-RESULTS-OUT

Indicates the main-results-file (mrf) to which the simulation results binary data is written.

*INDEX-IN

Indicates the file from which the Simulation Results ASCII index file is read for a restarted run.

*MAIN-RESULTS-IN

Indicates the file from which the Simulation Results main binary results file is to be read for a restarted run.

*BINDATA-IN

Subkeyword *BINDATA-IN causes the *BINARY_DATA option to use the file whose path is given by one of the following cases. **Note:** An absolute pathname starts with a directory delimiter; on Windows there may be a leading drive (letter and colon). Any other pathname is treated as relative.

1. *filename* or *BINDATA-IN is **absent**: The path is assumed to have the same root name (including directory) as the main data file, but extension ".cmgbin".
2. *filename* is **absolute**: The path is *filename*.

3. *filename* is **relative**: *filename* is relative to the directory containing the main data file and is converted internally to the corresponding absolute or relative pathname.

*SR3-IN

Indicates the file from which the Simulation Results file (SR3) is read for a restarted run

*GEOMECHOUT

Indicates the file to which the formatted output generated by the geomechanical model is written, if this option is chosen. If *GEOMECHOUT is absent, or *PROMPT is used, this output is written to the main output file given by *OUTPUT.

*SR2ASCOUT

Indicates the file to which the ASCII copy of the SR2 main binary results file is to be written.

"

Empty string, indicating that the default file name will be used.

'file_name'

A character string which is the file name, limited to 80 characters. Acceptable file names depend on the operating system being used.

*PROMPT

Indicates that the user will be prompted for this file name via the screen/keyboard, if the file is required. This is the default if the *FILENAMES keyword is missing from the data set.

*SCREEN

Indicates that this file will go to the screen (standard output device).

*NONE

Indicates that this input file is not provided. History data will start at the starting time of this run. This option is useful when the input file in question has been lost, or when the user wishes to ignore previous history data.

DEFAULTS:

If any required input/output file name has not been specified via *FILENAMES, then *PROMPT is assumed. For example, *FILENAMES implies *FILENAMES *OUTPUT *PROMPT.

If *FILENAMES is used to assign the file name but the name_options list is absent then " is used, triggering the default file names, below. For example, *FILENAMES *OUTPUT implies *FILENAMES *OUTPUT".

Default File Names:

A default file name is used only if the name_option " is specified. Defaults may be specified for any of the files independently. However, the default name of some files depends on the name chosen for other files, e.g., the data file.

A default file name has the form 'xxxxxxxx.yyy', with root 'xxxxxxxx' and suffix 'yyy'. The suffix depends on the option associated with the file, e.g., 'out' for main output, 'rst' for restart and 'srf' for SRF. The root is derived from another file name as follows: (a) discard suffix with '.' delimiter, (b) discard everything except the right- most group of alphanumeric and '-' characters, and (c) discard everything after column 80. For example, the DOS pathname '\CMG\DATA\TEST-3.DAT' has the root 'TEST-3'.

The following table summarizes the root and suffix of the default file names.

File	Root Based On	Suffix
*OUTPUT	data file name	.out
*INDEX-OUT	*OUTPUT	.irf
*RESTARTOUT	*OUTPUT	.rst
*MAIN-RESULTS-OUT	*OUTPUT	.mrf
*INDEX-IN	(no default)	
*RESTARTIN	*INDEX-IN	.rst
*MAIN-RESULTS-IN	*INDEX-IN	.mrf
*GEOMECHOUT	*OUTPUT	.geo
*SRFOUT	*OUTPUT	.srf
*SRFIN	*RESTARTIN	.srf

With this defaulting system, the user is able to perform a series of 'bootstrapped' restart runs by changing only the *OUTPUT and *INDEX-IN, *SR3-IN or *RESTARTIN file names for each run.

Binary Data File Pathname

Subkeyword *BINDATA-IN causes the *BINARY_DATA option to use the file whose path is given by one of the following cases. **Note:** An absolute pathname starts with a directory delimiter; on Windows there may be a leading drive (letter and colon). Any other pathname is treated as relative.

1. *filename* or *BINDATA-IN is **absent**: The path is assumed to have the same root name (including directory) as the main data file, but extension ".cmgbin".
2. *filename* is **absolute**: The path is *filename*.
3. *filename* is **relative**: *filename* is relative to the directory containing the main data file and is converted internally to the corresponding absolute or relative pathname.

Example 1: Data file is 'cycle.dat', so

```
*OUTPUT default is 'cycle.out'.
If the *OUTPUT default is used, then
*RESTARTOUT default is 'cycle.rst' and
*SRFOUT default is 'cycle.srf'.
```

Example 2: Data file is 'cycle.dat'. The first run segment has

```
*FILENAMES *OUTPUT 'trial1'.
*RESTARTOUT default is 'trial1.rst'
*SRFOUT default is 'trial1.srf'
```

For the second run segment, modify 'cycle.dat' to add the *RESTART keyword, and use

```
*FILENAMES *OUTPUT 'trial2.out', and
*FILENAMES *RESTARTIN 'trial1.rst'.
```

Do not rename or delete files.

```
*RESTARTOUT default is 'trial2.rst'
*SRFOUT default is 'trial2.srf'
*SRFIN default is 'trial1.srf'
```

Example 3: Data file is 'cycle.dat', so *OUTPUT default is 'cycle.out'. If the *OUTPUT default is used, then

```
*INDEX-OUT default is 'cycle.irf' and
*MAIN-RESULTS-OUT default is 'cycle.mrf'.
```

CONDITIONS:

*FILENAMES must be the first keyword(s) in the data file. If not, the user will be prompted for the file names.

File names cannot be specified in an arbitrary location in I/O Control section. They must be the first keywords in a data set.

All output files are opened with 'UNKNOWN' status and therefore are unprotected from overwriting.

All input files are opened with 'OLD' status and so must be present for the simulation to proceed.

EXPLANATION:

CMG's Simulation Results File System

The Simulation Results file system writes information in one of two formats specified by keyword *SRFORMAT: IMRF or SR3.

The IMRF format consists of two files that work together. These are the index-results-file (IRF), and the main-results-file (MRF). The graphics post-processors Results Graph and Results 3D, along with Report Writer, require the IRF and MRF files.

These files are required also for restart runs along with the restart (rst) file.

The SR3 file holds the same information as the IMRF files, but its internal format is optimized for speed of access by Results. The SR3 format will eventually replace the IMRF format.

GEM can currently produce either the IRF/MRF files, or the SRF file, as directed by the *RESULTFILE keyword.

Example:

```
** use default for output restart 'root.rst'
** Prompt for input SRF file *SRFIN by leaving
** it out of the data set.
*FILENAMES *OUTPUT    'root.out'
           *RESTARTOUT
           *SRFOUT    'name.srf'
           *RESTARTIN 'input.rst'
```

Run-Time Dimensioning (Optional) *DIM

PURPOSE:

Specify a value for a parameter controlling GEM's memory estimation for the simulation.

FORMAT:

*DIM dim_variable dim_value
where dim_variable can be one of:
*MAXPERCENT_OF_FULLYIMPLICITBLOCKS or *MDIMPL
*NULL_BLOCKS
*MAX_NONZERO_L_OR_U_ENTRIES or *MDICLU
*MDPTCN
-or-
'dim_variable' and/or 'dim_value' have been reported in an error message.

DEFINITIONS:

*DIM
Keyword indicating that a memory allocation estimate is to follow.

*MAXPERCENT_OF_FULLYIMPLICITBLOCKS, *MDIMPL
GEM treats a certain, and varying, fraction of grid blocks in a fully implicit manner throughout a simulation. The dim_value provided here should be an estimate for the maximum of the fractions seen during the simulation, expressed in percent; thus, a number between 1 and 100 is expected.

*NULL_BLOCKS
The dim_value following should be an estimate for the number of non-active blocks present in the data set. If the simulation involves a dual porosity system (*DUALPOR or *DUALPERM), the dim_value should be the smaller of the number of non-active blocks in the Matrix system and the number of non-active blocks in the fracture system.

*MAX_NONZERO_L_OR_U_ENTRIES, *MDICLU
The dim_value following should be an estimate for the maximum number of non-zero entries in the each of the L- and U-factors required by AIMSOL. This value is referred to as *MDICLU by certain GEM output messages.

*MDPTCN
Controls the amount of space required for storing inter-cell connection arrays. This sizing can affect several aspects of simulator dimensioning, including Grid Module Scratch space.

dim_variable
There are other 'dim_variable's that are used in GEM error messages. These variables have upper case names of the form '*MDXXXX' where 'XXXX' represents a short identifier for the variable.

dim_value

An integer value for the required estimate.

*DIMSUM

Appears with no dim_value following it. Causes a detailed list of dimensioning information to be written to the screen (which may be captured in a run log file) that helps evaluate memory usage.

DEFAULTS:

MDIMPL defaults to 35 (%) and *NULL_BLOCKS defaults to an internally generated count of non-active blocks. Note that the latter estimate will not account for blocks set null by cut-offs (*PVCUTOFF) or pinched-out blocks set by tolerance (*PINCHOUT-TOL). Special internal defaults are also provided for *MDICLU and *MDPTCN.

CONDITIONS:

These keywords must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

GEM uses run-time dimensioning. The simulator pre-scans portions of the data file to see how much computer memory is required to do a simulation. It then attempts to obtain as much memory as it thinks it requires from the computer, and begins the simulation if the memory is available. If there is insufficient memory, GEM will display a message and stop.

GEM is able to detect numbers of blocks, the requirements for various options (such as *DUALPOR), number of components, and the number of wells, well completions and groups, among other items. GEM is also able to estimate certain parameters, such as number of non-active blocks, and can proceed in most cases with just this information. However, there are cases where GEM's estimates are incorrect or are too large, and some additional control is required to get better estimates. Keywords have been provided which can override portions of the internal calculations, and provide more precise size information.

When GEM fails to obtain sufficient memory, it usually prints diagnostic messages suggesting which *DIM keywords to use and with which 'dim_values'. These other 'dim_variable's have upper case names of the form '*MDXXXX' where 'XXXX' represents a short identifier for the variable. Most error messages report the existing setting for the variable and a suggested, or required, new size. Then, '*DIM *MDXXXX xxx', where 'xxx' is the suggested new size, should be used. If GEM does not know what size is required, it will not report a suggested size. Then, run GEM with *DIM *DIMSUM to see what setting is currently being used for that 'dim_variable', and then use '*DIM *MDXXXX xxx' where 'xxx' has chosen to be a value that is larger than the current size (perhaps 50% larger). Some experimentation may be required.

The degree of implicitness, which affects AIMSOL linear solver array size tuning, is the most important adjustment provided by the *DIM keywords. Other *DIM parameters can also be used to tune a simulation to use less memory.

Example:

If a simulation has 552 null blocks, use

```
*DIM *NULL_BLOCKS 552
```

to optimize memory use.

Example:

Errors like the following can occur:

```
== FATAL ERROR (from subroutine: AIMPTR) == ...
Increase Solution Array Dimension MDLU.
Required Dimension = xxx (Current = yyy)
...
```

For this case, the simulator is able to estimate the required space but indicates which 'dim_variable' is affected. Rerun with '*DIM *MDLU xxx', to acquire sufficient space.

Example:

Errors like the following can also occur:

```
== FATAL ERROR (from subroutine: GRCNC7) == ...
Insufficient Connection space.  Rerun with increased space.
Use the *DIM *MDPTCN keywords.
...
```

For this case, the simulator is unable to estimate the required space. Rerun with *DIM *DIMSUM to discover the current setting for *MDPTCN, and then rerun with '*DIM *MDPTCN xxx', where 'xxx' has been chosen to be a somewhat larger value than was used in the failed run (such as 50% larger).

Example:

Use the *WPRN *ITER *ALL keyword sequence (see below) to print, in the standard output file, information about the current Degree of Implicitness and linear solver dimensioning. The information appears as follows:

```
--- Degree of implicitness = ...
Solver Array    Dimensions:
Jacobian Array  Dimension = ... (maximum MDALP= ...)
Solution Vector Dimension = ... (maximum MDV  = ...)
Diagonal Array  Dimension = ... (maximum MDDD = ...)
LU Factors      Dimension = ... (maximum MDLU = ...)
...
```

This information can be used (perhaps in a restarted run) to determine the Degree of Implicitness required by GEM at some time. The value provided above (when converted from a fraction to a percent) can be entered using '*DIM *MDIMPL xx' to carry on a run that stopped due to memory size limitations due to a large degree of implicitness.

Interrupt and CTRL-C Control (Optional) *INTERRUPT

PURPOSE:

*INTERRUPT indicates how GEM is to respond when the user interrupts the simulation run by simultaneously pressing the "CTRL_Shift" and "C" keys; that is, by entering CTRL-C.

FORMAT:

```
*INTERRUPT      (*RESTART-STOP)
                (*WRST-CONT)
                (*STOP)
                (*INTERACTIVE)
```

DEFINITIONS:

*RESTART-STOP
: This subkeyword specifies that the current timestep be completed, all specified output data be written to all output files and a restart record be written if restart writing has been enabled (*WRST 0 has not been used).

*WRST-CONT
: This subkeyword specifies that a restart record be written (when possible) and the simulation continue.

*STOP
: This subkeyword specifies that the simulation run be terminated immediately, without completing the current timestep. However, all files are to be properly closed.

*INTERACTIVE
: This subkeyword specifies that the user will be prompted to select further action to be taken.

DEFAULTS:

Optional keyword. If it is not specified, then the user is prompted interactively for further action when a CTRL-C is entered.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group.

EXPLANATION:

This keyword allows different action to be taken when GEM receives a "CTRL-C" interrupt (or a "kill -2 <PID_Number>" signal is received on a UNIX platform). In particular, a clean simulator termination may be effected without leaving unclosed files. Also, this facility allows belated selection of restart writing options.

Note that the initial response from the simulator after entering "CTRL-C" may only occur after the current timestep has been completed. Also, repeated "CTRL-C" keystrokes, or using "CTRL-BREAK" on an PC, may cause an undesired immediate run termination, leaving improperly closed files.

Project Main Title (Optional) *TITLE1

PURPOSE:

*TITLE1 is used for project identification.

FORMAT:

*TITLE1 string

DEFINITIONS:

string
Any alphanumeric character string enclosed in single quotes. (40 characters maximum)

DEFAULTS:

Optional keyword. Defaults to blanks.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

This keyword identifies an alphanumeric character string used for project identification. It will appear both in printed output and in the Simulation Results File.

The string must be on the line immediately following the keyword *TITLE1. Any character string with embedded blanks or commas must be enclosed with single quotes.

Example:

```
*TITLE1
'DUAL POROSITY/DUAL PERMEABILITY RUN NO.1'
```

Project Second Title (Optional) *TITLE2

PURPOSE:

*TITLE2 is used for project identification. It is used in addition to *TITLE1 to provide a second line for project identification.

FORMAT:

*TITLE2 string

DEFINITIONS:

string
Any alphanumeric character string enclosed in single quotes. (80 characters maximum)

DEFAULTS:

Optional keyword. Defaults to blanks.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

This keyword identifies an alphanumeric character string used for project identification. It will appear in the printed output but not in the Simulation Results File.

The string must be on the line immediately following the keyword *TITLE2. Any character string with embedded blanks or commas must be enclosed with single quotes.

Examples:

```
*TITLE1
'DUAL POROSITY/DUAL PERMEABILITY RUN NO.1'
*TITLE2
'Run by A.B. staff, December 16, 1988. C.D. Co.'
```

Project Third Title (Optional) *TITLE3

PURPOSE:

*TITLE3 is used for project identification. It is used in addition to *TITLE1 and *TITLE2 to provide a third line for project identification.

FORMAT:

*TITLE3 string

DEFINITIONS:

string
Any alphanumeric character string enclosed in single quotes. (80 characters maximum)

DEFAULTS:

Optional keyword. Defaults to blanks.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

This keyword identifies an alphanumeric character string used for project identification. It will appear in the printed output but not in the Simulation Results File.

The string must be on the line immediately following the keyword *TITLE3. Any character string with embedded blanks or commas must be enclosed with single quotes.

Examples:

```
*TITLE1
'DUAL POROSITY/DUAL PERMEABILITY RUN NO.1'
*TITLE2
'Run by A.B. staff, December 16, 1988. C.D. Co.'
*TITLE3
'IMEX 4.0; 4200 grid blocks; variable thickness'
```

Case Identification (Optional) *CASEID

PURPOSE:

*CASEID is used to identify specific case runs. It is also used in the Simulation Results File to identify data curves for plots.

FORMAT:

*CASEID string

DEFINITIONS:

string
Any alphanumeric character string enclosed in single quotes. (8 characters maximum)

DEFAULTS:

Optional keyword. Defaults to blanks.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

This keyword identifies an alphanumeric character string used to name specific case runs. It will also be output to the Simulation Results File, where it is used as the default legend to identify data curves for plots. The string must be on the same line as the *CASEID keyword.

Example:

```
*CASEID 'Case 21c'
```

Check Only (Optional) *CHECKONLY

PURPOSE:

*CHECKONLY indicates that the entire data set, including the well data is to be checked for format errors, but the time-stepping will not be done.

FORMAT:

*CHECKONLY

DEFAULTS:

Optional keyword. No check on the well data is performed unless this keyword is present.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group. It should be placed as early as possible in the data set. This keyword cannot be used in restart runs where the keyword *RESTART, *RESTIME, or *RESDATE is present.

EXPLANATION:

Normally (i.e., without *CHECKONLY), syntax, storage allocation and range checking of data is done as it is read. Initialization data (all but recurrent) is processed at the beginning of the run, so errors in that part of the data are detected and reported immediately.

However, recurrent data is read when it is needed, as the simulation time progresses. Therefore, errors in recurrent data will be detected and reported later in the run. This can be inconvenient for large runs. Keyword *CHECKONLY allows you to scan your entire data set to the end very quickly so that data errors are detected and reported immediately.

In fact, the only part of the simulation not done in scan mode is the timestep calculation. This means that all the reading, storage allocation, checking, echoing, printing and Simulation Results dumping are done. For example, you can view the initial conditions, and hence your grid, in RESULTS from a Simulation Results file generated by a scan mode run.

It is recommended that you keep near the top of your data set a line consisting of *CHECKONLY. Normally this keyword is disabled (commented out). You can quickly enable the keyword and run the data in scan mode. Remember to disable the keyword before submitting the actual run.

A license is not required to run GEM in scan mode, allowing you to validate data while your licenses are occupied running simulations.

Diary Format (Optional) *DIARY, *DIARY2

PURPOSE:

*DIARY specifies the type of timestep information printed to the screen and to the output file during a simulation. *DIARY2 has the same functionality as *DIARY and can be used additionally to create a secondary log file.

FORMAT:

```
*DIARY     (*CHANGES)
           (*CHANGES-UNCONV)
           (*CHANGES-EXP)
           (*WELL-INFO)
           (*WELL-INFO-BRIEF)
           (*WELL-INFO-EXP)
*DIARY2    (*CHANGES)
           (*CHANGES-UNCONV)
           (*CHANGES-EXP)
           (*WELL-INFO)
           (*WELL-INFO-BRIEF)
           (*WELL-INFO-EXP)
```

DEFINITIONS:

*CHANGES

This option specifies that after each timestep a line specifying the size and location of the maximum changes in pressure, saturation, composition, during the timestep; average material balance error, and cumulative solver failures till this timestep is printed to the terminal screen and to the output file.

*CHANGES-UNCONV

This option specifies that after each timestep a line specifying the size and location of the maximum changes in pressure, saturation, composition, during the timestep; average material balance error, and cumulative solver failures till this timestep; and convergence status flags for the current timestep is printed to the terminal screen and to the output file. The convergence status flag include, primary variables, residual of simulation equations, well residuals, constraint equation residuals; and geomechanics convergence flag for two-way geoemechanical coupling.

*CHANGES-EXP

This option specifies that after each timestep a line specifying the size and location of the maximum changes in pressure, saturation, composition, during the timestep; average material balance error, and cumulative solver failures till this timestep is printed to the terminal screen and to the output

file. Most of the columns are printed in engineering (exponential) format, as in versions prior to 2009.01.

*WELL-INFO

This option specifies that after each timestep a line specifying the current field oil, gas, and water production rates, the water cut and GOR for the field, the field gas and water injection rates, the average reservoir pressure; average material balance error, cumulative solver failures till this timestep; and the number of wells on production, water/gas injection and shut-in is written to the terminal screen and to the output file.

*WELL-INFO-BRIEF

This option specifies that after each timestep a line specifying the current field oil, gas, and water production rates, the water cut and GOR for the field, the field gas and water injection rates, the average reservoir pressure; average material balance error, and cumulative solver failures till this timestep is written to the terminal screen and to the output file.

*WELL-INFO-EXP

This option specifies that after each timestep a line specifying the current field oil, gas, and water production rates, the water cut and GOR for the field, the field gas and water injection rates, the average reservoir pressure; average material balance error, and cumulative solver failures till this timestep is written to the terminal screen and to the output file. Most of the columns are printed in engineering (exponential) format, as in versions prior to 2009.01.

DEFAULTS:

Optional keyword. Default is *CHANGES.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL or in WELLS AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword allows the user to choose the type of information which is printed during a run to allow the user to monitor the progress of the run. The information printed under the *CHANGES format is appropriate for detecting problems in the simulation; the information printed under the *WELL-INFO is more useful for monitoring the effectiveness of the recovery scheme as the run progresses.

*CHANGES-UNCONV provides the status of convergence flags (0=converged, 1=unconverged) which can be useful in adjusting the convergence tolerances in a run, and thereby reducing the average material balance error and getting more realistic physical results from a simulation run. Also see keyword *CONVERGE under chapter **Numerical Methods Control.**

*CHANGES-EXP and *WELL-INFO-EXP are useful when you wish to use the values written after each timestep for your own customized post-processing. None of the fields are ever blanked out in these two cases. The other *DIARY sub-keywords may zero or blank-out certain fields, for instance if the absolute value of a quantity is less than 1.E-10. In case of

convergence flags, if a particular convergence criterion is not in use, corresponding status is blanked out.

The *DIARY2 keyword allows creation of a secondary log file that has root name of the data file and an extension ".log2". It is particularly useful when both changes-based as well as well-performance based summaries are necessary for analysis. Note that the secondary log file does not carry warning or error messages that are written essentially to the regular log and out files. Keyword *DIARY2 is not supported by Builder 2010.10. See also command-line input "*-log2*".

Examples:

```
*DIARY *WELL-INFO
*DIARY2 *CHANGES
```

Input Data Units (Optional) *INUNIT

PURPOSE:

*INUNIT specifies the input data units.

FORMAT:

*INUNIT (*SI)
(*FIELD)

DEFINITIONS:

*SI
This option specifies SI units for input data.

*FIELD
This option specifies FIELD units for input data.

DEFAULTS:

Optional keyword. Default is *SI.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

Input data may be entered using one of the above options shown in the following Units Table. Display in the GEM output will be in the same units. In certain cases the display from the RESULTS graphical display system will be in slightly different units, but the units will always be identified.

IMPORTANT NOTE: Most of the units for properties entered in the Component Properties section are independent of the setting made here for *INUNIT. Check that section for appropriate unit descriptions and ensure that the correct conversions have been made, if required.

Example:

```
*INUNIT *FIELD
```

UNITS TABLE

	*SI	*FIELD
Time	days	days
Liquid Volumes	m3	ft3 or bbl
Gas Volume	m3	ft3
Pressure	kPa	psi
Absolute Permeability	md	md
Viscosity	cp	cp
Mass Density	kg/m3	lbm / ft3
Molar Density	gmol/m3	gmol/ft3
Length, Distance	m	ft
Porosity	fraction	fraction
Saturation	fraction	fraction
Temperature	deg C	deg F
Critical Pressure	atm	atm
Critical Temperature	deg K	deg K
Critical Volume	m3/(k-gmol)	m3/(k-gmol)

Load Balancing Status Printing for Non-Solver Tasks (Optional)

***LBPRINT**

PURPOSE:

*LBPRINT controls the printing of load-balancing status for parallel runs for non-solver tasks, like Jacobian building, variable updates, convergence checking, etc.

FORMAT:

*LBPRINT	(*ON) (*OFF)

DEFINITIONS:

*ON

Turn on the printing of load balancing status.

*OFF

Turn off the printing of load balancing status.

DEFAULTS:

Optional keyword. Default is *OFF. *LBPRINT on a line by itself (i.e., without any following sub-keywords) is treated as *LBPRINT *ON.

CONDITIONS:

This keyword, if present, must appear in the INPUT/OUTPUT CONTROL keyword group, or/and in the WELL AND RECURRENT DATA keyword group. The keyword is meant to be used in presence of *DPLANES keyword or –doms command-line option. Otherwise, the keyword is ignored.

EXPLANATION:

For parallel runs load-balancing status for non-solver tasks could be helpful in using the available computing resources optimally to reduce the run-time (see *JLOADBAL and associated input in Numerical Methods Control section). Initial values are always printed at the start of time-stepping. Subsequent printing of load-balancing updates can optionally be controlled by keyword *LBPRINT.

Note that for a model with fast changing implicitness pattern or with a very large number of grid-blocks the load-balancing output could be large if *LBPRINT *ON is selected and not switched *OFF subsequently. To manage the printing use *LBPRINT *ON or *OFF in WELL AND RECURRENT DATA section at desired date/time inputs.

Example:

```
*LBPRINT *ON
```

Level and Class Structure Printing for Parallel Linear Solver (Optional) *LCSPRINT

PURPOSE:

*LCSPRINT controls the printing of level and class structure for parallel linear solver, PARASOL.

FORMAT:

*LCSPRINT	(*ON) (*OFF)

DEFINITIONS:

*ON
Turn on the printing of level and class structure.

*OFF
Turn off the printing of level and class structure.

DEFAULTS:

Optional keyword. Default is *OFF. *LCSPRINT on a line by itself (i.e., without any following sub-keywords) is treated as *LCSPRINT *ON.

CONDITIONS:

This keyword, if present, must appear in the INPUT/OUTPUT CONTROL keyword group, or/and in the WELL AND RECURRENT DATA keyword group. The keyword is meant to be used in presence of *PARASOL keyword or –parasol command-line option. Otherwise, the keyword is ignored.

EXPLANATION:

For parallel linear solver the level and class structure of parallel domains could be useful to analyze the run. Initial values always printed at the start of time-stepping. Subsequent printing of level and class structure updates can optionally be controlled by keyword *LCSPRINT.

Note that for a model with fast changing implicitness pattern or with a very large number of grid-blocks the level and class structure output could be quite overwhelming if *LCSPRINT *ON is selected and not switched *OFF subsequently. To manage the printing use *LCSPRINT *ON or *OFF in WELL AND RECURRENT DATA section at desired date/time inputs.

Example:

```
*LCSPRINT *ON
```

Data Range Checking (Optional) *RANGECHECK

PURPOSE:

*RANGECHECK controls the data range check feature.

FORMAT:

```
*RANGECHECK     (*ON)
                (*OFF)
```

DEFINITIONS:

*ON

Turn on the range check feature.

*OFF

Turn off the range check feature.

DEFAULTS:

Optional keyword. Default is *ON.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

EXPLANATION:

Most input data is examined to determine if it is within an expected range of numbers. Specifying *RANGECHECK *OFF will disable the non-critical data range checking.

*RANGECHECK *OFF also will suppress the printing of all "warning" messages. Error messages will always be printed.

Example:

```
*RANGECHECK *OFF
```

Maximum Number of Error Messages (Optional) *MAXERROR

PURPOSE:

*MAXERROR specifies the maximum number of error messages before the simulation terminates.

FORMAT:

*MAXERROR num

DEFINITIONS:

num
An integer indicating the maximum number of error messages.

DEFAULTS:

Optional keyword. Default is 20.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file. The value for num must be between 1 and 100.

EXPLANATION:

During data input, when a syntax or range error occurs, the simulator will print an error message, then attempt to continue scanning the input data.

Simulation is stopped if there are errors in the initialization data. Thus, initialization is not done and the well data is not read. If initialization is done, but there are errors in the well data, then simulation is stopped at this point. In both cases, the run is terminated before the *MAXERROR value is reached.

Some syntactic errors may result in the simulator getting so confused that it would print out dozens of meaningless error messages. Thus, if the number of error messages exceeds the maximum number of errors, the simulator immediately stops.

Example:

```
** Stop simulation if 50 errors encountered.
*MAXERROR 50
```

Separator Stage Composition Output (Optional) *WELSEP-ICPRT

PURPOSE:

*WELSEP-ICPRT controls the printing of separator stage compositions to the output file for the production wells.

FORMAT:

*WELSEP-ICPRT (*ON)
(*OFF)

DEFINITIONS:

*ON
Turn on printing of the well separator stage compositions.

*OFF
Turn off printing of the well separator stage compositions.

DEFAULTS:

Optional keyword. If *WELSEP-ICPRT does not appear in the data file the default is *OFF. If *WELSEP-ICPRT appears without the sub keyword, the default is *ON.

CONDITIONS:

This keyword, if present, must appear in the INPUT/OUTPUT CONTROL keyword group, or in the WELL and RECURRENT data group.

EXPLANATION:

The liquid and vapor stream compositions from different stages of production well separators may be of interest in certain cases, which can be printed out to the output file using keyword *WELSEP-ICPRT. Note that this keyword is applied to the separators of all of the production wells. For a model with a large number of producers, the output could be overwhelming in which case *WELSEP-ICPRT *OFF could be used in the recurrent data section after the period of interest.

Example:

```
*WELSEP-ICPRT *ON
```

Restart Record Writing Frequency (Optional) *WRST, *REWIND

PURPOSE:

*WRST controls the frequency of writing restart records to the output restart file.

*REWIND controls the rewinding of the restart file just before writing a new restart record.

FORMAT:

*WRST (*freq* | *TIME | *TNEXT | (elkwd *eltime*)), where
elkwd = any one of *ELAPSED-HRS, *ELAPSED-MINS, or *ELAPSED-SECS

*REWIND (*num*)

DEFINITIONS:

freq

A non-negative integer to specify the frequency (timesteps). Restart data are written every freq timesteps.

If *freq* is zero, restart records are not written and no output restart file is prepared. The restart frequency MUST then remain 0 throughout the run, and no other restart writing commands are permitted.

If freq is positive, the output restart file is opened and *freq* may be redefined in the recurrent data though another appearance of *WRST. If *freq* is positive, a restart record is written after initialization (timestep 0) and also after the final timestep if the simulation is terminated by the simulator, and also every freq timesteps in between.

*TIME

Write data to the restart file at every time specified by subsequent recurrent *TIME or *DATE keywords.

*TNEXT

Write data to the restart file at the next *TIME or *DATE keywords, and then suspend restart writing until another *WRST keyword is encountered (in the WELL AND RECURRENT DATA section).

*ELAPSED-HRS *eltime*

Write data to the restart file when simulation elapsed time is equal or exceeds *eltime* hours from the elapsed time when last restart was written. The simulator (restart writing) clock is refreshed to zero after writing the restart. The next restart is written again when simulation time exceeds *eltime* hours and so on till end of the simulation. If necessary, the keyword may be used in the WELL AND RECURRENT DATA section, to input a different *eltime*.

*ELAPSED-MINS *eltime*

Similar to *ELAPSED-HRS above, but *eltime* is in minutes.

*ELAPSED-SECS *eltime*

Similar to *ELAPSED-HRS above, but *eltime* is in seconds.

A zero or negative entry for *eltime* is treated as a signal for cancellation of further restart writing based on *ELAPSED-HRS, *ELAPSED-MINS, or *ELAPSED-SECS.

num

Frequency of rewinding the restart file, where *num* is the maximum number of restart records allowed to accumulate on the restart file before it is rewound. *num* must be greater than 0. If *num* is 1, then only the last restart is ever available.

DEFAULTS:

Optional keyword. If *WRST does not appear, the default is to write restart data at time zero and at the last simulation time.

If *REWIND is not present the restart file is never rewound. If '*num*' is absent, then *num* = 1 is assumed.

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the frequency of records in the restart file may be changed during the simulation, provided that *WRST 0 does not appear initially.

A zero or a negative entry for *eltime* after the *ELAPSED-xxxx keyword signals cancellation of further restart writing based on *ELAPSED-xxxx.

EXPLANATION:

Restart records store a "snap-shot" of reservoir conditions at a particular time. Using a restart record you can restart a simulation from some midpoint in a run. This allows you to try different well production strategies, produce more detailed output, or make other changes without the expense of repeating the entire simulation run. Note that a large number of values are written to the restart file for each of the “snap-shots”. It is necessary that you write restarts only at the frequency, time, and elapsed times that are essential for your future runs. Excessive restart writing can adversely impact on your simulation time and can easily fill the disk space.

Multiple occurrence of *WRST keyword lines:

Multiple *WRST keyword lines are permitted but these must be used with caution. When such lines appear excepting with sub-keyword *ELAPSED-xxxx, the last *WRST line overrides the previous ones. If *ELAPSED-xxxx happens to be on one or more of the *WRST keyword lines in presence of other *WRST lines, then the last *WRST line without *ELAPSED-xxxx is honored together with the last *WRST line with *ELPASED-xxxx.

GEM performs its restarts from binary files which carry the .RST extension and which are written solely for restarts. Restart information is NOT contained in the .IRF/.MRF pair or SR3 for GEM but these files (IRF/MRF or SR3) are required to identify parent runs. The user must take care to preserve .RST files if restarts are to be done later. GEM restart files cannot be moved from one platform to another in general.

Note that if *WRST 0 is specified, no restart file will be opened and *WRST cannot be redefined later in the recurrent (well) data section. If no restart record is required initially, but required later during the run, use

```
*WRST   100000
```

instead and redefine *WRST at the appropriate time, or use *WRST *TNEXT.

For parent runs that write pressure (and temperature) information to Simulation Results or RST file for subsequent geomechanics post-processing (GEOPOSTPRO), it is recommended not to use *WRST 0.

Restart records can be written after entering a "Ctrl-C" provided that *WRST 0 does not appear in the data file.

Examples:

```
** Write restart record every 10 timesteps.
*WRST 10
** Write restart at the every time change.
*WRST
** Write restart at every one hour.
*WRST *ELAPSED-HRS 1
** rewind the output restart file before
** writing the sixth restart record.
*REWIND 5
```

Restart Timestep (Optional) *RESTART, *RESTIME, *RESDATE

PURPOSE:

*RESTART, *RESTIME or *RESDATE indicates that this is a restarted run.

FORMAT:

*RESTART (*nstart*)

*RESTIME *restime*

*RESDATE *year month day*

DEFINITIONS:

nstart

Specify the timestep number after which to restart the simulation. Command-line argument '-restart' is another way to specify *RESTART. The *nstart* corresponding to a particular time step may be obtained from the table of restarts available from the previous run that is printed at the end of the output file of the previous run.

restime

Specify time (days) of the timestep from which to restart the simulation. This option works best when *restime* corresponds to a simulation reference time specified via *TIME or *DATE in the recurrent data section of the previous run. Also, *restime* may be a non-reference time but it must match the time of the target restart record within the first 7 decimal digits. Use *RESTART when *RESTIME picks incorrectly from a group of records whose times do not differ in the first 7 digits. Command-line argument '-restime' is another way to specify *RESTIME. The *restime* corresponding to a particular time step may be obtained from the table of restarts available from the previous run that is printed at the end of the output file of the previous run.

Year month day

Specify the date of the timestep from which to restart the simulation. *year* is the integer year not exceeding 5000000. *month* is the integer month in the range 1 to 12. *day* is the day-of-month in the range 0 to 31, plus the decimal fraction-of-day starting from midnight. For example, noon on Aug 19, 1990 is represented as "*RESDATE 1990 8 19.5". Command-line argument '-resdate' is another way to specify *RESDATE.

This option works best when this date corresponds to a simulation reference time specified via *DATE in the recurrent data section of the previous run. The fractional-day date of a particular time step may be obtained from the table of restarts available from the previous run that is printed at the end of the output file of the previous run.). Also, this date may be a non-reference time but it must match the date of the target restart record within the first 12 decimal digits (first 4 digits for *year*, next 2 digits for *month*, last 6 digits for

day). Use *RESTART when *RESDATE picks incorrectly from a group of records whose dates differ less than this.

DEFAULTS:

Optional keyword. If this keyword is not present in the input data file, then no restart records are read and the simulation begins at time zero.

If *RESTART, *RESTIME or *RESDATE is present, but is not followed by a timestep number, time or date the default is to restart at the last timestep in the incoming restart file.

If *RESTART, *RESTIME or *RESDATE (or their command-line argument) is present, then the restart files denoted by *INDEX-IN and *MAIN-RESULTS-IN are required and must contain the restart record corresponding to the specified time or timestep number.

A fatal error message is issued if a combination of *RESTART, *RESTIME or *RESDATE is present, or when each appears multiple times in the data file. Only one of them is allowed to be used.

*Keyword entries *RESTART, *RESTIME or *RESDATE, if present in data file, will get precedence over their command line option counterparts.*

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group.

EXPLANATION:

Restart records store a "snap-shot" of reservoir conditions at a particular time, namely the end of the timestep at which the restart record was written. These keyword permit GEM to restart from a previously written restart record stored in a .RST file. If a record for the actual timestep number is not in the file, GEM attempts to use one nearby.

Example 1:

```
** Restart at the 10th timestep.
*RESTART 10

** Restart at the last timestep in the restart file.
*RESTART
```

Example 2 :

```
** Restart at timestep corresponding to 731.1205 days.
*RESTIME 731.1205

** Restart at the last timestep in the restart file.
*RESTIME
```

Example 3:

```
** Restart at timestep corresponding to date 1995 05 29.1
*RESDATE 1995 05 29.1

** Restart at the last timestep in the restart file.
*RESDATE
```

Output Printing Frequency (Optional) *WPRN

PURPOSE:

*WPRN controls the frequency of writing data to the output print files.

FORMAT:

*WPRN	(*WELL)	(freq)
		(*TIME)
		(*TNEXT)
	(*GRID)	(freq)
		(*TIME)
		(*TNEXT)
	(*ITER)	(*ALL)
		(*MATRIX)
		(*NEWTON)
		(*BRIEF)
		(*NONE)

DEFINITIONS:

*WELL freq

Write well results to the output file every freq timesteps, where freq is an integer. If freq is zero, no well results are written.

*WELL *TIME

Write well results to the output file at every time specified by subsequent recurrent *TIME or *DATE keywords in the input file.

*WELL *TNEXT

Write well data to the output file at the next *TIME or *DATE keywords, and then suspend writing until another *WPRN keyword is encountered (in the WELL AND RECURRENT DATA section).

*GRID freq

Write grid results to the output file every freq timesteps, where freq is an integer. If freq is zero, no grid results are written.

*GRID *TIME

Write grid results to the output file at every time specified by subsequent recurrent *TIME or *DATE keywords in the input file.

*GRID *TNEXT

Write grid data to the output file at the next *TIME or *DATE keywords, and then suspend writing until another *WPRN keyword is encountered (in the WELL AND RECURRENT DATA section).

*ITER *ALL

Provide full details of matrix solution and Newton iterations at every timestep. Also provides Degree of Implicitness and AIMSOL dimensioning information.

*ITER *MATRIX

Provide details of matrix solution only at every timestep.

*ITER *NEWTON

Provide details of Newton iterations at every timestep.

*ITER *BRIEF

Provide a summary of timestep convergence behavior.

*ITER *NONE

Provide no information on timestep behavior.

DEFAULTS:

Grid results are written every *TIME or *DATE card. Well results are written every time Grid results are written. The default for *ITER is *BRIEF.

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail in the print file may be changed during the simulation.

EXPLANATIONS:

Examples:

```
** Write well results at every time change.
*WPRN *WELL *TIME

** Write grid results every 10 timesteps.
*WPRN *GRID 10

** Write details of Newton iterations
** at every timestep.
*WPRN *ITER *NEWTON
```

Items in the Output Print File (Optional) *OUTPRN

PURPOSE:

*OUTPRN identifies what information is written to the output print file.

FORMAT:

*OUTPRN	(*WELL)	(*ALL)	
		(*RESERVOIR)	
		(*LAYER)	
		(*BRIEF)	
		(*PSPLIT)	
		(*PSPLITNOCOMP)	
		(*AQPHASEPROP)	
	(*GRID)	(*NONE)	
		(grid_list)	
	(*RES)	(*ALL)	
		(*NONE)	
	(*FLUX_SECTOR)	(*ALL)	(mass-mole selector)
		(*INFLUX)	(sc-rc selector)
		(*INTERSEC)	(ijk-con selector)
		(*NONE)	

DEFINITIONS:

*WELL

This subkeyword specifies that well results will be written to the output print file.

*GRID

This subkeyword specifies that grid results will be written to the output print file.

*RES

This subkeyword specifies that input reservoir properties will be printed at the start of the simulation run.

*FLUX_SECTOR

This subkeyword specifies that flux sector summary will be written to the output print file. The output frequency is controlled by *WPRN *WELL. Also see **Flux-Sector Reports**, below.

*ALL

Write all possible variables to the output print file. It cannot be used together with *GRID. When used after *WELL, well information at surface conditions, reservoir conditions, and on a layer-by-layer basis are printed. For *FLUX-SECTOR option, it means both influx and inter-sector flux will be printed out.

*RESERVOIR

Write well phase rates at surface and reservoir conditions.

*LAYER

Write well rate information at surface conditions and on a layer-by-layer basis.

*BRIEF

Write a summary of well variables at surface conditions to the output print file. With this setting for *OUTPRN *WELL, sector and flux-sector information is NOT printed to the output file; for all other settings, sector information is printed and flux-sector information is printed according to *OUTPRN *FLUX_SECTOR specifications.

*PSPLIT /PSPLITNOCOMP

Production split information is output, with optional compositions.

*AQPHASEPROP

Aqueous phase stream properties table is output for all wells (production and injection) by layer and for the entire well. Number of moles of each component (including soluble gaseous hydrocarbon species as CO_2 and ionic species such as Ca++) produced/injected, molality or concentration of each component as well as thermodynamic and transport properties such as salinity, stream density and stream viscosity at reservoir and at separator conditions is written out to the main output file (file with the extension .out). For these calculations, composition of the aqueous stream, pressure and temperature must be known. For calculations at reservoir conditions, the pressure and temperature for a producing/injecting layer is the corresponding block pressure and temperature. For a producing layer the composition of the aqueous stream is the corresponding aqueous phase composition in the grid block. For an injection well the composition is the same for all injection layers and is specified by the user via the INCOMP keyword. For calculations at surface conditions, pressure and temperature are STP (1 atm and 15.6 deg C). For calculations at surface conditions for a production well, the soluble gaseous hydrocarbon species such as CO_2, CH_4 are assumed to partition to the hydrocarbon phase. Therefore the surface aqueous stream will not have any soluble gaseous hydrocarbon species. If water vaporization is turned ON then the water component in the vapor stream will partition to the aqueous phase. Therefore the number of moles of water component produced at surface will be larger than the number of moles of water component produced at reservoir condition. Ionic species such as Ca++ always remain in the aqueous phase.

*INFLUX

Influx statistics will be written to the output print file.

*INTERSEC

Inter-sector flux statistics will be written to the output print file.

*NONE

No variables will be written to the output print file.

mass-mole selector

*MASS	Flux sector summary contains mass fluxes of individual components
*MOLE	Flux sector summary contains molar fluxes of individual components

If neither present, component fluxes are not reported.

sc-rc selector

*SC	Flux sector summary contains phase flux values in surface condition
*RC	Flux sector summary contains phase flux values in reservoir condition

If neither present, *RC is selected.

ijk-con selector

*SUM	Flux sector summary contains summation of connections in all three direction
*ICON	Flux sector summary contains flow across connections in I direction
*JCON	Flux sector summary contains flow across connections in J direction
*KCON	Flux sector summary contains flow across connections in K direction

If none of above selectors present, *SUM is selected.

grid_list

A series of character strings to identify which variables to write to the output print file.

The grid_list for *GRID follows where 'comp_name' is a valid component name (see *COMPNAME):

*PRES	Reservoir Pressure
*DROP	Pressure drop from time zero
*DATUMPRES	Block pressure corrected to datum depth
*IMPL	Implicit-explicit state map
*SO	Oil saturation
*SG	Gas saturation
*SW	Water saturation
*TSO	Trapped oil saturation
*RHOO	Oil molar density
*RHOG	Gas molar density
*RHOW	Aqueous molar density
*FRG	Gas phase mole fraction in o/g system
*PCW	Capillary pressure for water
*PCG	Capillary pressure for gas
*KRINTER	Relative permeability interpolant value
*KRO	Oil relative permeability
*KRG	Gas relative permeability
*KRW	Water relative permeability
*VISO	Oil viscosity
*VISG	Gas viscosity
*VISW	Water viscosity
*MWO	Oil molecular weight
*MWG	Gas molecular weight
*DENO	Oil mass density
*DENG	Gas mass density
*DENW	Water mass density

*SALIN	Aqueous phase salinity (molality)
*SATP	Saturation pressure (an approximate estimate, see explanation)
*SIG	Interfacial tension between oil and gas
*SIGMAGW	Interfacial tension between gas and water (used in Foam models). A negative value indicates absence of foam in the block.
*SIGMAOW	Interfacial tension between oil and water (used in non-Foam models)
*PSIG	Pseudo interfacial tension between oil and gas
*TGIP	Total Gas in Place, SC
*SOLID	Solid mass and adsorbed mass; resistance factor
*ADS ‘comp_name’	Adsorbed gmol/grid-block volume of ‘comp_name’
*VELOCRC	Phase velocities at reservoir conditions
*Z ‘comp_name’	Global mole fraction of component ‘comp_name’
*X ‘comp_name’	Mole fraction of component ‘comp_name’ in oil phase
*Y ‘comp_name’	Mole fraction of component ‘comp_name’ in gas phase
*W ‘comp_name’	Mole fraction of component ‘comp_name’ in aqueous phase
*K ‘comp_name’	Gas/oil k-value of component ‘comp_name’
*ZALL	Global mole fraction of hydrocarbon components
*XALL	Mole fraction of all component s in oil phase
*YALL	Mole fraction of all components in gas phase
*WALL	Mole fraction of all components in the aqueous phase
*CAPN	Capillary number, gas-oil (output only when *VELDEPRP is used)
*CAPNGW	Capillary number, gas-water; for foam models with *IFTTABLE. A negative value indicates absence of foam in the block.
*CAPNOW	Capillary number, oil-water; for rel-perm interpolation for non-foam models with *IFTTABLE
*DIFFUO ‘comp_name’	Effective oil phase diffusion coefficient for ‘comp_name’
*DIFFUG ‘comp_name’	Effective gas phase diffusion coefficient for ‘comp_name’
*DIFFUW ‘comp_name’	Effective aqueous phase diffusion coefficient for ‘comp_name’
*POROS	Current porosity
*PERM	Current permeabilities
*PERMEFF	Current effective permeabilities
*PERMINT	Current intrinsic permeabilities
*TEMP	block temperature
*SGHYS	Hysteresis – Gas saturation corresponding to reversal point from drainage to imbibition; starting point of drainage to imbibition scanning curve
*SGRHYS	Hysteresis – Maximum trapped gas saturation, end point of drainage to imbibition scanning curve; gas saturation where gas relative permeability is zero
*SGDTHY	Hysteresis – Dynamic trapped gas saturation
*RFO	Oil permeability resistance factor
*RFG	Gas permeability resistance factor
*RFW	Water permeability resistance factor
*IPSTCA	Parasol Class Map
*ZFACO	Z compressibility factor of oil phase
*ZFACG	Z compressibility factor of gas phase
*COMPRO	Compressibility of oil phase
*COMPRW	Compressibility of water phase

*COMPRG	Compressibility of gas phase
*COMPRT	Total compressibility of all fluid phases
*OILH	Oil phase enthalpy
*GASH	Gas phase enthalpy
*WATH	Water phase enthalpy
*FLUIDH	Fluid enthalpy(where fluid enthalpy is (phase saturation times phase molar density) weighted average of enthalpies of all phases present in a block)
*MOLALITY ‘comp_name’	Aqueous phase molality of gaseous or aqueous species
*MINERAL ‘comp_name’	Change in moles of mineral components
*DPORMNR	Porosity change due to mineral reactions
*PH	pH
*EQVFRIEX ‘comp_name’	Equivalent fraction of comp_name
*EQVFRIEXN ‘comp_name’	Equivalent fraction of comp_name determined using maximum CEC over all the grid-blocks.
*ACTIV ‘comp_name’	Activity of aqueous component ‘comp_name’
*ACTIVCOEF ‘comp_name’	Activity coefficient of aqueous component ‘comp_name’
*LSATINDM ‘comp_name’	Log10 (saturation index) of mineral component ‘comp_name’
*ITERVLE	Number of iterations required to achieve flash convergence
*FMC1SURF	Surfactant mole fraction contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC2COIL	Critical Oil Sat contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC3CAPN	Capillary number contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC4GCAPN	Critical Generation Cap number contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC5OMF	Oil Mole Fraction contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC6SALT	Critical Salt Mole Fraction contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC7PERM	Permeability Dependence contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMCDRYOUT	Foam Dryout contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FLXSCTMAP	Flux sector number map

The following are available only with *GEOMECH:

*STRESI	Effective I-direction stress (X or R)
*STRESJ	Effective J-direction stress (Y or theta)
*STRESK	Effective K-direction stress (Z)
*STRESINVF	First invariant of effective stress tensor
*STRESINVS	Second invariant of stress effective tensor
*STRESSH	Shear stress (Y-Z or R-Z) for plane strain only
*STRESSHIJ	Shear stress on IJ plane
*STRESSHIK	Shear stress on IK plane
*STRESSHJK	Shear stress on JK plane
*STRESMXP	Maximum principal stress (+ for compressive, – for tensile)
*STRESMNP	Minimum principal stress (+ for compressive, – for tensile)
*STRESINT	Intermediate principle stress (+ for compressive, – for tensile)

*VMSTRESS	Von Mises stress
*STRNEPL	Effective Plastic strain
*STRESEFF	Mean effective stress (+ for compressive, – for tensile)
*STRESSM	Mean total stress (+ for compressive, – for tensile)
*TSTRESI	Total normal stress in I direction
*TSTRESJ	Total normal stress in J direction
*TSTRESK	Total normal stress in K direction
*STRESNORM	Effective stress normal to fracture
*STRESRAD	Radial total stress (available only for 3D radial grid)
*STRESTAN	Tangential total stress (available only for 3D radial grid)
*ESTRESRAD	Radial effective stress (available only for 3D radial grid)
*ESTRESTAN	Tangential effective stress (available only for 3D radial grid)
*SHSTRRAD	Radial shear stress (available only for 3-D radial grid)
*DISPLRAD	Radial displacement (available only for 3D radial grid)
*DISPLTAN	Tangential displacement (available only for 3D radial grid)
*SAFACTOR	Shear safety factor (see *GOUTSRF)
*TSAFACTOR	Tensile safety factor (See *GOUTSRF)
*REACFORCE	Reaction force vector (see *GOUTSRF)
*PRMXDIR	Vector of maximum principle effective stress (*GRID only)
*PRMNDIR	Vector of minimum principle effective stress (*GRID only)
*STRAINI	I-direction normal strain (X or R)
*STRAINJ	J-direction normal strain (Y or theta)
*STRAINK	K-direction normal strain (Z)
*STRAINSH	Shear strain
*STRAINSHIJ	Shear strain on IJ plane
*STRAINSHIK	Shear strain on IK plane
*STRAINSHJK	Shear strain in JK plane
*STRNMXP	Maximum principle strain
*STRNMNP	Minimum principle strain
*STRAINVOL	Volumetric strain
*STRESSDIFF	Stress differential = maximum principal stress – minimum principal stress
*VOIDRATIO	Void Ratio = porosity/(1-porosity)
*VPOROSGEO	Reservoir porosity calculated from geomechanics module
*VPOROSTGEO	True porosity calculated from geomechanics module
*PORDIFF	Difference between geomechanics and reservoir porosity (VPOROSGEO minus POROS)
*VERDSPLGEO	Vertical displacement “up” based on geomechanics
*SUBSIDGEO	Vertical displacement “down” (subsidence) based on geomechanics
*VDISPL	Vector of grid displacement (*GRID only) **For more on displacement outputs see EXPLANATION, below.**
*YLDSTATE	Stress state = 0 In Elastic state = 1 On shear failure envelope = 2 On the compressive cap = 3 At the corner (intercept between cap and shear failure envelope) = 4 On the tensile cutoff surface
*BIOT	Biot’s constant
*GCOHESION	Cohesion value

*HARDENING	Hardening parameter
*POISSON	Poisson's ratio
*POISSONEQ	Equivalent Poisson's ratio for jointed rock-mass
*YIELD	Yielding stress
*YOUNG	Young's elastic modulus
*YOUNGEQ	Equivalent Young's modulus for jointed rock mass
*JYOUNGMI	Young's modulus for jointed rock mass in I direction
*JYOUNGMJ	Young's modulus for jointed rock mass in J direction
*JYOUNGMK	Young's modulus for jointed rock mass in K direction
*THEXPCOEF	Linear thermal expansion coefficient of rock
*THCOMPR	Thermal compressibility of rock
*BULKVOL	Bulk volume
*GEORTYPE	Rock type number for geomechanical material

DEFAULTS:

Optional keyword. Defaults are *WELL *BRIEF, *RES *NONE, and *GRID *NONE and *FLUX_SECTOR *NONE.

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail in the print file may be changed during the simulation.

EXPLANATION:

For dual porosity and dual permeability runs, both the fracture and matrix grid variable will be printed, example:

```
*OUTPRN *GRID *SG
```

prints both the matrix and fracture gas saturations.

Note that

```
*OUTPRN *GRID *SO
*OUTPRN *GRID *SG
*OUTPRN *GRID *SW
```

can also be written as:

```
*OUTPRN *GRID *SO *SG *SW
```

When *OUTPRN *WELL *RESERVOIR or *OUTPRN *WELL *ALL is specified, well mobility-weighted grid block pressure, grid block datum pressure, and drawdown are printed to the output file. These mobility-weighted quantities are weighted averages of values for the individual completions, with the weighting factors proportional to the total mobilities of the phases in the completion grid blocks.

When *OUTPRN *GRID *SATP is specified, a saturation pressure calculation is done at each block's global composition. An attempt at a dew point calculation is performed, unless the block is oil filled, when a bubble point calculation is done. An initial search is carried out for a 2-phase region; if it is not found then the calculation will abort for that block and zero pressure will be reported. Otherwise, the saturation pressure will be reported.

When *OUTPRN *PSPLIT or *OUTPRN *PSPLITNOCOMP is specified, well, group and sector information are printed to the output file in production split form, in addition to the usual output. Production split output lists the total volumes and rates split up according to reservoir origin; that is, oil arising as a black oil (separator oil from the reservoir oil phase), oil arising as a condensate (separator oil from the reservoir gas phase), gas arising as solution gas (separator gas from the reservoir oil phase), or gas arising as free gas (separator gas from the reservoir gas phase). If *PSPLIT is used, then the compositions of the production split fluids are output as well. If an INL stream is present, then INL arising from the reservoir oil and gas phases is also reported.

*OUTPRN *GRID *SATP provides only an approximate value of saturation pressure, without doing rigorous calculation for each of the grid blocks as a time-saving measure. For precise value it is recommended to output global composition, pressure and temperature of the grid block and use that in WinProp to get the saturation pressure.

Flux-Sector Reports

The purpose of **flux sectors** defined by *FLUX_SECTORNAMES and *FLUX_ISECTOR is to report net fluid flux (a) into each reservoir region and (b) between adjacent regions. Each flux sector is identified in reports by its name as well as its number. For example, flux sector #1 is identified by “FLUXSEC-1” as well as its given name.

Sub-keyword *INTERSEC enables the reporting of net flux through the boundary between each pair of adjacent flux sectors. If there are n_{fs} flux sectors, then there can be up to $n_{fs} \cdot (n_{fs}-1)/2$ boundaries. No report is made for a pair of sectors that have no common boundary. The order of sectors in the boundary descriptor determines the sign of the flux. For example, for boundary “Flow from FLUXSEC-1 to FLUXSEC-2”, flow from flux sector #1 to #2 is positive while flow from #2 to #1 is negative. For computational purposes a flux boundary consists of a list of interblock connections, where the two blocks of each connection are members of the two flux sectors. Each reported quantity is summed over all the individual interblock connections that make up that boundary. Each flux boundary’s report will appear as one column in the text output.

Sub-keyword *INFLUX enables the reporting of net flux into each flux sector from all adjacent flux sectors. A descriptor for flux sector #1 would be “Flow to FLUXSEC-1”. The reported values are net, that is, flow into the sector is positive and flow out is negative. Each flux sector’s influx report will appear as one column in the text output.

Selection of Results File Format (Optional) *RESULTFILE

PURPOSE:

*RESULTFILE identifies in what format the graphical display files are to be written – the older SRF system, the RESULTS-displayable Simulation Results (IRS/MRF) system, or both.

The older SRF system is seldom used and this keyword is becoming obsolete.

FORMAT:

```
*RESULTFILE   (*SRF)
              (*SR2)
              (*BOTH)
```

DEFINITIONS:

*SRF

This subkeyword specifies that the graphical display file will be written in the SRF (Simulation Results File) format only. This is the format that was used exclusively by GEM in versions through 94.00. A single file with the extension .srf is produced.

*SR2

This subkeyword specifies that the graphical display files will be written in the RESULTS-displayable SR2 format. Two files will be produced with the extensions .irf and .mrf .

*BOTH

This subkeyword specifies that graphical display information should be written in BOTH the SRF and the SR2 formats. Three file are produced, with extensions .srf, .irf, and .mrf .

DEFAULTS:

Optional keyword. If *RESULTFILE does not appear in the data, the default is *RESULTFILE *SR2. If *RESULTFILE is encountered with no following keyword, the action is equivalent to *RESULTFILE *SR2 .

CONDITIONS:

If used, this keyword must appear in the INPUT/OUTPUT CONTROL keyword group.

EXPLANATION:

Versions of GEM through 94.00 wrote display information using the SRF file format exclusively.

Use keyword *SRFORMAT for a choice between IRF/MRF and SR3 formats.

Selection of Additional Plain Text SR2 Main File (Optional)

***SR2ASCII**

PURPOSE:

*SR2ASCII indicates that if graphical display files are being written in the Simulation Results format (under *RESULTFILE, or *SRFORMAT), an ASCII copy of the binary Main Results File (.mrf) or SR3 (.sr3) will also be produced, with the extension .asc.

FORMAT:

*SR2ASCII

DEFINITIONS:

*SR2ASCII

Indicates that the ASCII copy of the main results file will be written. If Simulation Results files are not written in the current run this keyword is ignored by the simulator.

DEFAULTS:

Optional keyword. If *SR2ASCII does not occur in the data set, the ASCII copy of the binary Main Results File is NOT written.

CONDITIONS:

If used, this keyword must appear in the INPUT/OUTPUT CONTROL keyword group.

EXPLANATION:

The ASCII Main Results File is primarily used for debugging, but may be useful to the user who wishes to write his own utility programs to extract certain values.

Floating-Point Precision for Writing Output to SR2 Main Files (Optional) *SR2PREC

PURPOSE:

*SR2PREC indicates input of the floating-point precision to which the output to graphical display files is to be written.

FORMAT:

*SR2PREC (*SINGLE | *DOUBLE)

DEFINITIONS

*SR2PREC

Flags the precision with which floating point data is written to the binary SR2 file. Use *SINGLE for 4 bytes and *DOUBLE for 8 bytes. *SINGLE is not available with *XDR *OFF. Use the default *SINGLE unless there is special need for extremely high accuracy in visualization.

DEFAULTS:

If *SR2PREC is absent, then *SR2PREC *SINGLE is assumed.

CONDITIONS:

*SR2PREC, if present, should appear in the I/O Control data section only.

EXPLANATION:

Before version 2011.10, GEM always wrote the Simulation Results data records in 8 byte. In version 2011.10 this value was changed to 4 bytes. Since GEM writes its restart records to a separate file (extension .rst), always in 8 bytes, changing the precision of Simulation Results records will not affect the GEM restart runs. The GEOPOSTPRO runs using default single precision Simulation Results records are also unlikely to be affected. The default settings will ensure that the size of main-results-file (.mrf) or sr3 file (.sr3) is substantially reduced especially for data sets writing large amount of information to Simulation Results file for visualization. This will result in improved loading of simulation results files to RESULTS software for visualization.

In rare situations, if required, it is possible to revert back to pre-2011.10 default of Simulation Results precision by specifying *SR2PREC *DOUBLE in the I/O Control section of the data file.

Simulation Results File Format (Optional) *SRFORMAT

PURPOSE:

Specify the file format for passing simulation results to CMG viewing software.

FORMAT:

*SRFORMAT *sr_format*

sr_format = *SR3 |*IMRF | *SR3_IMRF

DEFINITIONS:

*SR3

Write simulation results in the SR3 file format only. An IRF/MRF file set will not be created. A child restart run needs the SR3 file that was written by the parent run. The corresponding command-line argument is "-sr3".

*IMRF

Write simulation results in the IRF/MRF file format only. An SR3 file will not be created. A child restart run needs the IRF/MRF file set that was written by the parent run. The corresponding command-line argument is "-imrf".

*SR3_IMRF

Write simulation results in both the SR3 and IRF/MRF file formats. Both SR3 and IRF/MRF files will be created. A child restart run needs the SR3 file that was written by the parent run. The corresponding command-line argument is "-sr3_imrf".

DEFAULTS:

If *SRFORMAT is absent and all three command-line arguments are absent, then *SR3_IRMF is assumed.

CONDITIONS:

This keyword may appear only in the IO Control data section.

If both *SRFORMAT keyword data and the *SRFORMAT command-line arguments are used, the command-line is overridden by the keyword data in GEM.

EXPLANATION:

Simulation results are written to a file as the simulation run progresses. The contents of this file can be viewed graphically during and after the simulation run. There are two file formats available for storing the simulation results.

The IRF/MRF file set consists of a text index file (filename suffix ".irf") which indexes the contents of the corresponding binary file (filename suffix ".mrf"). These files are used by CMG products "Results Graph", "Results 3D" and "Report Writer" which have been available for some years.

The SR3 file is a single file with name suffix ".sr3" and is used by the CMG product "Results" introduced in 2016. This file stores the same information as the IRF/MRF files but is optimized for use by "Results".

File Names

By default, the SR3 output file has a base name that is the same as the textual output (.out) file. Use *FILENAME sub-keyword *SR3-OUT to override this default.

A restart run requires a name for the input restart file(s). Use *FILENAME sub-keyword(s) or command-line argument "-r" to specify the (base) name of input restart file(s). *FILENAME sub-keywords *INDEX-IN, *SR3-IN and *RESTARTIN can be used to specify the base name of the input restart file(s), for any of the *SRFORMAT options. For example, assume that a data set uses *FILENAME *INDEX-IN ′test.irf′ to specify the input restart file for the *SRFORMAT *IMRF file format. If that same data is run with the *SR3 file format option specified by command-line argument "-sr3", the input restart file's base name is isolated from *INDEX-IN and the ".sr3" suffix is added to get ′test.sr3′. Alternatively, the data could be changed to *FILENAME *SR3-IN ′test.sr3′.

Note that it is *SRFORMAT, not *FILENAME, that determines the expected format of the input restart file. *FILENAME merely provides the (base) name of a file when it is used.

As before GEM restart file continues to be written in a separate binary format with file extension .rst.

Interaction with Other Options

The *REWIND option in GEM is unaffected by *SRFORMAT.

Geo post-processing keyword *GEOPOSTINPUT *SR2 will use the *SR3 file for obtaining pressure/temperature if the prior run was made with *SRFORMAT *SR3 or *SRFORMAT *SR3_IMRF.

Examples:

The following data enables SR3-only usage, if no command-line argument is used.

```
*SRFORMAT *SR3
```

The following command-line argument will force the usage of SR3 format only, overriding any *SRFORMAT keyword data.

```
gm201600.exe –f input_data –sr3
```

Simulation Results Writing Frequency (Optional) *WSRF

PURPOSE:

*WSRF controls how frequently well and/or grid information is written to the output Simulation Results File.

FORMAT:

```
*WSRF    (*WELL)          freq
                          (*TIME)
                          (*TNEXT)
         (*GRID)          freq
                          (*TIME)
                          (*TNEXT)
         (*GRIDDEFORM)    freq
                          (*TIME)
                          (*TNEXT)
```

DEFINITIONS:

*WELL freq

Write well results to the Simulation Results File every freq timesteps, where freq is an integer.

*WELL *TIME

Write well results to the Simulation Results File at every time specified by the subsequent recurrent *TIME or *DATE keywords in the input file.

*WELL *TNEXT

Write well results to the Simulation Results File at the next *TIME or *DATE keywords, and then suspend writing until another *WSRF keyword is encountered (in the WELL AND RECURRENT DATA section).

*GRID freq

Write grid results to the Simulation Results File every freq timesteps, where freq is an integer.

*GRID *TIME

Write grid results to the Simulation Results File of Simulation Results files at every time specified by subsequent recurrent *TIME or *DATE keywords in the input file.

*GRID *TNEXT

Write grid results to the Simulation Results File at the next *TIME or *DATE keywords, and then suspend writing until another *WSRF keyword is encountered (in the WELL AND RECURRENT DATA section).

*GRIDDEFORM

Pertains to writing of information for grid deformation due to geomechanics effects. Dumping *GRIDDEFORM information more frequently than the default can increase the size of the Simulation Results files significantly.

Freq, *TIME and *TNEXT apply to *GRIDDEFORM in the same way as they do for *GRID and *WELL above.

DEFAULTS:

Optional keyword. If *WSRF is not present in the input data file, the defaults are: Well data is written at every timestep, while grid results are written to the Simulation Results File at all well change times. These defaults are equivalent to the data lines:

```
*WSRF    *WELL    1
*WSRF    *GRID    *TIME
```

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail and frequency of records in the Simulation Results files may be changed during the simulation.

EXPLANATIONS:

If the following are specified:

```
*WSRF *WELL 0
*WSRF *GRID 0
```

no Simulation Results files will be opened and *WSRF cannot be redefined later in the recurrent data section. If no results information is required initially, but is required later during the simulation run, enter freq as a large integer, e.g. 100000, in the I/O Control section, and override it with the appropriate frequency in the well and recurrent data section.

Visualizing Geomechanics Grid Deformation

Keyword *GRIDDEFORM allows the user to view grids in Results that deform with time as calculated by the geomechanics module. This feature is available in Results only for corner-point grid type. GEM writes to the Simulation Results file grid definition data that tells Results the type, structure and appearance of the grid. If the *GRIDDEFORM option is requested, GEM tells Results that the grid is corner-point type and does any necessary conversion. If the user specified *GRID *CART in data then the conversion is exact, that is, Results will draw an initial grid that looks exactly like the user's Cartesian grid. If the user specified *GRID *RADIAL in data then the conversion at initial conditions is exact in the I-K (R-Z) plane only. Several issues arise from this technique.

1. Initial conditions are plotted with the original grid whereas the "converted" grid is used to display all subsequent times. For the radial grid case you may notice a change in the grid when going from initial time to subsequent times.

2. For a “converted” 3D radial grid, circular arcs are replaced with straight lines between block corners; for example, for ntheta = 3 the areal (I-J) plane appears as a triangle instead of a circle. The angular direction of a 2D radial grid cannot be displayed at all.
3. A “converted” 3D radial grid does not include the innermost radial block, so both fluid-flow and geomechanics quantities cannot be viewed for that block. However, the fluid-flow equations are still solved for that block.
4. Without *GRIDDEFORM the grid is displayed as “radial” and the innermost radial block shows a value that is (1) correct for fluid-flow quantities and (2) an average of surrounding block values for geomechanics quantities.

The writing of grid deformation information is controlled also by the frequency of geomechanics updating specified by keyword *GCUPDATE in the Geomechanics section.

Items in Simulation Results File (Optional) *OUTSRF

PURPOSE:

*OUTSRF identifies what information is written to the Simulation Results file.

FORMAT:

```
*XDR        (*ON | *OFF)
*OUTSRF     (*WELL) or        (well_list)
            (*SPECIAL)        *PSPLIT
                              *PSPLITNOCOMP
                              *LAYER          (*ALL)
                                              (*DOWNHOLE)
                              *WELLPI         (*NONE)
                              *WSTRMOLDEN
                              *WSTRMASDEN
                              *SIMPERF
                              *WELL-STATUS
            (*GRID)           *NONE
                              (grid_list)
            (*RES)            (*ALL)
                              (*NONE)
            (*FLUX_SECTOR)    (*ALL)          (mass-mole selector)
                              (*NONE)         (sc-rc selector)
                                              (ijk-con selector)
```

DEFINITIONS:

*XDR

The binary (data) file may be written in External Data Representation (XDR) format as well as the binary format native to your platform. Use of XDR allows the SR2 binary file(s) to be written on one platform and read on another. For example, the SR2 files can be generated on a UNIX workstation server and then accessed with RESULTS or the Report Writer on a PC. If the Simulation Results (IRF/MRF) is in XDR format, then the keyword “XDR” will appear near the top of the index file (IRF).

*WELL or *SPECIAL

This subkeyword specifies that well or special variables will be written to the Simulation Results file(s). The following well variables are always generated for each well:

well bottom-hole pressure

well tubing-head pressure

surface oil rate

surface gas rate

surface water rate

surface intermediate liquid rate

cumulative oil production/injection

cumulative gas production/injection

cumulative water production/injection

cumulative intermediate liquid production/injection

well mobility-weighted datum pressure

well block pressure (pressure in grid block of well's reference completion)

The mobility-weighted datum pressure is a weighted average of the datum pressures in the grid blocks in which the well is completed, with the weighting factors proportional to the total phase mobility in the completion grid block.

Production split information is also controlled by this keyword.

*PSPLIT / PSPLITNOCOMP

Production split information is output, with optional compositions. These keywords allow plotting of surface as well as reservoir volumes including those for in place fluid, IPF, of different production splits like Black-Oil, Condensate, Free-Gas etc., via RESULTS.

*WELLPI

Specification of OUTSRF WELL WELLPI will turn on calculation of well productivity / injectivity index at surface conditions for oil, gas and water streams for all wells. For a well with a single layer the well pi for a given surface stream is the ratio of surface rate divided by the pressure draw down. For a multi-layer completion a well pi calculation is no longer straight forward since the draw down will be different for each layer. The values reported with OUTSRF WELL WELLPI are based on the assumption of unit draw down for each layer. Therefore for a multi-layer well completion the value reported is only an approximation since actual drawdown can vary significantly between layers or completions. The values are written to the RESULTS output files. These well productivity/injectivities will appear in the list of well quantities for each surface stream that are available for plotting by Results Graph.

*WSTRMOLDEN

Specification of *OUTSRF *WELL *WSTRMOLDEN will turn on calculation of well stream molar density at surface conditions for oil, gas, water, intermediate liquid and wet-gas streams for all wells. The values are written to the RESULTS output files. These well stream molar densities will appear in the list of well quantities for each surface stream that are available for plotting by Results Graph.

*WSTRMASDEN

Specification of *OUTSRF *WELL *WSTRMASDEN will turn on calculation of well stream mass density at surface conditions for oil, gas, water, intermediate liquid and wet-gas streams for all wells. The values are written to the RESULTS output files. These well stream mass densities will

appear in the list of well quantities for each surface stream that are available for plotting by Results Graph.

*SIMPERF

Specification of *OUTSRF *SPECIAL *SIMPERF will activate output of simulation performance indicators, i.e., *ELTSCUM (elapsed time in seconds), *ELTRATE(Elapsed time in seconds per simulation day), *TSTEPCUM (Cumulative number of timesteps), *NCYCCUM (Cumulative Newton cycles), *TSCUTCUM (Cumulative timestep cuts), *DELT (Timestep size, days), *NCYCPTS (Newton cycles per timestep), *SOLITPN (Solver iterations per Newton cycle), *MBERROR (Material balance error, %), *AVGIMPL (Average implicitness, %) and *MEMUSAGE (Memory usage, MB) for the run. These indicators can also be individually output simply by specifying, *OUTSRF *SPECIAL xxx, where xxx is one of the performance indicators.

*WELL-STATUS

This subkeyword specifies that status of wells (on production, on water injection, on gas injection, or shutin) during a timestep will be written for all the wells in the model to the Simulator Results file system. For status of wells in individual groups please refer to output from origin group in the RESULTS graph.

*LAYER

This subkeyword specifies that well performance values will be written for all layers (completions) of wells specified to the Simulator Results file system. The default is *NONE (see DEFAULTS below). Using *ALL specifies that layer information will be written for all wells to the Simulator Results file system. Rates/volumes will be written at surface conditions only.

*DOWNHOLE

This is a subkeyword of *LAYER *ALL and specifies that well performance values will be written to the SR2 file system at reservoir conditions in addition to surface conditions.

Use of *LAYER *option, with* *ALL *DOWNHOLE, *can increase the size of the Simulation Results file substantially especially for models with a large number of wells.*

*GRID

This subkeyword specifies that grid results will be written to the Simulation Results File or the Simulation Results files.

*RES

This keyword alone or followed by *ALL specifies that certain initial reservoir quantities such as permeability, reference porosity, and grid block relative permeability endpoints, will be written as a time 0 record to the Simulator Results files for display by the RESULTS graphical display system.

*FLUX_SECTOR

This subkeyword specifies that flow values of flux sectors will be written to the Simulator Results files. The output frequency is controlled by *WSRF *WELL. The output can be found under the LEASE origin in RESULTS GRAPH.

*NONE

No variables will be written to the SRF or SR2 files.

*ALL

Specifies that all of the fixed list of initial *RES properties will be written to the SR2 files. For *FLUX-SECTOR option, it means both influx and inter-sector flux will be written out.

*OBHLOSSCUM

Net cumulative energy lost (-) or gained (+) by the overburden heat loss model.

*OBHLOSSRAT

Rate of net energy lost (-) or gained (+) by the overburden heat loss model.

Well_list

A series of keywords to identify which well or special variables to write to the SRF or Simulator Results files. These values can then be displayed as time series. Despite their appearance under well_list, many of these variables refer to particular grid blocks rather than to wells.

Valid keywords in the well_list for *WELL are as follows, where block_id refers to a grid cell identifier in UBA notation, ‘well_name’ refers to a valid well name, and ‘comp_name’ refers to a valid component name. (If a ‘well_name’ is followed by a UBA block identifier (block_id), then that block must be in the well’s completion.)

*DELT	timestep size
*PAVG	average reservoir pressure
*TAVG	average reservoir temperature
*ELTSCUM	Elapsed time in seconds
*ELTSRATE	Elapsed time in seconds per simulation day
*TSTEPCUM	Cumulative number of timesteps
*NCYCCUM	Cumulative Newton cycles
*SOLITCUM	Cumulative solver iterations
*TSCUTCUM	Cumulative timestep cuts
*NCYCPTS	Newton cycles per timestep
*SOLITPN	Solver iterations per Newton cycle
*MBERROR	Material balance error, %
*AVGIMPL	Average implicitness, %
*MEMUSAGE	Memory usage (MB)
*PRES block_id	pressure at block_id
*SO block_id	oil saturation at block_id
*SG block_id	gas saturation at block_id
*SW block_id	water saturation at block_id
*TEMP block_id	temperature at block_id

*RHOO block_id	oil molar density at block_id
*RHOG block_id	gas molar density at block_id
*RHOW block_id	Aqueous molar density at block_id
*DENO block_id	oil mass density at block_id
*DENG block_id	gas mass density at block_id
*DENW block_id	water mass density at block _id
*KRO block_id	oil relative permeability at block_id
*KRG block_id	gas relative permeability at block_id
*KRW block_id	water relative permeability at block_id
*KRINTER block_id	relative permeability interpolant value
*PCG block_id	capillary pressure for gas at block_id
*PCW block_id	capillary pressure for water at block_id
*SIG block_id	interfacial tension between oil and gas at block_id
*SIGMAGW block_id	Interfacial tension between gas and water at block_id (used in Foam models)
*SIGMAOW block_id	Interfacial tension between oil and water at block_id (used in non-Foam models)
*PSIG block_id	Pseudo interfacial tension between oil and gas at block_id
*TSO block_id	trapped oil saturation at block_id
*FRG block_id	gas phase mole fraction at block_id
*VISO block_id	oil viscosity at block_id
*VISG block_id	gas viscosity at block_id
*VISW block_id	water viscosity at block_id
*MWO block_id	oil molecular weight at block_id
*MWG block_id	gas molecular weight at block_id
*BHP ‘well_name’ block_id	wellbore pressure for well ‘well_name’ at layer block_id
*TROW ‘well_name’ block_id	oil transmissibility for well ‘well_name’ at layer block_id
*TRGW ‘well_name’ block_id	gas transmissibility for well ‘well_name’ at layer block_id
*TRWW ‘well_name’ block_id	water transmissibility for well ‘well_name’ at layer block_id
*WSPE ‘well_name’	well specifications for well ‘well_name’
*WKRO ‘well_name’ block_id	oil relative permeability for well ‘well_name’ at layer block_id
*WKRG ‘well_name’ block_id	gas relative permeability for well ‘well_name’ at layer block_id
*WKRW ‘well_name’ block_id	water relative permeability for well ‘well_name’ at layer block_id
*ADS ‘comp_name’ block_id	adsorbed gmol/grid-block volume of component ‘comp_name’ in block_id
*K ‘comp_name’ block_id	gas/oil k-value of component ‘comp_name’ at block_id
*X ‘comp_name’ block_id	oil composition of component ‘comp_name’ at block_id
*Y ‘comp_name’ block_id	gas composition of component ‘comp_name’ at block_id
*Z ‘comp_name’ block_id	global composition of component ‘comp_name’ at block_id
*W ‘comp_name’ block_id	aqueous composition of component ‘comp_name’ at block_id
*RMOL ‘comp_name’ block_id	mole of ‘comp_name’ per block volume at block_id
*DROP block_id	pressure drop from time zero at block_id

*ZWEL ‘comp_name’ ‘well_name’	Mole fraction of ‘comp_name’ in total well stream of ‘well_name’
*XWEL ‘comp_name’ ‘well_name’	Mole fraction of ‘comp_name’ in surface oil from ‘well_name’
*YWEL ‘comp_name’ ‘well_name’	Mole fraction of ‘comp_name’ in surface gas from ‘well_name’
*IWEL ‘comp_name’ ‘well_name’	Mole fraction of ‘comp_name’ in surface intermediate liquid from ‘well_name’
*FOIP	oil in place in the fracture
*FIIP	intermediate liquid in place in the fracture
*FGIP	gas in place in the fracture
*FWIP	water in place in the fracture
*HCPVIN	% hydrocarbon pore volume injected (non-aqueous components)
*MOIP	oil in place in the matrix
*MIIP	intermediate liquid in place in the matrix
*MGIP	gas in place in the matrix
*MWIP	water in place in the matrix
*TOIP	total oil in place in the reservoir
*TIIP	total intermediate liquid in place in the reservoir
*TGIP	total gas in place in the reservoir
*TWIP	total water in place in the reservoir
*RECO	total oil recovery
*RECI	total intermediate liquid recovery
*RECG	total gas recovery
*WINFLUX	total cumulative water influx from aquifer
*CAPN block_id	Capillary number gas-oil at block_id (output only when *VELDEPRP is used)
*CAPNGW block_id	Capillary number gas-water at block_id, ; for foam models with *IFTTABLE
*CAPNOW block_id	Capillary number, oil-water at block_id; for non-foam models with *IFTTABLE
*PERMI block_id	I-direction absolute permeability at block_id
*PERMJ block_id	J-direction absolute permeability at block_id
*PERMK block_id	K-direction absolute permeability at block_id
*POROS block_id	Current porosity
*SGHYS block_id	Hysteresis – Gas saturation corresponding to reversal point from drainage to imbibition; starting point of drainage to imbibition curve at block_id
*SGRHYS block_id	Hysteresis – Maximum trapped gas saturation, end point of drainage to imbibition scanning curve; gas saturation where gas relative permeability is zero, at block_id
*SGDTHY block_id	Hysteresis – Dynamic trapped gas saturation, at block_id
*SALIN block_id	Aqueous phase salinity (molality) at block_id
*OILH block_id	Oil phase enthalpy at block_id
*GASH block_id	Gas phase enthalpy at block_id
*WATH block_id	Water enthalpy at block_id

*FLUIDH block_id	Fluid enthalpy at block_id (where fluid enthalpy is (phase saturation times phase molar density) weighted average of enthalpies of all phases present in the block)
*WWEL ‘comp_name’ ‘well_name’	Mole fraction of ‘comp_name’ in aqueous stream of ‘well_name’ at surface conditions
*WWELR ‘comp_name’ ‘well_name’	Mole fraction of ‘comp_name’ in aqueous stream of ‘well_name’ at reservoir conditions
*DENWW ‘well_name’	Mass density of aqueous stream for well ‘well_name’ at surface conditions
*DENWWR ‘well_name’	Mass density of aqueous stream for well ‘well_name’ at reservoir conditions
*MOLALITYW ‘comp_name’ ‘well_name’	Molality of ‘comp_name’ in aqueous stream of ‘well_name’ at surface conditions
*MOLALITYWR ‘comp_name’ ‘well_name’	Molality of ‘comp_name’ in aqueous stream of ‘well_name’ at reservoir conditions
*SALINWW ‘well_name’	Salinity of aqueous stream for well ‘well_name’ at surface conditions
*SALINWWR ‘well_name’	Salinity of aqueous stream for well ‘well_name’ at reservoir conditions
*VISWW ‘well_name’	Viscosity of aqueous stream for well ‘well_name’ at surface conditions
*VISWWR ‘well_name’	Viscosity of aqueous stream for well ‘well_name’ at reservoir conditions
*ITERVLE block_id	Number of iterations required to achieve flash convergence in block_id
*NBLKFFAIL	Number of blocks with flash convergence failure
*FMC1SURF block_id	Surfactant mole fraction contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)
*FMC2COIL block_id	Critical Oil Sat contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)
*FMC3CAPN block_id	Capillary number contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)
*FMC4GCAPN block_id	Crit. Gen. Cap number contributor to FM in block-id; (-1 not used; 0 destroyer; else contributor)
*FMC5OMF block_id	Oil Mole Fraction contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)
*FMC6SALT block_id	Critical Salt Mole Fraction contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)
*FMC7PERM block_id	Permeability dependence contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)
*FMCDRYOUT block_id	Foam Dryout contributor to FM in block_id; (-1 not used; 0 destroyer; else contributor)

<u>Well Variables for GHG Models</u>

*GHGSOL	Total CO_2 dissolved in the aqueous phase
*GHGAQU	Total CO_2 in the form of aqueous ions
*GHGMNR	Total CO_2 in the form of precipitated mineral
*GHGGAS	Total CO_2 in the gaseous state
*GHGLIQ	Total CO_2 in the liquid state
*GHGSCRIT	Total CO_2 in the super-critical fluid state

*GHGTHY	Total CO_2 in the trapped gas due to Hysteresis
*MOLALITY ‘comp_name’ ‘block_id’	Aqueous phase molality of gaseous or aqueous species
*MINERAL ‘comp_name’ ‘block_id’	Change in moles of mineral component
*DPORMNR ‘comp_name’ ‘block_id’	Porosity change due to mineral reactions
*RFO ‘block_id’	Resistance factor due to mineral reactions
*PH ‘block_id’	pH
*EQVFRIEX ‘comp_name’ ‘block_id’	Equivalent fraction for component ‘comp_name’ in block ‘block_id’
*EQVFRIEXN ‘comp_name; ‘block_id’	Equivalent fraction for component ‘comp_name’ in block ‘block_id’ determined using maximum CEC over all the grid-blocks.
*ACTIV ‘comp_name’ ‘block_id’	Activity of aqueous component ‘comp_name’ in block ‘block_id’
*ACTIVCOEF ‘comp_name’ ‘block_id’	Activity coefficient of aqueous component ‘comp_name’ in block ‘block_id’
*LSATINDM ‘comp_name’ ‘block_id’	Log10 (saturation index) of mineral component ‘comp_name’ in block ‘block_id’

The presence of any of the above variables *GHGSOL through *GHGTHY for GHG models also activates writing of CO_2 inventory in the reservoir, in the output file. The ‘block_id’ refers to the block address at which the associated variables above will be output.

The following are available only with *GEOMECH:

STRESI block_id	Effective I-direction stress (X or R)
STRESJ block_id	Effective J-direction stress (Y or theta)
STRESK block_id	Effective K-direction stress (Z)
STRESSINVF block_id	First invariant of stress tensor
STRESSINVS block_id	Second invariant of stress tensor
STRESSH block_id	Shear stress (Y-Z or R-Z) for plane strain only
STRESSHIJ block_id	Shear stress on IJ plane
STRESSHIK block_id	Shear stress on IK plane
STRESSHJK block_id	Shear stress on JK plane
STRESMXP block_id	Maximum principal stress (+ for compressive, – for tensile)
STRESMNP block_id	Minimum principal stress (+ for compressive, – for tensile)
STRESSDIFF block_id	Stress differential at block_id = maximum principal stress – minimum principal stress
STRESINT block_id	Intermediate principle stress (+ for compressive, – for tensile)
STRESRAD block_id	Radial total stress (available only for 3D radial grid)
STRESTAN block_id	Tangential total stress (available only for 3D radial grid)
ESTRESRAD block_id	Radial effective stress (available only for 3D radial grid)
ESTRESTAN block_id	Tangential effective stress (available only for 3D radial grid)
*SHSTRRAD block_id	Radial shear stress (available only for 3-D radial grid)
DISPLRAD block_id	Radial displacement (available only for 3D radial grid)
DISPLTAN block_id	Tangential displacement (available only for 3D radial grid)
VMSTRESS block_id	Von Mises stress
VOIDRATIO block_id	Void Ratio = porosity/(1- porosity)

STRNEPL block_id	Effective Plastic strain
STRESEFF block_id	Mean effective stress (+ for compressive, – for tensile)
STRESSM block_id	Mean total stress (+ for compressive, – for tensile)
TSTRESI block_id	Total normal stress in I direction
TSTRESJ block_id	Total normal stress in J direction
TSTRESK block_id	Total normal stress in K direction
STRESNORM block_id	Effective stress normal to fracture
SAFACTOR block_id	Shear safety factor for block_id (see *GOUTSRF)
TSAFACTOR block_id	Tensile safety factor for block_id (see *GOUTSRF)
PRMXDIR block_id	Vector of maximum principle effective stress (*GRID only)
PRMNDIR block_id	Vector of minimum principle effective stress (*GRID only)
STRAINI block_id	I-direction normal strain (X or R)
STRAINJ block_id	J-direction normal strain (Y or theta)
STRAINK block_id	K-direction normal strain (Z)
STRAINSH block_id	Shear strain
STRAINSHIJ block_id	Shear strain on IJ plane
STRAINSHIK block_id	Shear strain on IK plane
STRAINSHJK block_id	Shear strain in JK plane
STRNMXP block_id	Maximum principle strain
STRNMNP block_id	Minimum principle strain
STRESSDIFF block_id	Maximum principal stress – minimum principal stress
STRAINVOL block_id	Volumetric strain
VPOROSGEO block_id	Reservoir porosity calculated from geomechanics module
VPOROSTGEO block_id	True porosity calculated from geomechanics module
PORDIFF block_id	Difference between geomechanics and reservoir porosity (VPOROSGEO minus POROS)
VERDSPLGEO block_id	Vertical displacement “up” based on geomechanics
SUBSIDGEO block_id	Vertical displacement “down” (subsidence) based on geomechanics
VDISPL block_id	Vector of grid displacement (*GRID only)

For more on displacement outputs see EXPLANATION, below.

YLDSTATE block_id	Stress state = 0 In Elastic state = 1 On shear failure envelope = 2 On the compressive cap = 3 At the corner (intercept between cap and shear failure envelope) = 4 On the tensile cutoff surface
BIOT block_id	Biot’s constant
GCOHESION block_id	Cohesion value
HARDENING block_id	Hardening parameter
*POISSON block_id	Poisson’s ratio
*POISSONEQ block_id	Equivalent Poisson’s ratio for jointed rock-mass
*YIELD block_id	Yielding stress
*YOUNG block_id	Young’s elastic modulus
*YOUNGEQ block_id	Equivalent Young’s modulus for jointed rock mass

*JYOUNGMI block_id	Young's modulus for jointed rock mass in I direction
*JYOUNGMJ block_id	Young's modulus for jointed rock mass in J direction
*JYOUNGMK block_id	Young's modulus for jointed rock mass in K direction
*THEXPCOEF block_id	Linear thermal expansion coefficient of rock
*THCOMPR block_id	Thermal compressibility of rock
*BULKVOL block_id	Bulk volume
*GEORTYPE block_id	Rock type number for geomechanical material

grid_list

A series of keywords to identify which grid variables to write to the Simulator Results files. One value is written for each grid block in the reservoir.

The grid_list for *GRID is given by the following, 'comp_name' refers to a valid component name (see *COMPNAME):

*PRES	reservoir pressure
*DATUMPRES	block pressure corrected to datum depth
*DROP	pressure drop from time zero
*SO	oil saturation
*SG	gas saturation
*SW	water saturation
*TSO	trapped oil saturation
*RHOO	oil molar density
*RHOG	gas molar density
*RHOW	aqueous molar density
*FRG	gas phase mole fraction in o/g system
*PCW	capillary pressure for water
*PCG	capillary pressure for gas
*KRINTER	relative permeability interpolant value
*KRO	oil relative permeability
*KRG	gas relative permeability
*KRW	water relative permeability
*VISO	oil viscosity
*VISG	gas viscosity
*VISW	water viscosity
*MWO	oil molecular weight
*MWG	gas molecular weight
*DENO	oil mass density
*DENG	gas mass density
*DENW	water mass density
*SATP	saturation pressure (an approximate estimate, see explanation)
*SIG	interfacial tension between oil and gas
*SIGMAGW	Interfacial tension between gas and water (used in Foam models)
*SIGMAOW	Interfacial tension between oil and water (used in non-Foam models)
*PSIG	Pseudo interfacial tension between oil and gas
*TIGP	Total Gas in Place, SC
*SOLID	solid precipitated/flowing/deposited mass; resistance factor
*ADS 'comp_name'	adsorbed gmol/grid-block volume of 'comp_name'

*VELOCRC	phase velocities at reservoir conditions
*Z ‘comp_name’	global mole fraction of ‘comp_name’
*X ‘comp_name’	mole fraction of ‘comp_name’ in oil phase
*Y ‘comp_name’	mole fraction of ‘comp_name’ in gas phase
*W ‘comp_name’	mole fraction of ‘comp_name’ in aqueous phase
*K ‘comp_name’	gas/oil k-value of ‘comp_name’
*SALIN	Aqueous phase salinity (molality)
*ZALL	Global mole fraction of all hydrocarbon components
*XALL	Mole fraction of all components in oil phase
*YALL	Mole fraction of all components in gas phase
*WALL	Mole fraction of all components in the aqueous phase
*CAPN	capillary number, gas-oil (output only when *VELDEPRP is used)
*CAPNGW	capillary number, gas-water, for foam models with *IFTTABLE
*CAPNOW	Capillary number, oil-water, for non-foam models with *IFTTABLE
*STRMLN	block-face based phase velocities used in Results to construct streamlines
*POROS	Current porosity
*PERM	Current permeabilities
*PERMEFF	Current effective permeabilities
*PERMINT	Current intrinsic permeabilities
*TEMP	block temperature
*SGHYS	Hysteresis - Gas saturation corresponding to reversal point from drainage to imbibition; starting point of drainage to imbibition scanning curve
*SGRHYS	Hysteresis – Maximum trapped gas saturation, end point of drainage to imbibition scanning curve; gas saturation where gas relative permeability is zero
*SGDTHY	Hysteresis – Dynamic trapped gas saturation
*RFO	Oil permeability resistance factor
*RFG	Gas permeability resistance factor
*RFW	Water permeability resistance factor
*ZFACO	Z compressibility factor of oil phase
*ZFACG	Z compressibility factor of gas phase
*COMPRO	Compressibility of oil phase
*COMPRW	Compressibility of water phase
*COMPRG	Compressibility of gas phase
*COMPRT	Total compressibility of all fluid phases
*OILH	Oil phase enthalpy
*GASH	Gas phase enthalpy
*WATH	Water phase enthalpy
*FLUIDH	Fluid enthalpy(where fluid enthalpy is (phase saturation times phase molar density) weighted average of enthalpies of all the phases present in a block)
*IMPL	Adaptive Implicit flag value
*OBHLOSS	Over/underburden heat loss rate (see *HEAT-LOSS)
*IPSTCA	Parasol class map
*ITERVLE	Number of iterations required to achieve flash convergence

*FMC1SURF	Surfactant mole fraction contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC2COIL	Critical Oil Sat contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC3CAPN	Capillary number contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC4GCAPN	Critical Generation Cap number contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC5OMF	Oil Mole Fraction contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC6SALT	Critical Salt Mole Fraction contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMC7PERM	Permeability Dependence contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FMCDRYOUT	Foam Dryout contributor to FM; (-1 not used; 0 destroyer; else contributor)
*FLXSCTMAP	Flux sector number map
*FLUXCON	Output connection based phase fluxes to the Simulation Results file. This information can be accessed by ResultsReport using the *CONNECTION-FOR keyword in the 'rwd' file. Oil flux is obtained using *CONNECTION-FOR 'FLUXCONO' "*output-time*"(in days). Similarly water and gas flux are obtained using the 'FLUXCONW' and 'FLUXCONG' strings respectively in ResultsReport. Due to the large number of calculations required to compute face velocities associated with each connection, it is recommended that the *WSRF *TNEXT option be used to explicitly select (and thereby reduce) the amount of information when *FLUXCON is turned on. Otherwise run time may substantially increase and the Simulation Results file may become excessively large.

<u>Grid Variables for GHG Models</u>

*MOLALITY 'comp_name'	Aqueous phase molality of gaseous or aqueous species
*MINERAL 'comp_name'	Change in moles of mineral components
*DPORMNR	Porosity change due to mineral reactions
*RFO	Permeability resistance factor due to mineral reactions
*PH	pH
*EQVFRIEX 'comp_name'	Equivalent Fraction for 'comp_name'
*EQVFRIEXN 'comp_name'	Equivalent Fraction for 'comp_name' determined using maximum CEC over all the grid-blocks.
*ACTIV 'comp_name'	Activity of aqueous component 'comp_name'
*ACTIVCOEF 'comp_name'	Activity coefficient of aqueous component 'comp_name'
*LSATINDM 'comp_name'	Log10 (saturation index) of mineral component 'comp_name'

The following are available only with *GEOMECH:

STRESI	Effective I-direction stress (X or R)
STRESJ	Effective J-direction stress (Y or theta)
STRESK	Effective K-direction stress (Z)
STRESINVF	First invariant of stress tensor

STRESSINVS	Second invariant of stress tensor
STRESSH	Shear stress (Y-Z or R-Z) for plane strain only
STRESSHIJ	Shear stress on IJ plane
STRESSHIK	Shear stress on IK plane
STRESSHJK	Shear stress on JK plane
STRESMXP	Maximum principal stress (+ for compressive, – for tensile)
STRESMNP	Minimum principal stress (+ for compressive, – for tensile)
STRESSDIFF	Stress differential = maximum principal stress – minimum principal stress
STRESINT	Intermediate principle stress (+ for compressive, – for tensile)
STRESRAD	Radial total stress (available only for 3D radial grid)
STRESTAN	Tangential total stress (available only for 3D radial grid)
ESTRESRAD	Radial effective stress (available only for 3D radial grid)
ESTRESTAN	Tangential effective stress (available only for 3D radial grid)
*SHSTRRAD	Radial shear stress (available only for 3-D radial grid)
DISPLRAD	Radial displacement (available only for 3D radial grid)
DISPLTAN	Tangential displacement (available only for 3D radial grid)
*SAFACTOR	Shear safety factor (see *GOUTSRF)
*TSAFACTOR	Tensile safety factor (See *GOUTSRF)
REACFORCE	Reaction force vector (see *GOUTSRF)
TSTRESJ	Total normal stress in J direction
TSTRESK	Total normal stress in K direction
STRESNORM	Effective stress normal to fracture
PRMXDIR	Vector of maximum principle effective stress (*GRID only)
PRMNDIR	Vector of minimum principle effective stress (*GRID only)
STRAINI	I-direction normal strain (X or R)
STRAINJ	J-direction normal strain (Y or theta)
STRAINK	K-direction normal strain (Z)
STRAINSH	Shear strain
STRAINSHIJ	Shear strain on IJ plane
STRAINSHIK	Shear strain on IK plane
STRAINSHJK	Shear strain in JK plane
STRNMXP	Maximum principle strain
STRNMNP	Minimum principle strain
STRAINVOL	Volumetric strain
VPOROSGEO	Reservoir porosity calculated from geomechanics module
VPOROSTGEO	True porosity calculated from geomechanics module
VOIDRATIO	Void Ratio = porosity/(1-porosity)
PORDIFF	Difference between geomechanics and reservoir porosity (VPOROSGEO minus VPOROS)
VERDSPLGEO	Vertical displacement “up” based on geomechanics
SUBSIDGEO	Vertical displacement “down” (subsidence) based on geomechanics
VDISPL	Vector of grid displacement (*GRID only)

For more on displacement outputs see EXPLANATION, below.

YLDSTATE	Stress state = 0 In Elastic state = 1 On shear failure envelope = 2 On the compressive cap = 3 At the corner (intercept between cap and shear failure envelope)

	= 4 On the tensile cutoff surface
*BIOT	Biot's constant
*GCOHESION	Cohesion value
*HARDENING	Hardening parameter
*POISSON	Poisson's ratio
*POISSONEQ	Equivalent Poisson's ratio for jointed rock-mass
*YIELD	Yielding stress
* YOUNG	Young's elastic modulus
*YOUNGEQ	Equivalent Young's modulus for jointed rock mass
*JYOUNGMI	Young's modulus for jointed rock mass in I direction
*JYOUNGMJ	Young's modulus for jointed rock mass in J direction
*JYOUNGMK	Young's modulus for jointed rock mass in K direction
*THEXPCOEF	Linear thermal expansion coefficient of rock
*THCOMPR	Thermal compressibility of rock
*BULKVOL	Bulk volume
*GEORTYPE	Rock type number for geomechanical material
mass-mole selector	
*MASS	Flux sector summary contains mass fluxes of individual components
*MOLE	Flux sector summary contains mass fluxes of individual components
If neither present, component flows are not reported	
sc-rc selector	
*SC	Flux sector summary contains phase flux values in surface condition
*RC	Flux sector summary contains phase flux values in reservoir condition
If neither present, *RC is selected	
ijk-con selector	
*SUM	Flux sector summary contains summation of connections in all three direction
*ICON	Flux sector summary contains flow across connections in I direction
*JCON	Flux sector summary contains flow across connections in J direction
*KCON	Flux sector summary contains flow across connections in K direction

If none of above selectors present, *SUM is selected.

DEFAULTS:

Optional keyword. If no *OUTSRF *WELL line is encountered, the invariant well property list specified under *WELL above is written to the SRF with the frequency specified by the *WSRF *WELL line. If *OUTSRF *RES does not appear in the data, the default is not to write the initial information to the Simulation Results files.

*OUTSRF *RES with no following sub-keyword is equivalent to *OUTSRF *RES *ALL. See above for the *XDR default.

Well layer output is not written to Simulator Results output unless requested via *OUTSRF *WELL *LAYER and associated sub-keywords. The default is *OUTSRF *WELL *LAYER

*NONE. If no sub-keyword follows after *OUTSRF *WELL *LAYER, then *NONE is assumed. Well and block pressures are written under both *OUTSRF *WELL *LAYER *ALL and *OUTSRF *WELL *LAYER *ALL *DOWNHOLE.

If flux-sectors are present but *OUTSRF *FLUX_SECTOR is absent, *ALL *RC *SUM is assumed.

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail in the display file may be changed during the simulation. Initial conditions can be output without doing any timesteps, and viewed, by using *CHECKONLY.

Note that velocity-dependent output, VELOCRC and STRMLN, cannot currently be used with dynamic gridding. Capillary number can only be out put when the velocity-dependent relative permeability option (VELDEPRP) is used.

EXPLANATION:

For null blocks and the fracture position for non-fractured blocks in a dual-porosity run, zero grid values are assumed.

Note that:

```
*OUTSRF *GRID *SO
*OUTSRF *GRID *SG
*OUTSRF *GRID *SW
*OUTSRF *GRID *PRES
*OUTSRF *WELL *K 'comp2' 12 5 7
*OUTSRF *WELL *BHP 'well2' 5 5 5
```

can also be written as:

```
*OUTSRF *GRID *SO *SG *SW *PRES
*OUTSRF *WELL *K 'comp2' 12 5 7 *BHP 'well2' 5 5 5
```

When *OUTSRF *PSPLIT or *OUTSRF *PSPLITNOCOMP is specified, well, group, and sector information are output to the Simulator Results files in production split form, in addition to the usual output. Production split output lists the total volumes and rates split up according to reservoir origin; that is, oil arising as a black oil (separator oil from the reservoir oil phase), oil arising as a condensate (separator oil from the reservoir gas phase), gas arising as solution gas (separator gas from the reservoir oil phase), or gas arising as free gas (separator gas from the reservoir gas phase). If *PSPLIT is used, then the compositions of the production split fluids are output as well. If an INL stream is present, then INL arising from the reservoir oil and gas phases is also reported.

Example:

To write to Simulator Results file the oil saturation at both the matrix and fracture blocks at i = 12, j = 2, k = 3, use

```
*OUTSRF *WELL *SO 12 2 3 MT *SO 12 2 3 FR
```

*OUTSRF *GRID *SATP provides only an approximate value of saturation pressure, without doing rigorous calculation for each of the grid blocks as a time-saving measure. For precise value it is recommended to output global composition, pressure and temperature of the grid block and use that in WinProp to get the saturation pressure.

Geomechanics Vector Plots

Several **SRF_GRID** items cause geomechanics information to be written to the Simulator Results file which allows RESULTS to generate the corresponding vector plot. Each of these keywords results in writing of three grid-length arrays to the Simulator Results file, one for each direction. These keywords are available only with the geomechanics module (*GEOMECH).

PRMXDIR:	Maximum principle effective stress
PRMNDIR:	Minimum principle effective stress
VDISPL:	Grid displacement

Vertical Displacement and Subsidence

Several items in the **SRF_GRID** list specify vertical formation displacement and are summarized as follows. Each quantity is the displacement of each block centre since the start of the simulation.

Quantity	Basis	Sign	Special History
VERDSPLPOR	Porosity	+ is up	yes
SUBSIDPOR	Porosity	+ is down	yes
VERDSPLGEO	Geomechanics	+ is up	yes
SUBSIDGEO	Geomechanics	+ is down	yes
VDISPL (K-dir)	Geomechanics	+ in *KDIR	no

Quantities **VERDSPLPOR** and **SUBSIDPOR** are based on a simple porosity calculation, assuming the grid bottom is fixed and all porosity change goes toward gross thickness change. These quantities are available for all porosity options including *GEOMECH. They are intended as a rough estimate of vertical displacement and may not be an accurate indicator of displacement in more complex situations. The value reported for each block is the sum of incremental values from that block's centre to the bottom of the block column. A block's incremental value is $(\varphi-\varphi_o)\cdot\Delta_z\cdot|g_z/g|$, where φ_o is initial porosity, φ is current porosity, Δ_z is Z-direction block size and g_z/g is gravity component in the Z direction. The value assigned to blocks in *HYBRID, *MINC and *SUBDOMAIN refined grids will be the value of that grid's parent block. For a zero-porosity block this incremental value is zero. This incremental value is what is reported for output quantity **SBDZ** in the subsidence sense (+ is down).

VERDSPLPOR and **SUBSIDPOR** differ only in the sign. **VERDSPLPOR** reports vertical displacement upward (heave) as positive and downward as negative, so its curve on an X-Y plot (versus time or distance) will rise and fall in the same sense as the block centre position. **SUBSIDPOR** shows vertical downward displacement (subsidence) as positive and upward displacement as negative. You can choose which quantity to plot according to your preference. Note that the grid's K direction defined by *KDIR does not enter into the definition of these output quantities.

VERDSPLGEO and **SUBSIDGEO** are based on formation strain calculated by the *GEOMECH option which estimates the movement of grid nodes (block corners) in two or three dimensions. For these quantities K-direction grid node displacements are averaged to obtain the vertical displacement of each block centre. Note that local strain calculations are performed also for zero-porosity and geomechanics-only blocks. Under suitable conditions this displacement

will correspond roughly to the porosity-based calculation described above. The relationship between **VERDSPLGEO** and **SUBSIDGEO** is the same as for the porosity-based quantities.

VDISPL is the three-dimensional displacement vector of the block centre, derived from strain calculations of the *GEOMECH option and split into X, Y and Z directions. It applies also to zero-porosity and geomechanics-only blocks. Displacement is relative to the grid origin and axes directions so its sign will depend on *KDIR. VDISPL in the Z (vertical) direction corresponds to **VERDSPLGEO** for *KDIR *UP and to **SUBSIDGEO** for *KDIR *DOWN. No special history is available for VDISPL but **VERDSPLGEO** or **SUBSIDGEO** can be used for vertical displacement.

CO_2 Inventory Output (Optional) *INVENTORY-CO2

PURPOSE:

This keyword activates the CO_2 inventory calculations and writing of the same to the main output file.

FORMAT:

*INVENTORY-CO2

DEFINITIONS:

*INVENTORY-CO2

This keyword activates the CO_2 inventory calculations. The amount of CO_2 stored in the reservoir (aquifer) under various forms are calculated: gas, liquid, trapped, super-critical, soluble, aqueous components and minerals. To output these variables to the Simulator Results file for plotting, the user must put the following keywords in the Input/Output control section:

*OUTSRF *WELL *GHGGAS *GHGLIQ *GHGTHY *GHGSCRIT *GHGSOL *GHGAQU *GHGMNR

The calculation of CO_2 inventory in aqueous and mineral components uses factors input via *CO2-AQU-FACT and *CO2-MNR-FACT in the Component Properties section.

DEFAULTS:

Optional keyword. If no *INVENTORY-CO2 line is encountered, and none of the keywords *GHGGAS *GHGLIQ *GHGTHY *GHGSCRIT *GHGSOL *GHGAQU *GHGMNR are found in the *OUTSRF *WELL line, CO_2 inventory calculations are not done.

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL.

EXPLANATIONS:

Presence of any one or more of the *GHGGAS *GHGLIQ *GHGTHY *GHGSCRIT *GHGSOL *GHGAQU *GHGMNR keywords in *OUTSRF *WELL line will activate CO_2 inventory calculations irrespective of the presence of *INVENTORY-CO2 keyword. If the factors for aqueous and mineral component CO_2 inventory are not input via *CO2-AQU-FACT and/or *CO2-MNR-FACT in the Component Properties section, the CO_2 inventory in aqueous and/or mineral components will essentially be output as zero.

Geochemical Reactions Output (Optional) *OUTREACT

PURPOSE:

*OUTREACT directs echoing-out of geochemistry reactions and output of equilibrium-rate-annihilation matrix.

FORMAT:

*OUTREACT

DEFAULTS:

Optional keyword. If the keyword *OUTREACT is absent, output of geochemical reactions and the ERA matrix will not be done, unless *OUTPRN *RES *ALL is specified.

CONDITIONS:

If present, *OUTREACT must appear in the INPUT/OUTPUT CONTROL keyword group.

EXPLANATION:

The *OUTREACT keyword directs the simulator to echo-out chemical equilibrium, mineral dissolution and precipitation, and ion-exchange reactions. It also directs the output of ERA matrix to the output file.

Keyword *OUTPRN *RES *ALL, in absence of keyword *OUTREACT, also outputs the geochemistry reactions and the ERA matrix.

Well, Aquifer and Sector Summary Tables (Optional)

***SUMMARY**

PURPOSE:

*SUMMARY controls the printing of summary tables for individual wells and the summary tables for the *OUTSRF *WELL variables.

FORMAT:

*SUMMARY

DEFAULTS:

Optional keyword. If the keyword *SUMMARY is missing, summary table will not be printed.

If *SUMMARY is present in the data set, summary tables for individual wells, aquifers, sectors, and the Simulator Results well variables (special histories) are printed at the end of the printed output. The summary tabulates information from each timestep.

CONDITIONS:

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group.

EXPLANATION:

If no summary table is printed, the missing information can be recovered from the Simulator Results files if these files are written at every timestep (i.e. with *WSRF *WELL 1).

Debug Output (Optional) *DEBUG

PURPOSE:

*DEBUG introduces specification of frequency and form of writing debugging information to the output file. Currently the writing of Jacobian matrices is the only mode enabled in GEM.

FORMAT:

*DEBUG *JDUMP ts_range iter_range blk_range iopt)

DEFINITIONS:

*JDUMP

Introduces output of the Jacobian Matrix.

ts_range

Range of timesteps (in the format i1:i2) at which the Jacobian is to be printed.

iter_range

Range of Newtonian iterations (in the format i1:i2) at which the Jacobian matrix is to be printed.

block_range

Range of global grid block indices (in the format iglb1:iglb2) for which the Jacobian is to be printed.

iopt

Option integer:

0	:	print out only r.h.s. in solver ordering for all the blocks
1	:	print out diagonals, rows, columns and r.h.s. in global ordering for the block range specified by blk_range
2	:	print out diagonal submatrices only in global ordering for blk_range
3	:	print out only blocks with zero columns in the diagonal submatrix; global ordering
4	:	print out only blocks with zero columns in the diagonal submatrix or a zero entry on the diagonal of the diagonal submatrix; global ordering
5	:	print whole Jacobian in solver ordering in a form that can be read both by eye and by the AIMSOL driver program. blk_range is ignored under this option but ts_range and it_range are honored.
5+i	:	Here i = 1, ..., 4. Print first the global order output for iopt = i as defined above, then the solver order output as defined for iopt = 5.

If iopt is other than 5 and iglb1 = 1 and iglb2 is less than or equal to 120, an incidence map for the Jacobian matrix in global ordering is also printed. This map gives the location of non-zero submatrices of the Jacobian.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is not to print the debug information.

CONDITIONS:

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail in the output-file may be changed during the simulation.

EXPLANATION:

An example of *DEBUG *JDUMP follows:

```
** Dump derivatives at timestep 7 for iterations
** 2 and 3; global blocks 244, 245 and 246

*DEBUG *JDUMP 7 2:3 244:246 1
```

This keyword must be in the I/O section or the recurrent data section. There may be more than one *DEBUG *JDUMP lines.

Seek Parent Chain of Prior Simulator Response for Geo-Post Processing (Optional) *SEEKPARENT

PURPOSE:

Keyword *SEEKPARENT allows user to enable or disable looking for parent files that generated the existing prior simulator response output (IRF/MRF or SR3 or RST) for geo-post-processing. This keyword is used only for geo-post-processing (GEOPOSTPRO) runs.

FORMAT:

*SEEKPARENT *ON | *OFF

DEFINITIONS:

*SEEKPARENT
Indicate whether seeking of parent runs is enabled or disabled.

*ON
Turns seeking of parent chain on.

*OFF
Turns seeking of parent chain off.

DEFAULTS:

Optional keyword. By default *SEEKPARENT is set to *ON, i.e., if the keyword in not present in the data file, geo-post-processing will look up the parent chain that created the specified Simulator Results/restart file.

CONDITIONS:

The keyword, if present, should appear in the INPUT/OUTPUT CONTOL section only on a separate line.

EXPLANATION:

The geo-post-processing input (*GEOPOSTPRO) can specify the starting/ending timestep numbers that were ever encountered during the prior simulator response run. The geo-post-processing run starts from the specified Simulator Results and looks up the parent chain to get the first Simulator Results file set. The scanning of the Simulator Results/restart files for post-processing begins from the first Simulator Results file set and descends down the parent chain. The advantage is that no valid pressure/temperature response gets ignored if it falls between start and end of the geo-post-processing steps. However, for larger data sets where geomechanics calculations are needed only from a particular Simulator Results/restart file, seeking of parent chain is unnecessary. In such situations it may be preferable to use *SEEKPARENT *OFF, so that no attempt is made by the geo-post-processing to reach at the top of the parent chain.

Keyword *SEEKPARENT has no effect on non-geo-post-processing runs.

Also see related keyword *GEOPOSTPRO in GEOMECHANICS section.

Example:

```
*FILENAMES    *INDEX-IN    'run_5d.irf'
... ... ...

... ... ...

*SEEKPARENT  *OFF    ** Disable seeking of parent-chain
                     ** for run_5d.
```

Reservoir Description

Notes on Reservoir Description

COMPATIBILITY

GEM uses CMG's Grid Module, as do CMG's IMEX Black Oil Simulator and STARS Steam-and-Additive Reservoir Simulator.

The functionality of the Grid Module is nearly identical for all simulators. This compatibility should help preparing and converting data sets.

Note that CMG's BUILDER tools can be used to prepare the reservoir description data.

Fundamental Grid Definition (Required) *GRID

PURPOSE:

*GRID indicates the beginning of the keyword input describing the reservoir and its grid. This grid is the main (fundamental) grid and it is the first one defined.

FORMAT:

```
*GRID   (*CART)     ni nj nk
        (*VARI)     ni nj nk
        (*CORNER)   ni nj nk
        (*RADIAL)   ni nj nk   *RW   r
```

DEFINITIONS:

*CART
: Keyword indicating a rectangular Cartesian grid, with uniform depth/uniform thickness layers.

*VARI
: Keyword indicating a rectangular grid allowing layers of variable thickness located at different depths (variable depth/variable thickness layers).

*CORNER
: Keyword indicating a corner point grid, as further described in the following.

*RADIAL
: Keyword indicating a radial-angular cylindrical grid.

ni
: Number of grid blocks in the I direction.

nj
: Number of grid blocks in the J direction.

nk
: Number of grid blocks in the K direction.

*RW
: The radius of the wellbore located in the inner- most radial grid block(s) for the *RADIAL grid.

rw
: Radius of the wellbore (m | ft). Should be nonzero.

DEFAULTS:

Required keyword. No defaults.

If *RW is missing or 0.0, a default of 0.0762 m (0.25 ft) is supplied.

CONDITIONS:

This keyword should be the first keyword in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

The keyword *GRID indicates the grid type and the number of grid blocks within the fundamental (main) grid system. There is no default and one of the aforementioned subkeywords must be entered.

The *GRID keyword defines a grid consisting of ni * nj * nk blocks. The blocks are rectangular in shape for Cartesian and variable depth/variable thickness grids. They should be close to rectangular for corner point grids. Radial-angular cylindrical grids consist of stacked circular rings, where the rings may have angular subdivisions. (The wellbore radius is excluded from the first ring.)

Under *VARI, a full flow connection is maintained between adjacent blocks in the same layer (same layer number) even if they lie at different depths; this technique represents varying depths in the reservoir.

A wellbore volume of radius "rw" is removed from the inner-most ring in all layers of *RADIAL grids. This gives the inner-most ring a circular inner boundary that connects to the wellbore. The wellbore volume is not included in the volume of any of the active blocks.

I, J, and K indices are used to identify the blocks, where I runs in the range of 1 to ni, J runs in the range of 1 to nj and K runs in the range of 1 to nk. (The notation (I,J,K) is used to refer to specific blocks.) Blocks are ordered (numbered) with I increasing fastest, J next fastest, and K slowest.

For Cartesian and variable depth/variable thickness grids, I corresponds to the "x" direction, J to the "y" direction, and K to the "z" direction, where "x", "y" and "z" refer to a standard (right handed) coordinate system in the reservoir.

Corner point cells have all their corners specified in terms of an "x-y-z" triple defined by the incoming data. However, the I-J-K numbering system is still important because the default search methods for determining connections assume that incrementing the I or J index for each cell, and searching the resulting stack of potential neighbours by sweeping through K, should find all contacting cells. Thus, I-J planes should roughly correspond to geological layers indexed by K, even if the I axis and the "x" axis, or the J axis and the "y" axis, are not aligned.

For radial-angular cylindrical grids, I corresponds to the "r" (radial) direction, J to the "angular" (theta) direction, and K to the "z" direction.

By default, K increases vertically UPWARDS (against gravity) for all non-corner point grids. This assumption can be reversed, making the K index increase downwards with gravity, by using the *KDIR *DOWN keywords (see the *KDIR keyword following).

K always increases DOWNWARDS for corner point grids (meaning *KDIR *DOWN is always assumed for corner point grids). This setting cannot be changed.

A summary follows:

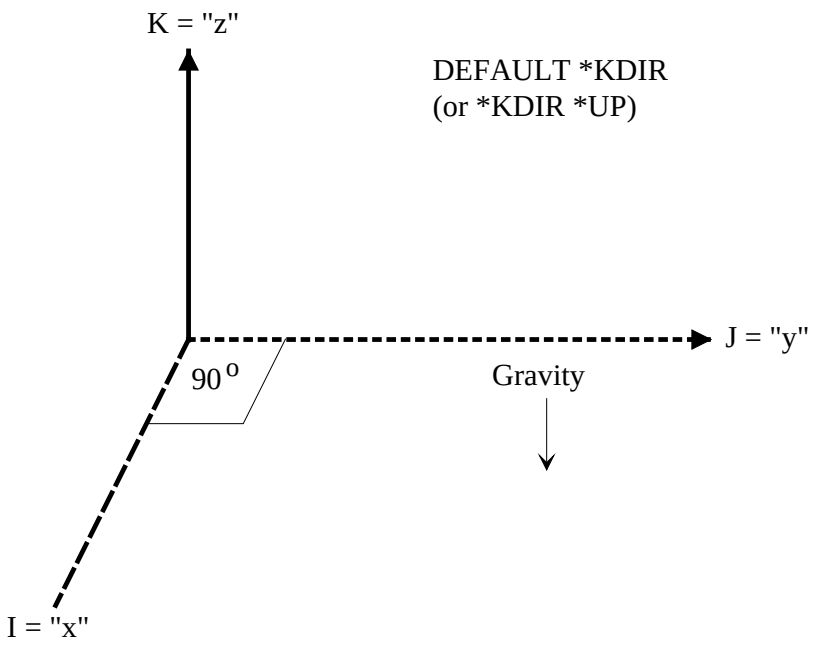
Grid Type	I	J	K	Layer Ordering
*CART	x	y	z	upward (default)
*VARI	x	y	z	upward (default)
*CORNER	x	y	z	always downwards
*RADIAL	r	theta	z	upward (default)

The grid can be tilted with respect to the gravity vector in some cases by use of the *DIP keyword (see the *DIP keyword following).

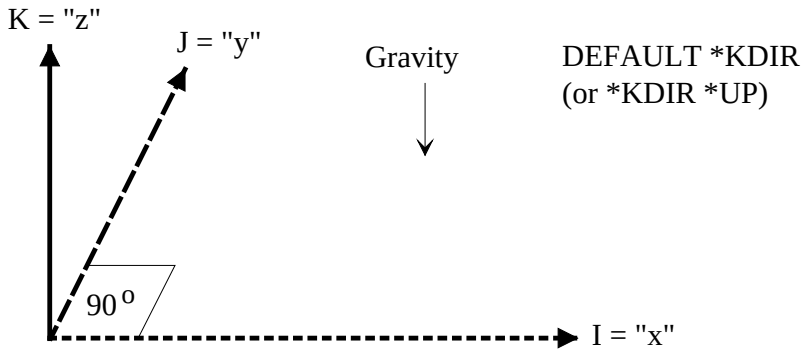
Rectangular:

If a Cartesian or variable depth/variable thickness grid is examined with the I index increasing towards the viewer, J should increase from left to right and K should increase upwards, provided that *KDIR has not been used (or *KDIR *UP is set). If a Cartesian or variable depth/variable thickness grid with *KDIR *DOWN, or a corner point grid, is examined with the J index increasing towards the viewer, I should increase from left to right and K should increase downwards.

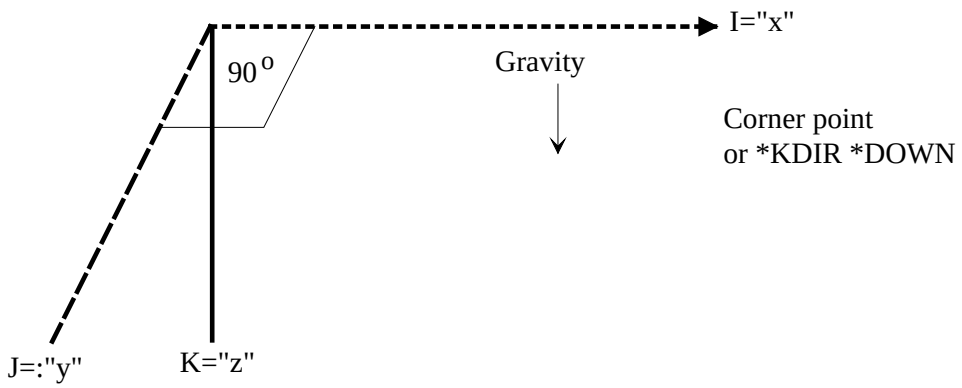
The coordinate system for a Cartesian or variable depth/variable thickness grid with the no KDIR specified (default), or *KDIR *UP specified, appears as follows:



or viewing the above from a different vantage point:



The coordinate system for a Cartesian or variable depth/variable thickness grid with *KDIR *DOWN specified, or a corner point grid, appears as follows:

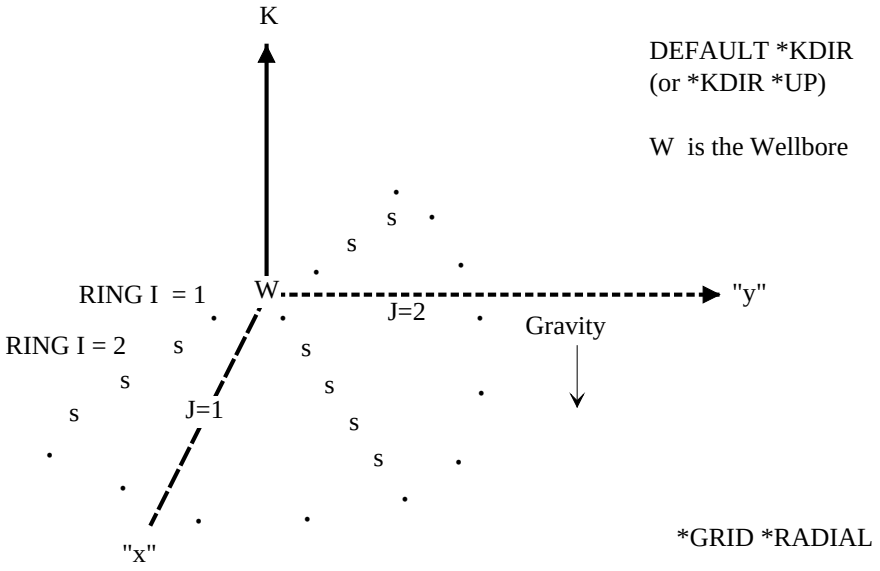


Radial:

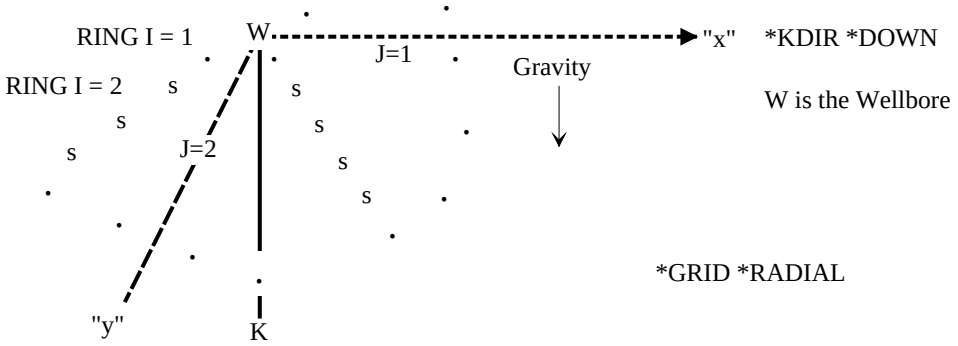
*RADIAL radial-angular cylindrical grids have rings indexed by I, counting from the innermost outwards. Each ring may be split into "nj" sectors indexed by J, counting counterclockwise around the K axis with the first sector lying over the I axis. Layers are indexed by K, starting at the reservoir bottom and counting upwards if no *KDIR keyword appears (or if *KDIR *UP is set), and counting downwards if *KDIR *DOWN is set.

Note that the first ring of *RADIAL grids is not subdivided even if nj exceeds 1, and will exclude a volume of radius "rw"; the latter volume is not assigned an I index value. Thus, the inner radius of the first (I = 1) grid block is given by "rw" and its outer radius is given by the sum of "rw" and the length of the first block in the I direction.

The coordinates for a radial-angular cylindrical grid with the no KDIR specified (default), or *KDIR *UP specified, appears as follows:



and when *KDIR *DOWN is set:



Grid block nodes are the points where the pressure is considered to be known. They are placed at block centres for rectangular grids and are determined as described in Pedrosa and Aziz for *RADIAL and *HYBRID grids.

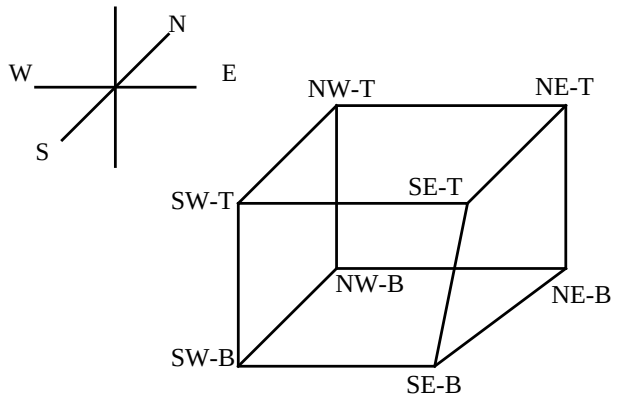
For the ordering of grid blocks, See Appendix B, FIGURE 1 for examples of a Cartesian grid system; FIGURE 2 for a cylindrical system and FIGURE 5 for a variable thickness/variable depth grid system.

Corner Point:

Corner point grids are made up of blocks which are defined by their eight corner points. Each corner point is described by giving its three coordinates, an "x"-, "y"- and "z"-coordinate, locating it in the subsurface. The "x"- and "y"- coordinates have usually been projected against a horizontal reference surface. The "z"- coordinate is usually the depth of the corner point measured downwards from that surface. Both positive or negative depths are valid, depending on the location of the reference surface with respect to the reservoir, although positive values are most common.

It takes 24 numerical values to determine a general corner point block. Certain keywords place assumptions on the corner point data however, so that it is not always necessary to read 24 values per block to define the grid. Details of corner point input are given later (see, for instance, the *COORD and *ZCORN keywords).

The following is a model for a corner point block, giving labels for its corners:

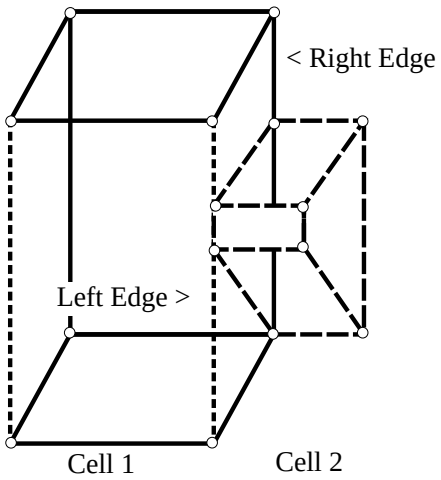


The block is the volume contained within the 6 faces, where each face is made by connecting the corner points with line segments as shown, and filling in the faces with a nonlinear (bilinear) interpolation. Faces need not be planar. The numerical results will be better if the cell shape is kept close to a rectangle however. The cell's barycentre is used as the pressure node.

GEM performs some tests to see if cells are not too distorted, however examining grids in advance with a visualization package is always recommended. Intrusions of one cell into another should be avoided.

Cells can make flow connections in two different ways. The usual situation is when four corners defining one face on one cell match (within a tolerance) the four corners defining a face on another cell. Projections (onto both sides) of the common face will be used in the transmissibility calculation for flow between the cells.

The other situation involves vertical faulting. If one of the side faces (not the top or bottom) of a cell is planar, and it overlaps a planar side face on another cell, and the two left and the two right edges of each face are collinear (that is, they lie on the same line), then a flow connection will be made based on the overlap area. An example follows:



Note that a cell's vertical faces need not always be planar although this is often the case. Corner point grids defined using the *COORD keyword (see later) involve cells that share vertical edges, and if the *COORD lines are parallel, their side faces will be planar. Note also that the face overlap need only occur to within a tolerance.

The tolerance level for the above cases is controlled by *CORNER-TOL (see following). Note that if cells intrude on each other to too large a degree, then corner points and faces cannot make contact and flow will not occur.

Examples:

a) Rectangular Cartesian grid with ten blocks in the "x" direction, five blocks in the "y" direction, and four blocks in the "z" direction (ni=10, nj=5, nk=4). Enter:

```
*GRID *CART 10 5 4
```

b) Variable depth/variable thickness rectangular grid with ten blocks in the "x" direction, three blocks in the "y" direction, and one layer in the "z" direction (ni=10, nj=3, nk=1). Enter:

```
*GRID *VARI 10 3 1
```

c) Corner point grid with 20 blocks in the "x" direction, 20 blocks in the "y" direction, and 5 layers (ni=20, nj=20, nk=5). Enter:

```
*GRID *CORNER 20 20 5
```

d) Radial-angular cylindrical grid with 15 blocks in the radial direction, no radial subdivisions, and 5 layers (ni=15, nj=1, nk=5). Enter:

```
*GRID *RADIAL 15 1 5
```

K Direction Index (Optional) *KDIR

PURPOSE:

*KDIR controls whether the deeper or the shallower grid blocks have the larger layer indices (where layer indices are usually denoted by K).

FORMAT:

```
*KDIR    (*UP)
         (*DOWN)
```

DEFINITIONS:

*UP

Indicates the K index increases upward, so larger layer numbers K mean shallower blocks.

*DOWN

Indicates the K index increases downward, so larger layer numbers K mean deeper blocks.

DEFAULTS:

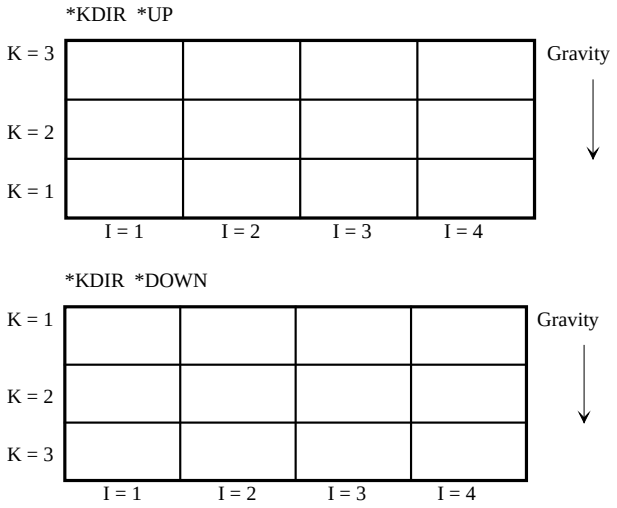
Optional keyword. Default: *KDIR *UP, except for Corner Point Grids (*GRID *CORNER) when *KDIR *DOWN is the default. The latter can not be changed.

CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

See the *GRID description above for details.



Convert Cartesian Grid to Corner Point (Optional)

***CONVERT-TO-CORNER-POINT**

PURPOSE:

Internally convert a Cartesian grid type to corner-point type.

FORMAT:

*CONVERT-TO-CORNER-POINT

DEFAULTS:

If this keyword is absent then no grid type conversion is done.

CONDITIONS:

This keyword converts only Cartesian grid types to corner-point type.

The option is not allowed when *KDIR *UP is used together with any natural fracture option.

This option cannot be used together with other grid modification keywords like *PINCHOUTARRAY.

EXPLANATION:

In a grid of type *VARI it is possible that the corners of adjacent blocks do not coincide. This condition can occur in Cartesian-based *VARI type grids that are known generally as variable-thickness and variable-depth.

Keyword *CONVERT-TO-CORNER-POINT converts type *VARI grid data with this condition to a corner-point grid that does not have this condition. The conversion is performed entirely at run time and does not appear in the data file or simulator data echo. Each new single corner location is simply the average of the previous different corner locations. Volumes and transmissibilities of individual grid blocks will differ from the previous grid, but fractional changes should be reasonable for a well-formed grid. Global quantities like total pore volume should be little different.

More extreme variable depth and thickness situations may not convert satisfactorily, in which case some manual adjustment of the original data is recommended. In addition, this conversion does not preserve the deliberate modelling of faults. In all cases, you can view both grid types in Results using data sets with and without this keyword.

Nine-Point Spatial Discretization (Optional) *NINEPOINT

PURPOSE:

*NINEPOINT controls the nine-point spatial discretization option.

FORMAT:

*NINEPOINT (*OFF | *IJ | *IK)

DEFINITIONS:

*OFF
Five-point discretization is used in all three planes.

*IJ
Nine-point discretization is used for the I-J plane, and five-point discretization is used for the J-K and I-K planes. This option is not available for *GRID option *RADIAL.

*IK
Nine-point discretization is used for the I-K plane, and five-point discretization is used for the J-K and I-J planes.

DEFAULTS:

If *NINEPOINT is absent then *NINEPOINT *OFF is assumed.

CONDITIONS:

The *NINEPOINT option may not be used together with *REFINE or natural fracture grid options.

Sub-option *IJ may not be used with *GRID *RADIAL.

Sub-option *IK may not be used together with block pinch-outs.

EXPLANATION:

Nine-Point Method

The transmissibilities in the nine-point finite difference approximation are calculated according to the Amoco method (SPE 16975, 1991).

Unlike STARS GEM does not support nine-point discretization for thermal conduction. For most field scale simulations, convection is the main heat transport mechanism and so nine-point discretization for thermal conduction is generally is not needed.

Pseudo 1-D Modelling

The *NINEPOINT option will give unexpected results if the grid is not constructed properly when modelling a pseudo-1D problem with a 2-D grid. The two reservoir boundaries that are parallel to the pseudo-1D direction must have their nodes on the reservoir boundary. In other words, the reservoir must be treated like a repeated pattern. Use the geometry modifiers *VAMOD to trim these boundary blocks in half so that the block nodes fall on the reservoir boundary.

Corner-point Grids

The nine-point formulation is based on orthogonal grid assumptions that are built into *GRID options *CART (IJ and IK planes) and *RADIAL (IK plane). On the other hand, corner-point grids generally are non-orthogonal and so violate nine-point assumptions. However, the target plane of a corner-point grid may in fact be orthogonal or only slightly non-orthogonal, in which case the nine-point option can be used with some confidence. Care should be taken that the plane to which 9-point is applied is not excessively non-orthogonal. For example, it is safe to use *NINEPOINT *IJ with a corner-point grid that is non-orthogonal only in the K direction, that is, the grid in the IJ plane looks Cartesian.

Block Dimensions for the I Direction (Conditional) *DI

PURPOSE:

*DI signals input of an array of grid block lengths for the I direction. For rectangular grids, the values are block widths measured in the I direction and for radial-angular cylindrical grids, the values are block widths measured in the radial direction.

ARRAY:

*DI

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. It is required for all grids except corner point grids, for which it is optional.

Several different methods are available for specifying block dimensions for corner point grids, including the use of the *DI keyword. See the *ZCORN keyword for a further discussion of corner point input.

All array reading options are valid; however, all blocks with the same I index must have the same block length.

EXPLANATION:

The keyword *DI defines the dimensions of the grid blocks in the I direction. The unit is (m | ft). The most commonly used sub-keywords are *CON (for uniform block sizes) and *IVAR (for non-uniform block sizes).

The acceptable range of values for block lengths in the I direction is 10^{-4} m ($3.23 \cdot 10^{-4}$ ft) to 10^{9} m ($3.28 \cdot 10^{9}$ ft).

Local Grid Refinement

For a locally refined grid, the default action is to divide the parent block into child blocks of uniform size. To over-ride this default, use sub-keyword *RG to specify the ratios of child block sizes. This ratio method makes it easy to specify child block sizes when parent block size is odd or unknown.

For example, consider parent block (4,6,9) that is 12.0 long and is refined into 3 child blocks. The default action is to make each child block 12.0/3 = 4.0 long. To specify child block sizes of 5.0, 2.0 and 5.0, use "*DI *RG 4 6 9 *IVAR 5.0 2.0 5.0". Alternatively you can use "*DI *RG 4 6 9 *IVAR 2.5 1.0 2.5" which has the same ratios.

Examples:

a) When I-direction grid-block sizes for a problem with *ni*=10 are:

1000,1000,1500,400,400,400,400,400,1000,1000.

Use:

*DI *IVAR 2*1000 1500 5*400 2*1000

b) When I-direction size of each grid block is 1200, use:

*DI *CON 1200

-or-

*DI *CON
1200

c) When block (3,5,8) with size 10 is refined non-uniformly into sizes 4,2,4, use

```
*REFINE 3 5 8 *INTO 3 3 2
*DI *CON 10
*DI *RG 3 5 8 *IVAR 4.0 2.0 4.0
```

This alternate for the last line specifies the same block sizes

```
*DI *RG 3 5 8 *IVAR 2 1 2
```

d) The widths of *ni*=10 radial-angular cylindrical blocks measured radially are:

2.00	2.32	5.01	10.84	23.40
50.55	109.21	235.92	509.68	1101.08

Use:

*DI *IVAR

2.0	2.32	5.01	10.84	23.4
50.55	109.21	235.92	509.68	1101.08

Note that the first block's inner radius is at (rw) and its outer radius is (rw+2), where (rw) is the value specified by *RW on the *GRID *RADIAL data input line.

Block Dimensions for the J Direction (Conditional) *DJ

PURPOSE:

*DJ signals input of an array of grid block lengths for the J direction. For rectangular grids, the values are block widths measured in the J direction and for radial-angular cylindrical grids, the values are angular extents for portions of the subdivided rings, expressed in degrees.

ARRAY:

*DJ

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. It is required for all grids except corner point grids, for which it is optional.

Several different methods are available for specifying block dimensions for corner point grids, including the use of the *DJ keyword. See the *ZCORN keyword for a further discussion of corner point input.

All array reading options are valid; however, all blocks with the same J index must have the same block length/angular extent. The most commonly used array reading subkeywords are the *JVAR and *CON options.

EXPLANATION:

For all grid types except cylindrical, keyword *DJ specifies grid-block dimensions in the J direction in unit (m | ft). The acceptable range of values for block lengths in the J direction is 10^{-4} m ($3.23 \cdot 10^{-4}$ ft) to 10^{9} m ($3.28 \cdot 10^{9}$ ft).

For cylindrical R-θ-Z grid type (*GRID *RADIAL), keyword *DJ specifies grid-block dimensions in the θ direction in unit of degrees. The sum of the J-direction dimensions must no exceed 360 degrees. When *nj*>1 and the sum of the J-direction dimensions is 360 degrees (full circle), block (I,1,K) is connected to block (I,*nj*,K), I=2:*ni*, K=1:*nk*. Note that the inner-most blocks (I=1) have no θ subdivisions.

The most commonly used sub-keywords are *CON (for uniform block sizes) and *JVAR (for non-uniform block sizes).

Local Grid Refinement

The comments in section **Local Grid Refinement** for keyword *DI and sub-keyword *IVAR apply for *DJ and sub-keyword *JVAR, respectively.

Examples:

a) J direction grid increments for a Cartesian grid with *nj*=10 are:

2000,2000,2500,4000,1500,1500,400,400,1000,2000.

Use:

*DJ *JVAR
2*2000 2500 4000 2*1500 2*400 1000 2000

b) J direction grid increments for a Cartesian grid are all 2200. Use:

*DJ *CON 2200

-or-

*DJ *CON
2200

c) When block (3,5,8) with size 10 is refined non-uniformly into sizes 4,2,4, use

```
*REFINE 3 5 8 *INTO 3 3 2
*DJ *CON 10
*DJ *RG 3 5 8 *JVAR 4.0 2.0 4.0
```

This alternate for the last line specifies the same block sizes

```
*DJ *RG 3 5 8 *JVAR 2 1 2
```

d) A *RADIAL grid has no angular subdivisions (*nj* = 1). Use:

*DJ *CON 360

Block Dimensions for the K Direction (Conditional) *DK

PURPOSE:

*DK signals input of an array of (gross) grid block thicknesses measured in the K direction.

ARRAY:

*DK

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. It is required for all grids except corner point grids. K direction grid block lengths should not be specified with the *DK keyword for corner point grids; see the corner point keyword descriptions following, such as *ZCORN.

All array reading options are valid; however, recall that all blocks with the same K index must have the same thickness for Cartesian grids; that is, the layers in Cartesian grids have uniform thicknesses. Different thicknesses in the same layer (same K) are permitted for *GRID *VARI and *GRID *RADIAL however.

Blocks can be assigned a zero thickness if they are to be considered as pinched out (see the discussions for *PINCHOUTARRAY and *PINCHOUT-TOL following).

EXPLANATION:

This keyword defines the dimensions of the grid blocks in the K direction. The unit is (m | ft). For cylindrical R-θ-Z grid type (*GRID *RADIAL), use *DK to specify block sizes in the axial (Z) direction.

The acceptable range of values for block lengths in the K direction is 10^{-4} m ($3.23 \cdot 10^{-4}$ ft) to 10^{9} m ($3.28 \cdot 10^{9}$ ft).

The most commonly used sub-keywords are *CON (for uniform block sizes) and *KVAR (for non-uniform block sizes).

Local Grid Refinement

The comments in section **Local Grid Refinement** for keyword *DI and sub-keyword *IVAR apply for *DK and sub-keyword *KVAR, respectively.

Examples:

a) K direction grid increments for a Cartesian grid with *nk*=8 are:

```
20,20,25,40,15,45,45,45.
```

Use:

```
*DK *KVAR
2*20  25  40  15  3*45
```

b) K direction grid increments for a Cartesian grid are all 22. Use:

```
*DK *CON  22.0
-or-
*DK *CON
22.0
```

c) When block (3,5,8) with size 10 is refined non-uniformly into sizes 2.5, 7.5, use

```
*REFINE 3 5 8 *INTO 3 3 2
*DK *CON 10
*DK *RG 3 5 8 *KVAR 2.5 7.5
```

This alternate for the last line specifies the same block sizes

```
*DK *RG 3 5 8 *KVAR 1 3
```

Depth (Conditional) *DEPTH

PURPOSE:

*DEPTH indicates input of a reservoir depth for a single grid block. This depth is usually to be measured to the block's centre, unless *TOP appears (see below).

FORMAT:

```
*DEPTH    (*TOP)      i  j  k  depth
          (*CENTRE)
```

DEFINITIONS:

*TOP
: Subkeyword indicating that the depth is to the top (centre of the top face) of the reference block.

*CENTRE
: Subkeyword indicating that the depth is to the centre of the reference block.

i
: I direction index of the reference block.

j
: J direction index of the reference block.

k
: K direction index of the reference block.

depth
: Depth to the centre (or top if *TOP is used) of the reference block in the reservoir (m | ft). The value may be of any sign.

DEFAULTS:

Conditional keyword. No defaults. *CENTRE is assumed if *TOP does not appear.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. One of *DEPTH, *DTOP, *DEPTH-TOP or *PAYDEPTH must be specified for *GRID *CART, *GRID *VARI, or *GRID *RADIAL. This keyword should not be used with corner point grids. If depth modifications are required for corner point grids, the *DEPTH-TOP or *PAYDEPTH keywords can be used.

EXPLANATION:

Depths are measured downwards from a horizontal reference surface. The I, J, K indices describe a grid block whose depth is known, the depth being measured to the centre/top of the grid block. The value may be positive or negative depending on the location of the reference surface, although positive values are more common.

Some kind of depth information is required for all simulations.

When *DEPTH is used, depths are assigned to all blocks based on the value provided. The calculation is made based on the blocks' thicknesses (*DK keyword) and the dip angles provided by the *DIP keyword (see *DIP keyword description following).

The subkeyword *CENTRE can be used if desired, although *DEPTH defaults to centre.

Example:

```
*DEPTH 1 1 1 2000.0
```

The acceptable range of values for depth is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

Depth to the Tops of Grid Blocks (Conditional) *DTOP

PURPOSE:

*DTOP indicates input of (ni * nj) depth values that provide the depths to the centre of the top face of each grid block in the top layer of the grid.

ARRAY:

*DTOP

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. One of *DEPTH, *DTOP, *DEPTH-TOP or *PAYDEPTH must be specified for *GRID *CART, *GRID *VARI, or *GRID *RADIAL. Use of this keyword, *DEPTH-TOP or *PAYDEPTH, is recommended for *GRID *VARI. *DEPTH-TOP and *PAYDEPTH, but not *DTOP, can be used for corner point grids if depth modifications are required.

If this keyword is used with *GRID *CART, the values in the *DTOP array should all be the same.

No array qualifiers or array reading options are permitted for this particular array keyword. A fixed number of values (ni * nj) is always expected.

EXPLANATION:

This keyword is usually used to define the depths of grid blocks for a variable depth/variable thickness grid (*GRID *VARI). A total of ni * nj depth values must be entered. The values are to be measured downwards from a horizontal reference surface to the centre of the tops of the grid blocks in the upper-most layer. The values may be positive or negative depending on the location of the reference surface. They are to be entered row by row with the I index changing fastest, and the J index slowest.

Note that the K index assumed for this array will be that of the uppermost layer; that is, it will be K = nk if *KDIR does not appear in the data set, or if *KDIR *UP has been specified, or it will be K = 1 if *KDIR *DOWN appears.

Depths are assigned to all blocks based on the depths provided by this keyword and the blocks' thicknesses (*DK keyword).

Example:

A variable depth/variable thickness grid with ni=6, nj=4 and nk=2 might use the following:

```
*DTOP
1000.0  1300.0  1250.0  1100.0  1200.0  1070.0
1070.0  1090.0  1080.0  1110.0  1120.0  1200.0
1000.0  1200.0  1110.0  1200.0  1200.0  1190.0
1070.0  1100.0  1100.0  1170.0  1070.0  1070.0
```

The acceptable range of values for depths is:

	SI m	Field ft
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

Depths to Centre of Pay (Conditional) *PAYDEPTH

PURPOSE:

*PAYDEPTH indicates input of depths to the centre of the net pay for each grid block in the reservoir. Since net pay is assumed to be centered in the grid blocks, this keyword has the effect of setting centered depths for all blocks.

ARRAY:

*PAYDEPTH

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. One of *DEPTH, *DTOP, *DEPTH-TOP or *PAYDEPTH must be specified for *GRID *CART, *GRID *VARI, or *GRID *RADIAL. Use of this keyword, or *DTOP, is recommended for *GRID *VARI.

*DEPTH-TOP or *PAYDEPTH can be specified for corner point grids. These keywords will override depths calculated from the "z" components of the corner point locations.

If this keyword is used with *GRID *CART, the depths in each layer (blocks with the same K index) should agree, and the depth differences between layers should be consistent with the gross thicknesses (*DK keyword).

EXPLANATION:

This keyword defines the depths to the pay of each individual grid block. All array qualifiers and array reading options are allowed for specifying the ni * nj * nk values.

The values are to be measured downwards from a horizontal reference surface to the centre of the grid block, which is where the net pay is assumed to be positioned. The values may be positive or negative depending on the location of the reference surface in the reservoir, although positive values are most common.

Since it is assumed that the (net) pay is centered in the block, the *PAYDEPTH array's depths can be directly assigned to each block's node.

When used for corner point grids, the paydepth values override depths calculated from the "z" components of the corner point locations. If a *PAYDEPTH value is not set for certain cells (as when *PAYDEPTH is used with the *IJK option and not all cells are touched) then depths for the remaining cells will revert to those taken from the "z" components. The actual corner point locations are not altered by *DEPTH-TOP or *PAYDEPTH, and grid visualizations are unaffected. Only the "Depth to Centers" array in the output echo (use *OUTPRN *RES *ALL) shows the results of using *DEPTH-TOP or *PAYDEPTH with corner point grids. Use of *DEPTH-TOP or *PAYDEPTH with corner point grids works like a vertical position modifier for the cells.

Example:

A variable depth/variable thickness grid with ni=6, nj=4 and nk=2 might use the following:

```
*PAYDEPTH *ALL
1000.0  1300.0  1250.0  1100.0  1200.0  1070.0
1070.0  1090.0  1080.0  1110.0  1120.0  1200.0
1000.0  1200.0  1110.0  1200.0  1200.0  1190.0
1070.0  1100.0  1100.0  1170.0  1070.0  1070.0
2000.0  2300.0  2250.0  2100.0  2200.0  2070.0
2070.0  2090.0  2080.0  2110.0  2120.0  2200.0
2000.0  2200.0  2110.0  2200.0  2200.0  2190.0
2070.0  2100.0  2100.0  2170.0  2070.0  2070.0
```

The acceptable range of values for depths is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

Depths to Top of Block (Conditional) *DEPTH-TOP

PURPOSE:

*DEPTH-TOP indicates input of depths to the top of each grid block in the reservoir.

ARRAY:

*DEPTH-TOP

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. One of *DEPTH, *DTOP *DEPTH-TOP or *PAYDEPTH must be specified for *GRID *CART, *GRID *VARI, or *GRID *RADIAL. Use of this keyword, *PAYDEPTH or *DTOP, is recommended for *GRID *VARI.

*DEPTH-TOP can be specified for corner point grids. The *DEPTH-TOP values will override depths calculated from the "z" components of the corner point locations.

If this keyword is used with *GRID *CART, the depths in each layer (blocks with the same K index) must agree, and the depth differences between layers must be consistent with the gross thicknesses (*DK keyword).

EXPLANATION:

This keyword defines the depths to the top of each individual grid block. All array qualifiers and array reading options are allowed for specifying the ni * nj * nk values.

The values are to be measured downwards from a horizontal reference surface to the top of the grid block. The values may be positive or negative depending on the location of the reference surface in the reservoir, although positive values are most common.

When used for corner point grids, the depth to top values override depths calculated from the "z" components of the corner point locations. If a *DEPTH-TOP value is not set for certain cells (as when *DEPTH-TOP is used with the *IJK option and not all cells are touched) then depths for the remaining cells will revert to those taken from the "z" components. The actual corner point locations are not altered by *DEPTH-TOP, and grid visualizations are unaffected. Only the "Depth to Centers" array in the output echo (use *OUTPRN *RES *ALL) shows the results of using *DEPTH-TOP with corner point grids. Use of *DEPTH-TOP with corner point grids works like a vertical position modifier for the cells.

Example:

A variable depth/variable thickness grid with ni=6, nj=4 and nk=2 might use the following:

```
*DEPTH-TOP *ALL
1000.0  1300.0  1250.0  1100.0  1200.0  1070.0
1070.0  1090.0  1080.0  1110.0  1120.0  1200.0
1000.0  1200.0  1110.0  1200.0  1200.0  1190.0
1070.0  1100.0  1100.0  1170.0  1070.0  1070.0
2000.0  2300.0  2250.0  2100.0  2200.0  2070.0
2070.0  2090.0  2080.0  2110.0  2120.0  2200.0
2000.0  2200.0  2110.0  2200.0  2200.0  2190.0
2070.0  2100.0  2100.0  2170.0  2070.0  2070.0
```

The acceptable range of values for depths is:

	SI **m**	**Field** **ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

Grid Tilt Angles (Conditional) *DIP

PURPOSE:

*DIP specifies the input of dip angles.

FORMAT:

*DIP	idip	jdip	(for *GRID *CART)
*DIP	idip		(for *GRID *RADIAL)

DEFINITIONS:

idip
: Tilt angle in degrees of the I axis above the horizontal.

jdip
: Tilt angle in degrees of the J axis above the horizontal. If jdip is absent, jdip = 0 is assumed. For *GRID *RADIAL, jdip is not used and so is ignored.

DEFAULTS:

If *DIP is absent, then idip = 0 and jdip = 0 are assumed.

If *DIP appears but jdip is absent, jdip = 0 is assumed.

CONDITIONS:

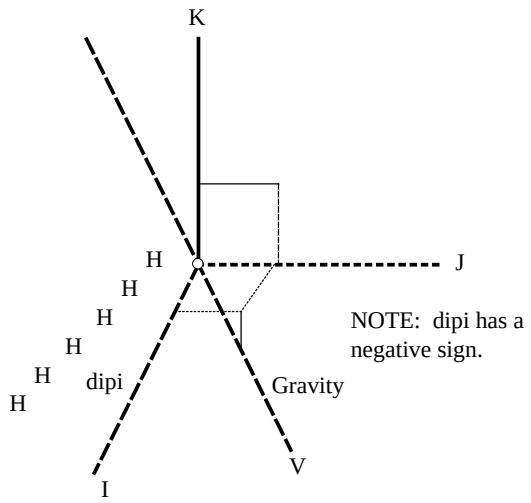
Keyword *DIP is used only with *GRID *CART and *GRID *RADIAL.

EXPLANATION:

The *DIP keyword allows a grid to tilt with respect to the gravity direction. It is generally used with *DEPTH for *GRID *CART and *GRID *RADIAL grids. *GRID *VARI and *GRID *CORNER have sufficient depth input flexibility provided by alternate keywords, so *DIP is not usually required for these grids. (The effect of using the *DEPTH and *DIP keywords could be duplicated by using an externally generated array of depths entered via the *DEPTH-TOP or *PAYDEPTH keywords.) For *GRID *CORNER, any tilt must be included in the corner locations.

The angles "idip" and "jdip" are measured in degrees and should lie between -90 to +90 degrees.

A figure for calculating "idip" follows, where the reference horizontal axis "H H H" is at right angles to the gravity vector and lies in the plane of the gravity vector and the I = "x" axis. This example shows a negative dip angle (I lies below H):



A similar construction involving the J axis obtains "jdip".

For *RADIAL grids, it is assumed that the centre of the first sector lies in the plane defined by the K axis and the gravity vector, and that the I axis is drawn down the centre of the first sector. Thus, a rotated version of the above figure provides a description of how to measure "idip".

Note that RESULTS 3D does not display the reservoir tilted with *DIP angles. For *GRID *CART it is recommended that tilts near 90 degrees be accomplished by changing the grid axis being used for vertical. For example, a vertical 1D grid can use ni = nj = 1 and no tilt as opposed to nj = nk = 1 with 90 degrees tilt.

Tilting a grid results in non-uniform depths in a K layer. When a radial grid is tilted, depth variations are meaningful only when nj > 1, that is there are angular divisions. Even when nj > 1 the innermost radial block has no angular division, so its node location and hence depth is somewhat arbitrary – especially when its innermost radius specified via *RW is non-zero. If modeling of depth near the tilted grid center is important, specify a small non-zero radial size for the I=1 block. This will make the I=2 block ring appear to be innermost; a well in the K-axis location should be completed in all nj blocks in this ring.

Example:

```
*DIP  1.0 0.0
```

The acceptable range of values for dip angles is:

	SI	Field
min	-90.0	-90.0
max	90.0	90.0

Corner-point Node Locations (Conditional) *NNODES, *NODEX, *NODEY, *NODEZ, *NODES, *BLOCKS, *BLOCKDFL

PURPOSE:

Specify the location and usage of each node of a corner-point grid.

FORMAT:

*NNODES *nnodes*
*BLOCKDFL

ARRAY:

*NODEX
*NODEY
*NODEZ
or
*NODES

*BLOCKS (*RG *uba*)

DEFINITIONS:

nnodes
Total number of unique nodes (corner points) in the fundamental grid.

*NODEX
Array of X coordinates (m | ft), one value for each of *nnodes* nodes. The order of values is given by the **node list**.

*NODEY
Array of Y coordinates (m | ft), one value for each of *nnodes* nodes. The order of values is given by the **node list**.

*NODEZ
Array of Z coordinates (m | ft), one value for each of *nnodes* nodes. The order of values is given by the **node list**.

*NODES
Array of X, Y and Z coordinates (m | ft), one group of three values for each of *nnodes* nodes. Each group of three values is the node's X, Y and Z coordinates, respectively. The order of groups is given by the **node list**.

*BLOCKS (*RG *uba*)
Array of **node numbers**, one group of eight values for each block in the grid. The order of blocks is given by the **block list**. Each group of eight values specifies the block's eight nodes (corner points). All instances of *BLOCKS refer to the same node list, that is, the node list contains nodes for both fundamental and refined grids. See EXPLANATION, below.

When *RG is absent, this keyword refers to the fundamental grid which has *ni·nj·nk* blocks specified by *GRID *CORNER.

When *RG is present, this keyword refers to a locally refined grid which has *nir·njr·nkr* child blocks specified by *REFINE for parent block *uba*.

*BLOCKDFL

Use this keyword instead of *BLOCKS for the fundamental grid, if the relationship between blocks and nodes is the same as for a Cartesian grid with no zero-size blocks. Such a grid has no collapsed blocks and no split nodes, so there will be $(ni+1)\cdot(nj+1)\cdot(nk+1)$ unique nodes. It is assumed that the node list is in "natural order". See EXPLANATION, below.

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

These keywords are available only with *GRID *CORNER.

These keywords should not be combined with any other keyword that specifies corner-point locations (e.g., *CORNERS).

Keywords *NODEX, *NODEY and *NODEZ must appear together, if at all.

Keyword *NODES must not appear with *NODEX, *NODEY and *NODEZ.

If *BLOCKS *RG *uba* is present, a locally refined grid must have been specified for parent block *uba* via *REFINE.

If parent block *uba* is refined via *REFINE but *BLOCKS *RG *uba* is absent, nodes are created internally to build a refined grid corresponding to refined-grid block sizes given either by default (uniform) or direct input (*DI *RG *uba*, etc.).

Keyword *BLOCKDFL is available only for the fundamental grid.

EXPLANATION:

See the EXPLANATION for *GRID *CORNER for the description of the notation used here for referring to block nodes (corner points).

Keyword *NNODES specifies the number of unique nodes in the grid. A fundamental Cartesian grid with no zero-size blocks has $(ni+1)\cdot(nj+1)\cdot(nk+1)$ unique nodes. A corner-point grid is more flexible so *nnodes* depends upon the actual sharing of nodes between neighbouring blocks. Using "split nodes" to model a fault or separated layers can increase *nnodes*, while collapsing blocks to model a pinched-out layer can decrease *nnodes*.

Node List and Node Numbers

The **node list** consists of the *nnodes* nodes in a certain order; the n'th node in the list is identified as **node number** "n". Keyword *BLOCKS (both with and without *RG) refers to those node numbers.

The ordering of nodes in the node list is arbitrary. However, it is very common to use "natural order" where the X coordinate changes fastest and the Z coordinate changes the slowest. Keyword *BLOCKDFL assumes that the node list for the fundamental grid is in natural order.

Block Order

The **block order** used by keyword *BLOCKS is "natural order", that is, the grid's I index varies the fastest and K index varies the slowest. For example, when $ni = nj = nk = 2$ the natural block order is

(1,1,1), (2,1,1), (1,2,1), (2,2,1), (1,1,2), (2,1,2), (1,2,2) and (2,2,2).

This natural block order applies also to child blocks in locally refined grids. For example, for

*REFINE 27,14,9 *INTO 2 2 2

the natural block order is

(27,14,9/1,1,1), (27,14,9/2,1,1), (27,14,9/1,2,1), (27,14,9/2,2,1),

(27,14,9/1,1,2), (27,14,9/2,1,2), (27,14,9/1,2,2) and (27,14,9/2,2,2).

Local Node Order

In keyword *BLOCKS, each block's group of eight node numbers specifies the block's eight nodes (corner points). The order of appearance of these node numbers in the group determines the role each node plays in relation to the block's neighbours. Refer to the diagram in section **Corner Point** in the EXPLANATION for *GRID. The eight nodes are identified with the corners of the illustrated block in this order:

SW-B, SE-B, NE-B, NW-B, SW-T, SE-T, NE-T and NW-T.

Local Grid Refinement

A locally refined grid is defined by keyword *REFINE which specifies its I-J-K structure. There are three methods available to specify the geometry of the individual child blocks, listed here in order of decreasing priority (first is highest priority).

*BLOCKS *RG	All required nodes must be in the input node list.
*DI *RG, etc.	Creates internally any required nodes that are not in the input node list. See EXPLANATION for *DI.
Default	Creates internally any required nodes that are not in the input node list. See EXPLANATION for *DI.

Examples:

Provide node location data for a grid with $ni = 3$, $nj = 2$ and $nk = 1$ in a reservoir dipping in the X direction. Block sizes are 100 units in the X direction, 200 units in the Y direction and 30 units in the Z direction. The grid origin is (0,0,2000) and the Z axis increases downward. There are $(ni+1)\cdot(nj+1)\cdot(nk+1) = 24$ unique nodes and $ni\cdot nj\cdot nk = 6$ blocks. The block-node relationship is the same as for a Cartesian grid.

```
*GRID *CORNER 3 2 1
*NNODES 24
*NODES  **   X     Y     Z
             0     0  2000  ** node #1
           100     0  2005  ** node #2
           200     0  2010  ** node #3
           300     0  2015  ** node #4
             0   200  2000  ** node #5
           100   200  2005  ** node #6
           200   200  2010  ** node #7
           300   200  2015  ** node #8
             0   400  2000  ** node #9
           100   400  2005  ** node #10
           200   400  2010  ** node #11
           300   400  2015  ** node #12
             0     0  2030  ** node #13
           100     0  2035  ** node #14
           200     0  2040  ** node #15
           300     0  2045  ** node #16
             0   200  2030  ** node #17
           100   200  2035  ** node #18
           200   200  2040  ** node #19
           300   200  2045  ** node #20
             0   400  2030  ** node #21
           100   400  2035  ** node #22
           200   400  2040  ** node #23
           300   400  2045  ** node #24
*BLOCKS
** SWB SEB NEB NWB SWT SET NET NWT
     1   2   6   5  13  14  18  17  ** (1,1,1)
     2   3   7   6  14  15  19  18  ** (2,1,1)
     3   4   8   7  15  16  20  19  ** (3,1,1)
     5   6  10   9  17  18  22  21  ** (1,2,1)
     6   7  11  10  18  19  23  22  ** (2,2,1)
     7   8  12  11  19  20  24  23  ** (3,2,1)
```

The following is an alternate form for the fundamental-grid data above.

```
*GRID *CORNER 3 2 1
*NNODES 24
*NODEX  0   100  200  300
        0   100  200  300
        0   100  200  300
        0   100  200  300
        0   100  200  300
        0   100  200  300
*NODEY  4*0 4*200 4*400
        4*0 4*200 4*400
*NODEZ  2000  2005  2010  2015
        2000  2005  2010  2015
        2000  2005  2010  2015
        2030  2035  2040  2045
        2030  2035  2040  2045
        2030  2035  2040  2045
*BLOCKDFL  ** Cartesian-like grid
```

The following is the above fundamental grid with a locally refined grid added.

```
*GRID *CORNER 3 2 1
*NNODES 34  ** 24 + 10
*NODEX  0   100  200  300
        0   100  200  300
        0   100  200  300
        0   100  200  300
        0   100  200  300
        0   100  200  300
        50  0    50   100  50
        50  0    50   100  50
*NODEY  4*0 4*200 4*400
        4*0 4*200 4*400
        0   3*100  200
        0   3*100  200
*NODEZ  2000  2005  2010  2015
        2000  2005  2010  2015
        2000  2005  2010  2015
        2030  2035  2040  2045
        2030  2035  2040  2045
        2030  2035  2040  2045
        2002.5  2000  2002.5  2005 2002.5
        2032.5  2030  2032.5  2035 2032.5
*BLOCKDFL  ** Cartesian-like grid
*REFINE 1 1 1 *INTO 2 2 1
*BLOCKS *RG 1 1 1
     1  25  27  26  13  30  32  31 ** /1,1,1
    25   2  28  27  30  14  33  32 ** /2,1,1
    26  27  29   5  31  32  34  17 ** /1,2,1
    27  28   6  29  32  33  18  34 ** /2,2,1
```

The acceptable range of values for corner points are:

	SI **m**	**Field** **ft**
min	-1.0E+20	-3.28E+20
max	1.0E+20	3.28E+20

Convert Corner Point Grid to Node Based Grid (Conditional)

***CRNCON_NODE**

PURPOSE:

*CRNCON_NODE signals a request for internal conversion of corner point grid to node based grid.

DEFAULTS:

Conditional keyword. Default is not to convert corner point grid in the data file to the node-based grid.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group and should only be used with corner point grids (*GRID *CORNER).

EXPLANATION:

Certain grid-processing tasks like generation of inter-block connections are performed more efficiently with the node based grids instead of corner point grids. This keyword facilitates conversion of the corner point grid internally into a node based grid. Considerable saving in grid-processing time can be achieved specially in models with a large number of grid-blocks.

However, there is an overhead in terms of additional memory requirement, since both corner point and node based arrays need to be retained during the simulation.

A command line option '-crncon_node' is available as an alternate to keyword *CRNCON_NODE.

Corner Point Depths for Corner Point Grids (Conditional)

***ZCORN**

PURPOSE:

*ZCORN signals input of an array of corner point depths for corner point grids.

ARRAY:

*ZCORN

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group and should only be used with corner point grids (*GRID *CORNER). This keyword should be combined with *DI and *DJ, or with *COORD, or with *XCORN and *YCORN, to define all the corner point locations for a corner point grid.

The *RG qualifier can be used with this keyword.

EXPLANATION:

See the general corner point discussion given with the *GRID *CORNER keyword for discussions of the notation used here. The unit is (m | ft).

See section **Local Grid Refinement** for keyword *CORNERS.

The *ZCORN keyword causes the reading of all depths ("z"-coordinates) of the 8 * ni * nj * nk corner points required to define the grid. The depths should be input as described in the following algorithm.

Algorithm for *ZCORN Ordering:

Operations should be done in the order shown. Note that the text before each task ([...]) describes how often to carry out that task.

```
Do the following for K = 1, ..., nk: [
        Do the following for J = 1, ..., nj: [
                Write NW-T and NE-T depths for block ( 1,J,K),
                        ...
                Write NW-T and NE-T depths for block (ni,J,K).
                Write SW-T and SE-T depths for block ( 1,J,K),
                        ...
                Write SW-T and SE-T depths for block (ni,J,K).
                        ]
```

Do the following for J = 1, ..., nj: [
Write NW-B and NE-B depths for block (1,J,K),
...
Write NW-B and NE-B depths for block (ni,J,K).
Write SW-B and SE-B depths for block (1,J,K),
...
Write SW-B and SE-B depths for block (ni,J,K).
]]

This completes the algorithm.

The "x"- and "y"-coordinates of the corner points must be provided by the *DI and *DJ keywords, or by the *COORD keyword, or by the *XCORN and *YCORN keywords. If the *DI and *DJ keywords are used, it will be assumed that corner point NW-T of block (1,1,1) is at "x"-coordinate 0.0 and "y"-coordinate 0.0 with increments to be provided by the *DI and *DJ keywords.

Examples:

Provide corner point depths for a ni = 4, nj = 2, nk = 1 *CORNER grid for a reservoir dipping in the "x"-coordinate direction whose layers are a constant 10 units thick.

```
*ZCORN
   2000  2001   2001  2002   2002  2003   2003  2004
   2000  2001   2001  2002   2002  2003   2003  2004
   2000  2001   2001  2002   2002  2003   2003  2004
   2000  2001   2001  2002   2002  2003   2003  2004
   2010  2011   2011  2012   2012  2013   2013  2014
   2010  2011   2011  2012   2012  2013   2013  2014
   2010  2011   2011  2012   2012  2013   2013  2014
   2010  2011   2011  2012   2012  2013   2013  2014
```

The acceptable range of values for corner points depths is:

	SI **m**	**Field** **ft**
min	-1.0E+20	-3.28E+20
max	1.0E+20	3.28E+20

Lateral Corner Point Locations for Corner Point Grids (Conditional) *XCORN, *YCORN

PURPOSE:

*XCORN signals input of an array of corner point "x"-coordinate locations for corner point grids.

*YCORN signals input of an array of corner point "y"-coordinate locations for corner point grids.

ARRAY:

*XCORN
-or-
*YCORN

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

These keywords must be in the RESERVOIR DESCRIPTION keyword group and should only be used with corner point grids (*GRID *CORNER). Both keywords should appear, and be combined with *ZCORN to define all the corner point locations for a corner point grid.

The *RG qualifier can be used with this keyword.

EXPLANATION:

See the general corner point discussion given for the *GRID *CORNER keyword for discussions of the notation used here. The unit is (m | ft).

See section **Local Grid Refinement** for keyword *CORNERS.

The *XCORN and *YCORN keywords each cause reading of all the (ni+1) * (nj+1) * (nk+1) "x"- and "y" coordinate values required to define the lateral locations of all points defining a *CORNER grid. The values should be input as described in the following algorithm.

Algorithm for *XCORN/*YCORN Ordering:

Operations should be done in the order shown. Note that the text before each task ([...]) describes how often to carry out that task.

```
Do the following for K = 1, ..., (nk + 1):  [
Do the following for J = 1, ..., (nj + 1):  [
Do the following for I = 1, ..., (ni + 1):  [
    I, J, K are less than ni, nj, nk, respectively:
        write the "x"- (or "y"-) coordinate of the NW-T point;
    J is less than nj, K is less than nk, and I = ni:
        write the "x"- (or "y"-) coordinate of the NE-T point;
    I is less than ni, K is less than nk, and J = nj:
        write the "x"- (or "y"-) coordinate of the SW-T point;
```

I is less than ni, J is less than nj, and K = nk:
write the "x"- (or "y"-) coordinate of the NW-B point;
I is less than ni, and J = nj, K = nk:
write the "x"- (or "y"-) coordinate of the SW-B point;
J is less than nj, and I = ni, K = nk:
write the "x"- (or "y"-) coordinate of the NE-B point;
K is less than nk, and I = ni, J = nj:
write the "x"- (or "y"-) coordinate of the SE-T point;
I = ni, J = nj, K = nk:
write the "x"- (or "y"-) coordinate of the SE-B point;
where the choice of "x"- or "y"- is determined by
whether *XCORN or *YCORN is being written.
]]]

This completes the algorithm. Note that I is ranging fastest, and K slowest, in the above; J is intermediate.

Examples:

Provide the *XCORN and *YCORN data for a ni = 4, nj = 2, nk = 1 *CORNER grid. Note that the "x"- direction grid spacing is uniformly 100 units and the "y"-direction grid spacing is uniformly 200 units.

```
*XCORN
  0  100  200  300  400
  0  100  200  300  400
  0  100  200  300  400
  0  100  200  300  400
  0  100  200  300  400
  0  100  200  300  400
*YCORN
  0    0    0    0    0
200  200  200  200  200
400  400  400  400  400
  0    0    0    0    0
200  200  200  200  200
400  400  400  400  400
```

The acceptable range of values for corner point coordinates is:

	SI m	Field ft
min	-1.0E+20	-3.28E+20
max	1.0E+20	3.28E+20

Line-Based Corner Point Locations for Corner Point Grids (Conditional) *COORD

PURPOSE:

*COORD signals input of an array of "x"- and "y"- coordinate corner point location information for corner point grids.

ARRAY:

*COORD

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group and should only be used with corner point grids (*GRID *CORNER). This keyword should be combined with *ZCORN to define all the corner point locations for a corner point grid. *COORD is especially useful when faults are present.

The *RG qualifier can be used with this keyword.

EXPLANATION:

See the general corner point discussion given for the *GRID *CORNER keyword for discussions of the notation used here. The unit is (m | ft).

See section **Local Grid Refinement** for keyword *CORNERS.

The *COORD keyword causes the reading of information defining the "x"- and "y"- coordinate locations for all corner points defining a *CORNER grid. *COORD expects that the corner points lie on lines that are predominantly vertical. There should be (ni+1)*(nj+1) such lines, and since the definition of each line requires the specification of two points, each itself requiring the specification of three coordinates, *COORD expects to read 2 * 3 * (ni + 1) * (nj + 1) values as described in the following algorithm.

Algorithm for *COORD Ordering:

Operations should be done in the order shown. Note that the text before each task ([...]) describes how often to carry out that task.

Do the following for J = 1, ..., (nj + 1): [
Do the following for I = 1, ..., (ni + 1): [
Firstly, ...
If I and J are less than ni and nj respectively, write the "x"-, "y"-, "z"-coordinates of a point that lies on a vertical line through the NW corner of block (I,J,1). This could be the "-B" or "-T" corner, or block (I,J,K)'s corner for any K, as all these points should be collinear.
If I = ni and J is less than nj, write the NE corner. If I is less than ni and J = nj, write the SW corner. If I = ni and J = nj, write the SE corner.

Secondly, ...
If I and J are less than ni and nj respectively, write the "x"-, "y"-, "z"-coordinates of another point that lies on a vertical line through the NW corner of block (I,J,1). This point should differ from the previous one only in its "z"- coordinate.
If I = ni and J is less than nj, write the NE corner. If I is less than ni and J = nj, write the SW corner. If I = ni and J = nj, write the SE corner.
]]

This completes the algorithm.

Note that I is ranging fastest, J slowest in the above.

As *COORD data only provides lines on which corner points must lie, *ZCORN array data is still required to locate the corner points along the lines.

Examples:

Provide *COORD data for a ni = 4, nj = 2, nk = 1 *CORNER grid. Note that the "x"-direction grid spacing is uniformly 100 units and the "y"-direction grid spacing is uniformly 200 units. (This example appears the same regardless of the value for nk.)

```
*COORD
     0    0 0     0    0  1    100    0  0   100    0  1
   200    0 0   200    0  1    300    0  0   300    0  1
   400    0 0   400    0  1

     0  200 0     0  200  1    100  200  0   100  200  1
   200  200 0   200  200  1    300  200  0   300  200  1
   400  200 0   400  200  1

     0  400 0     0  400  1    100  400  0   100  400  1
   200  400 0   200  400  1    300  400  0   300  400  1
   400  400 0   400  400  1
```

The acceptable range of values for corner point coordinates is:

	SI **m**	**Field** **ft**
min	-1.0E+20	-3.28E+20
max	1.0E+20	3.28E+20

Complete Corner Point Locations for Corner Point Grids (Conditional) *CORNERS

PURPOSE:

*CORNERS signals input of a complete array of corner point locations for corner point grids.

ARRAY:

*CORNERS

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group and should only be used with corner point grids (*GRID *CORNER). This keyword should not be combined with any other array-based corner point keywords. This keyword provides a complete array of all coordinate values required for all the corner points.

The *RG qualifier can be used with this keyword.

EXPLANATION:

See the general corner point discussion given with the *GRID *CORNER keyword for discussions of the notation used here.

This keyword causes the processing of 24 values for each block, with the first third of the values giving all the "x"-coordinates of all corner points, the second third giving all the "y"-coordinates, and the final third giving all the "z"-coordinates. Each group uses the same corner point ordering (as presented below), which is also the ordering used by the *ZCORN keyword. Only the choice of coordinate direction changes from group to group. Note that the third group of values is the same array that would be input using the *ZCORN keyword.

Algorithm for *CORNERS Ordering:

Operations should be done in the order shown. Note that the text before each task ([...]) describes how often to carry out that task.

Do the following three times with:
(1) "values" replaced by ""x"-coordinate values";
(2) "values" replaced by ""y"-coordinate values";
(3) "values" replaced by ""z"-coordinate values", the latter also being the depths:
[

```
Do the following for K = 1, ..., nk: [
    Do the following for J = 1, ..., nj: [
        Write NW-T and NE-T values for block ( 1,J,K)
          ...
        Write NW-T and NE-T values for block (ni,J,K)
        Write SW-T and SE-T values for block ( 1,J,K)
          ...
        Write SW-T and SE-T values for block (ni,J,K)
] ]
    Do the following for J = 1, ..., nj: [
        Write NW-B and NE-B values for block ( 1,J,K)
          ...
        Write NW-B and NE-B values for block (ni,J,K)
        Write SW-B and SE-B values for block ( 1,J,K)
          ...
        Write SW-B and SE-B values for block (ni,J,K)
] ] ]
```

This completes the algorithm.

This technique for corner point input will exhibit duplication in the first two groups of 8*ni*nj*nk values, due to the fact that corner points must lie on vertical lines.

Examples:

Provide *CORNERS data for a ni = 4, nj = 2, nk = 1 grid in a reservoir dipping in the "x"-coordinate direction. Note that the single layer is 10 units thick and that the "x"-direction grid spacing is 100 units and the "y"-direction grid spacing is 200 units.

```
*CORNERS
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400
    0   100   100   200   200   300   300   400

    0     0     0     0     0     0     0     0
  200   200   200   200   200   200   200   200
  200   200   200   200   200   200   200   200
  400   400   400   400   400   400   400   400
    0     0     0     0     0     0     0     0
  200   200   200   200   200   200   200   200
  200   200   200   200   200   200   200   200
  400   400   400   400   400   400   400   400

 2000  2001  2001  2002  2002  2003  2003  2004
 2000  2001  2001  2002  2002  2003  2003  2004
 2000  2001  2001  2002  2002  2003  2003  2004
 2000  2001  2001  2002  2002  2003  2003  2004
 2010  2011  2011  2012  2012  2013  2013  2014
 2010  2011  2011  2012  2012  2013  2013  2014
 2010  2011  2011  2012  2012  2013  2013  2014
 2010  2011  2011  2012  2012  2013  2013  2014
```

The acceptable range of values for corner points are:

	SI m	Field ft
min	-1.0E+20	-3.28E+20
max	1.0E+20	3.28E+20

Local Grid Refinement

Keyword *REFINE enables a locally refined grid (LGR) by specifying the parent block and its division into child blocks. By default the child blocks are of uniform *average* size – average since those block shapes may vary. To over-ride the default child-block geometries, use one of these two methods (different LGRs may use different methods):

1. Use sub-keyword *RG with *DI, *DJ, *DK to specify the ratios of child block sizes. In this case the child-block corners are positioned to give the targeted ratios for *average* block size and completely fill the parent block. See section **Local Grid Refinement** for keyword *DI.
2. Use sub-keyword *RG with a keyword that specifies corner points. The corner-point keyword may be different from that used to specify the fundamental grid.

Local Refined Grid (Conditional) *REFINE, *RANGE

PURPOSE:

*REFINE indicates the input of local refined grid.

FORMAT:

*REFINE	*block_address* *INTO *nir njr nkr*
*REFINE	*block_address* *INTO *nr ntheta nz* *HYBRID
	(*IDIR | *JDIR | *KDIR) *RW *rw*
	(*ALPHAI *alphai*) (*ALPHA *alpha*)
*REFINE	*INNERWIDTH W_{inner}
	block_address *INTO n_{ir} n_{jr} n_{kr}
	(*WF2 W_{f2}) (*K2INT k_{2int})
-or-	
*REFINE	*nir njr nkr*
*REFINE	*HYBRID *nr ntheta nz* (*IDIR | *JDIR | *KDIR) *RW *rw*
	(*ALPHAI *alphai*) (*ALPHA *alpha*)
*RANGE	*block_address*

DEFINITIONS:

*REFINE

Indicates the application of local grid refinement of the specified type and parameters to the specified block or range of blocks.

block_address

The address of the grid block(s) to which this refinement applies. Three forms are allowed:

1. Multi-level single-block UBA (see **Multi-level Regular Refinement**, below),
2. Single-level range *i1*(:*i2*) *j1*(:*j2*) *k1*(:*k2*), and
3. *BG; see **Assign Data to Block Groups** in the "Keyword Data Entry System" chapter.

*INTO

Indicates as new refinement. This must be present at the first appearance of *REFINE. Subsequent usage of *REFINE without *INTO causes the same refinement to be used.

nir

Number of refined blocks in the I direction within each fundamental grid block.

njr

Number of refined blocks in the J direction within each fundamental grid block.

nkr

Number of refined blocks in the K direction within each fundamental grid block.

nr

Number of radial subdivisions in the R-theta-Z local hybrid grid. Allowed values for *nr* are 2,3,4,... up to a maximum of 10.

ntheta

Number of theta subdivisions in the R-theta-Z local hybrid grid. Permitted values are 1 or 4. Theta subdivisions are not applied to the inner- most hybrid grid block.

nz

Number of Z-direction subdivisions in the R-theta-Z local hybrid grid. The "Z" direction of the hybrid grid is specified using the *IDIR, *JDIR, or *KDIR keywords. You are allowed to divide a fundamental grid into a maximum of 4 refined grids. Permitted values are 1,2,3 and 4.

*HYBRID

Indicates the use of hybrid grid refinement in which a Cartesian grid block (normally containing a well) is refined into a local cylindrical R-theta-Z grid.

*IDIR

Indicates that the "Z" axis of the hybrid grid is parallel to the I-direction of the fundamental grid.

*JDIR

Indicates that the "Z" axis of the hybrid grid is parallel to the J-direction of the fundamental grid.

*KDIR

Indicates that the "Z" axis of the hybrid grid is parallel to the K-direction of the fundamental grid. This is the default.

*RW *rw*

Define well radius (m | ft), which is also the inner radius of the innermost ring. The value must be greater than zero and must not exceed 1 m (3.28 ft). The volume inside this radius will be removed from the block. *RW is required with *HYBRID.

If a discretized wellbore is embedded in a hybrid grid, the wellbore radius from the *WELLBORE keyword will be used and this radius will be ignored.

*ALPHAI *alphai*

Define the ratio (R_1/R_0) of outer radius to inner radius for the first (innermost) ring. The inner radius is $R_0 = rw$, so the outer radius will be $R_1 = rw \cdot alphai$. Used only for isotropic *HYBRID cases. The value of *alphai* must exceed 1.

*ALPHA *alpha*

Define the ratio (R_i/R_{i-1}) of outer radius to inner radius for the remaining rings, i = 2 to *nir*-1, that is, $R_i = alpha \cdot R_{i-1}$. Used only for isotropic *HYBRID cases. The value of *alpha* must exceed 1.

*INNERWIDTH W_{inner}

Specify width W_{inner} (m | ft) of the inner-most planes of blocks for a complex fracture grid. These blocks are tagged as the fracture zone and can be referenced separately via grid-array qualifier *FZ or skipped via *NFZ. Outside of the fracture zone, block sizes normal to the fracture-zone plane increase logarithmically away from the fracture zone. The value of W_{inner} does not refer to the actual (intrinsic) width of the fracture (which is on the order of 0.001 m) but to the width of a fracture conduit on which an effective fracture permeability is calculated so as to maintain the conductivity of the original fracture. Normally values on the order of 2/3m to 1 m are used. The default is 2 feet or 0.6096 meters.

*WF2 W_{f2}

Specify secondary fracture width W_{f2} (m | ft). No default.

*K2INT k_{2int}

Specify secondary intrinsic permeability k_{2int} (md). No Default.

*WF2 and *K2INT re-define the intrinsic fracture width and fracture permeability of the secondary fracture network (natural fractures) within the SRV or planar fracture region.

W_{f2} and k_{2int} along with fracture spacing (*DIFRAC, DJFRAC,*DKFRAC) are used to model the effective permeability enhancement of the naturally fractured network within the SRV or planar fracture region caused by the hydraulic fractures. If *WF2 and K2INT are not present, the effective natural fracture permeability in the SRV or planar fracture region is read from the input permeability arrays (*PERMI *FRACTURE, *PERMJ *FRACTURE, *PERMK *FRACTURE).

*WF2 and *K2INT must be defined together. The use of one of these keywords without the other is an error.

DEFAULTS:

If *REFINE is absent, there are no locally refined grids.

If *REFINE is present but *INTO is absent, the immediately previous *INTO data is used; the first *REFINE must have *INTO.

There are no defaults for *nir*, *njr* and *nkr*. For *HYBRID there are no defaults for *nr*, *ntheta*, *nz* and *rw*. For hybrid grid, the inner radial block never has theta subdivisions.

When *HYBRID is used and none of *IDIR, *JDIR and *KDIR are specified, *KDIR is assumed.

If *ALPHAI is absent, $alphai = R_{parent}/[rw\cdot(nir+1)]$, where R_{parent} is the radius of the circle with the same area as the parent cell's face normal to the hybrid grid's axial direction. The outer radius of the first ring is $R_1 = rw \cdot alphai = R_{parent}/(nir+1)$.

If *ALPHA is absent, $alpha = [R_{parent}/ R_1]^{1/(nir-1)}$ where R_1 is the outer radius of the first (innermost) ring and R_{parent} is defined immediately above. This formula is based on the assumption that hypothetical radius $alpha\cdot R_{nir-1} = R_{parent}$. In reality, the outermost ring is shaped to fit the parent cell's rectangular perimeter.

For anisotropic media, the values for *ALPHAI and *ALPHA are calculated internally using much the same criteria as discussed above for the isotropic case, except that elliptical geometries are used.

CONDITIONS:

For regular refinement, there is no internal limit to the magnitudes of *nir*, *njr* and *nkr*. However, values larger than 3-5 tend to produce numerically inconsistent results at the interface between coarse and fine blocks.

Hybrid grid refinements can be used only with Cartesian grids, i.e. *GRID *CART.

*REFINE may not be used together with *NINEPOINT.

When defining a discretized wellbore inside a hybrid grid, the hybrid grid must be defined first. See detailed explanation for *WELLBORE.

The only types of multi-level refinement allowed are (1) Cartesian regular refinement, (2) Cartesian regular refinement with hybrid grid at the finest level, and (3) discretized wellbore in hybrid grid. At least 10 levels of refinement are available.

Areas with different degrees of refinement must be separated by at least one unrefined grid block. See the detailed descriptions below.

The *REFINE keyword can be used with *GRID *CART, *GRID *VARI or *GRID *CORNER but not with *GRID *RADIAL.

Pinched out (and null) cells can be marked as refined in data, without terminating the simulation, although these cells will remain inactive.

Local grid refinement may be used with natural fracture options *DUALPOR and *DUALPERM but not *MINC or *SUBDOMAIN.

EXPLANATION:

*REFINE may occur several times to define multiple regions or refinement types.

By default, refined grid blocks are assigned the properties of the fundamental block in which they reside. Alternatively the properties of the refined grid may be entered by using the *RG array qualifier keyword with any array keyword.

The variable depth/variable thickness option may be used with refined grids.

See Appendices E.6 and E.7 for further discussion.

REGULAR REFINEMENT

The parent block is refined into a child grid that is of the same type and orientation as the parent block. In each refined direction the refined block sizes are uniform. The I-J-K indices in the local grid follow the same sense as the parent block, but the local origin starts at the corner of the parent block closest to the global origin.

One rule applies to adjacent parent blocks with regular refinement: in each refined direction, parent blocks adjacent in that direction must be refined by the same amount normal to that direction. For example, if block (I,J,K) has been refined into *nir* x *njr* x *nkr*, then

- blocks (I-1,J,K) and (I+1,J,K) must be refined with the same *njr* and *nkr*, if at all;

- blocks (I,J-1,K) and (I,J+1,K) must be refined with the same *nir* and *nkr*, if at all; and
- blocks (I,J,K-1) and (I,J,K+1) must be refined with the same *nir* and *njr*, if at all

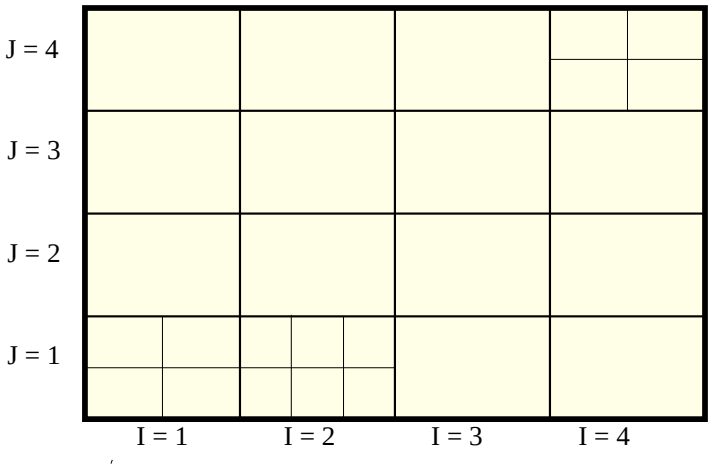
Example:

A grid system consists of 4 blocks in the I direction, 4 blocks in the J direction and 2 blocks in the K direction. Two columns of grid blocks are to be refined, with two refined blocks in each direction. The data file is as follows:

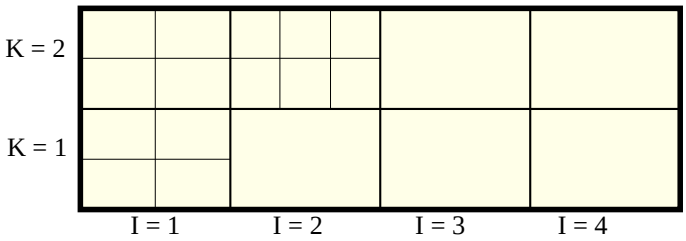
```
*GRID *CART 4 4 2
. . .
*REFINE 1 1 1:2    *INTO 2 2 2
*REFINE 4 4 1:2
*REFINE 2 1 2      *INTO 3 2 2
```

Note that two regions were assigned the same refinement type 2 x 2 x 2. Also, blocks (1,1,1) and (2,1,1) are adjacent in the I direction and so must have the same J- and K-direction refinement, but may have a different *nir*.

An areal view of the grid for K=2 would be:



A cross-section for J=1 would be:



Multi-level Regular Refinement

Regular local grid refinement may extend to more than one level. For example, this data fragment specifies 5 levels of 3x3 refinement in a single fundamental block.

```
*refine 5 3 2 *into 3 3 3
*refine 5 3 2 / 2 2 *into 3 3 3
*refine 5 3 2 / 2 2 2 / 2 2 2 *into 3 3 3
*refine 5 3 2 / 2 2 2 / 2 2 2 / 2 2 2 *into 3 3 3
*refine 5 3 2 / 2 2 2 / 2 2 2 / 2 2 2 / 2 2 2 *into 3 3 3
```

Note that no range is allowed in the parent *block_address* when it is refined, that is, has a slash in the UBA. Be aware that excessive use of multi-level refinement can increase the number of grid blocks significantly.

HYBRID REFINEMENT

The hybrid grid option refers to refining a parent block from a Cartesian grid into a local cylindrical grid whose "axial" direction may be oriented in either the global I, J or K direction. There are *nr* divisions in the radial direction, of which the outermost is formed to fit the shape of the parent block. The hybrid's angular direction is divided into either 1 or 4 divisions; the innermost radial division is always a full circle. The hybrid's axial direction is divided into *nz* uniform sections. It is anticipated, but not required, that a well will be placed inside the innermost radial division.

The *HYBRID option may affect results especially when a process is influenced strongly by near-wellbore phenomena, e.g., cyclic steam stimulation. The well can be horizontal or vertical. The wellbore and the corresponding hybrid grid axis must go through the centre of the grid block. Thus, the hybrid grid's local "Z-axis" may be in the global I ,J or K direction depending on the well direction.

Perpendicular to this axis the permeability may be equal (isotropic case) or not equal (anisotropic case). The aspect ratio of grid dimensions normal to the axial direction should not be too different from the square root of the corresponding absolute permeability ratio. Deviations of more than a factor of 1.25 can lead to large errors and so is not allowed.

ISOTROPIC CASE:

This is normally the case for a hybrid grid whose axis is in the vertical direction. The grid dimensions must be within a factor of 1.25 of square.

ANISOTROPIC CASE:

This is normally the case for a hybrid grid whose axis is in a horizontal direction. The aspect ratio of block size normal to the axial direction should be within a factor of 1.5 to 2 of the square root of the ratio of the corresponding absolute permeabilities. For example, a well horizontal in the X-direction with Ky = 10 Kz should have a grid aspect ratio of about delta_Y/delta_Z = square_root(10).

Two rules apply to adjacent parent blocks refined with *HYBRID:

1. For hybrid grids adjacent in the hybrid's axial direction, *nr* and *ntheta* must be the same, and *nz* may be different. This case is typical for modelling a hybrid grid around a well that passes through more than one parent block.
2. For hybrid grids adjacent in a direction other than the hybrid's axial direction, only *nz* must be the same.

Hybrid Grid Orientations

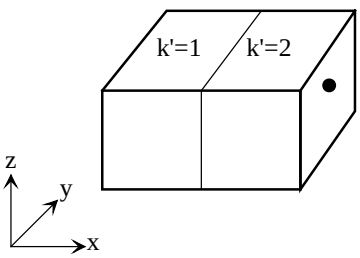
Normally, the user will need to know only which of *IDIR, *JDIR or *KDIR was specified in order to interpret the position of the individual hybrid grid blocks relative to the surrounding fundamental blocks. However, the precise meaning of the hybrid grid's local J and K indices is needed in order to assign non-uniform properties and conditions, and to interpret in detail the textual output.

In the following, x, y, z, I, J and K refer to the fundamental grid and similar primed (') symbols refer to the local cylindrical grid.

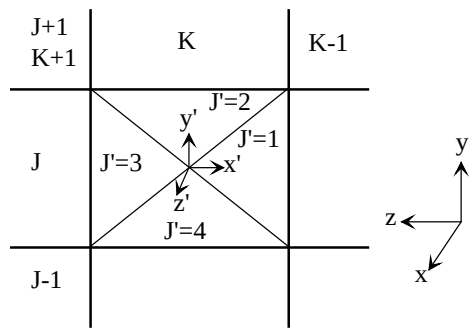
In each of the following orientation cases, both a "GLOBAL VIEW" and a "LOCAL VIEW" are shown. In the "LOCAL VIEW" the point-of-view is on the hybrid z' axis looking in the negative z' direction, i.e., z' points toward the viewer. Note that the only difference between the "LOCAL VIEW" of the cases is the relation to the global coordinates.

Well in I Direction (*IDIR): x = xo + z', y = yo + y', z = zo – x'

GLOBAL VIEW



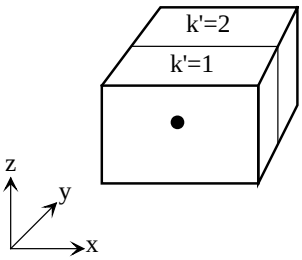
LOCAL VIEW



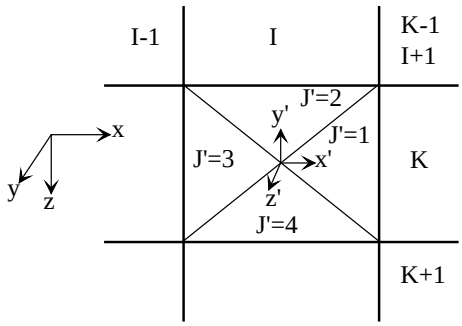
Adjacent block is connected to	(I,J,K-1) J'=1	(I,J+1,K) J'=2	(I,J,K+1) J'=3	(I,J-1,K) J'=4

Well in J Direction (*JDIR): x = xo + x', y = yo + z', z = zo – y'

GLOBAL VIEW

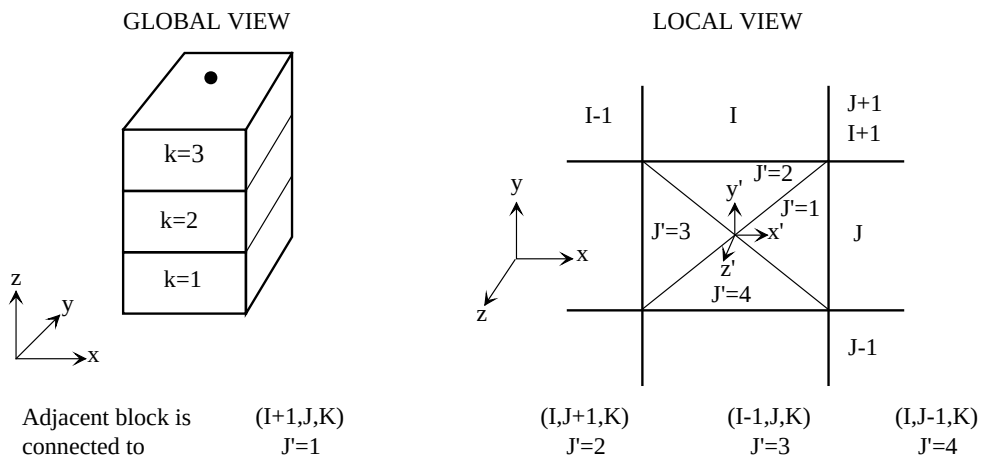


LOCAL VIEW



Adjacent block is connected to	(I+1,J,K) J'=1	(I,J,K-1) J'=2	(I-1,J,K) J'=3	(I,J,K+1) J'=4

Well in K Direction (*KDIR): x = xo + x', y = yo + y', z = zo + z'



Direction Dependent Data

The specification of direction-dependent data differs slightly from fundamental or regular refined grids. In the data entry keywords there is no explicit way to refer to a hybrid grid's local directions. For example, for entering permeability there are PERMI, PERMJ and PERMK but nothing explicitly for R, Theta and Z. The method used to refer to the hybrid grid direction is as follows.

For each orientation described above, there is correspondence between the I,J,K direction labels and the hybrid grid's local radial, angular and axial directions.

Orientation	Radial	Angular	Axial
*IDIR	K	J	I
*JDIR	I	K	J
*KDIR	I	J	K

Take the *IDIR orientation for example. You would use PERMK to modify permeability in the hybrid grid's local radial direction, PERMJ to modify the angular direction and PERMI to modify the axial direction. You would examine K Direction Block Size to find the block size in the hybrid grid's local radial direction, J Direction to find the angular size and I Direction to find the axial size.

This correspondence is reported in the textual output file in the grid summary section, for each hybrid grid. It applies to all direction dependent input (except block size)

- permeabilities
- transmissibility multipliers (constant and pressure-dependent)
- block area modifiers
- dispersion

and output

- block sizes
- permeabilities
- transmissibility multipliers (constant and pressure-dependent)

- transmissibilities
- conduction geometry factors
- block area modifiers
- dispersion.

Note that connection-based quantities such as transmissibility have an explicit Radial and Angular direction printout, but the Axial direction values are found through this direction correspondence.

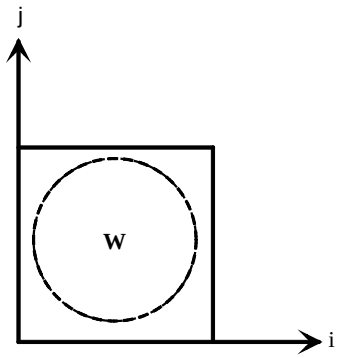
Block Sizes

The block sizes reported for the hybrid grid blocks are similar to those found for a cylindrical grid system. The radial and axial block sizes have standard definitions, except for the outermost radial block which acts as an interface between the radial grid and the surrounding Cartesian grid. The angular block size is the parent block's size in the corresponding direction instead of the midpoint arc length, and the radial block size is half the parent block size minus the last radius. For all hybrid grid cells, the product of block sizes in the three directions does not necessarily equal the block volume without volume modifier.

For the case with no angular subdivisions (*ntheta* = 1) the radial block size is based not on the entire outermost block volume, but the fraction associated with the outer face of interest. This fraction of the total volume is the same as the block's volume for the *ntheta* = 4 case.

Example: Hybrid refinement where *nr* = 2 and *ntheta* = 1

*REFINE 1 1 1 *INTO 2 1 1 *HYBRID *KDIR



Example: *nr* = 3 and *ntheta* = 4, with axial direction in the X-direction

*REFINE 1 1 1 *INTO 3 4 1 *HYBRID *IDIR

Example: Hybrid Grid Cell Geometry

Template data set “verify09.dat” has the following data:

```
*GRID *CART 5 5 6
*DI *CON 131.24
*DJ *CON 131.24
*DK *KVAR 41.5  6.561 2*19.683 2*6.561
*REFINE 3 3 3 *INTO 4 4 1
*HYBRID *RW 0.3 *KDIR
*POR *CON 0.35
```

Block sizes of parent cell (3,3,3) are $d_i = d_j = 131.24$ and $d_k = 19.683$. The hybrid grid is of isotropic type with *nir* = 4 radial divisions, 4 angular divisions and *rw* = 0.3. The hybrid grid axis is in the K direction, so axial block size is $L_z = d_k$. Let R_{i-1} and R_i be the inner and outer radii of radial block i, for 1=1 to *nir*, so $R_0 = rw = 0.3$. Let R_{parent} be the radius of a circle with the same area as the parent block normal to the axial direction.

$$R_{parent} = \sqrt{(d_i \cdot d_j/\pi)} = 74.044.$$

Since *ALPHAI is absent,

$$R_1 = R_{parent}/(nir+1) = 14.809.$$

Since *ALPHA is absent,

$$alpha = [R_{parent}/ R_1]^{1/(nir-1)} = 1.71, \text{ so}$$

$$R_2 = alpha \cdot R_1 = 25.323,$$

$$R_3 = alpha \cdot R_2 = 43.301 \text{ and}$$

hypothetical radius $alpha \cdot R_3 = 74.044 = R_{parent}$.

The remaining geometrical quantities are based on these facts: (1) innermost ring is a full circle, (2) remaining rings are divided into four sectors (of equal angle since $d_i = d_j$), and (3) outermost ring fits inside parent cell boundary.

The bulk volumes V_b and pore volumes $V_p = V_b \cdot 0.35$ are

$$V_{b1} = ½ \cdot (2\pi) \cdot (R_1^2 - R_0^2) \cdot L_z = 13555, V_{p1} = 4744,$$

$$V_{b2} = ½ \cdot (\pi/2) \cdot (R_2^2 – R_1^2) \cdot L_z = 6522.8, V_{p2} =2283,$$

$$V_{b3} = ½ \cdot (\pi/2) \cdot (R_3^2 – R_2^2) \cdot L_z = 19073, V_{p3} =6676 \text{ and}$$

$$V_{b4} = [¼ \cdot d_i \cdot d_j - ½ \cdot (\pi/2) \cdot R_3^2) \cdot L_z = 55769, V_{p4} =19519.$$

The radial block sizes L_R are

$$L_{R1} = R_1 – R_0 = 14.509,$$

$$L_{R2} = R_2 – R_1 = 10.514,$$

$$L_{R3} = R_3 – R_2 = 17.979 \text{ and}$$

$$L_{R4} = ½(131.24) – R_3 =22.319.$$

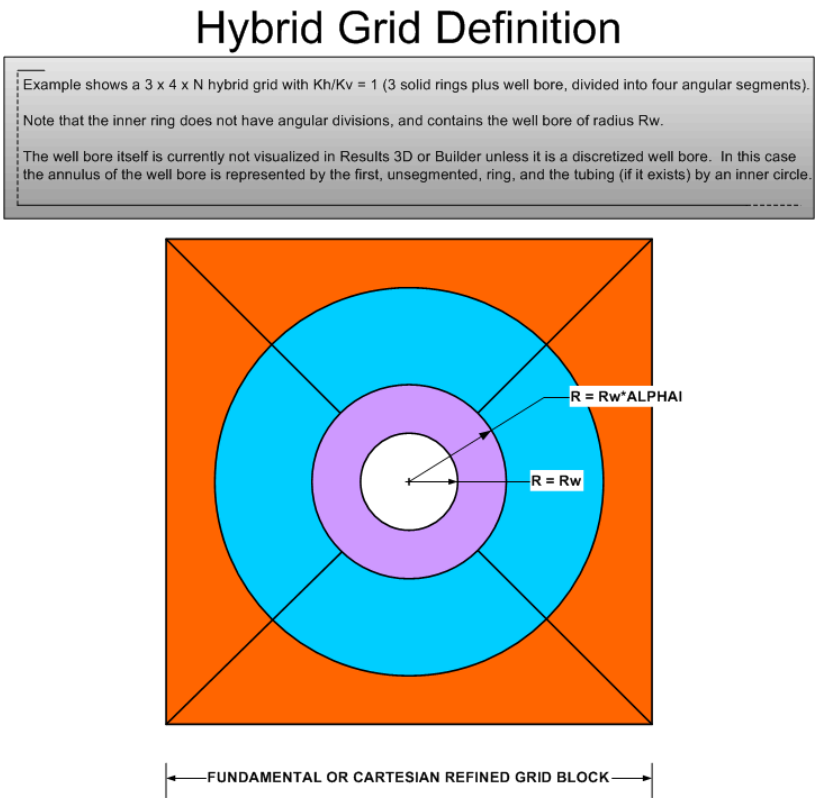
Use function $f(\alpha) = \exp[\ln(\alpha) \cdot \alpha^2/(\alpha^2-1) -½]$ to obtain angular block lengths L_θ

$$L_{\theta 1} = (2\pi) \cdot f(R_1/R_0) \cdot R_0 = 56.526,$$

$$L_{\theta 2} = (\pi/2) \cdot f(R_2/R_1) \cdot R_1 = 31.884,$$

$$L_{\theta 3} = (\pi/2) \cdot f(R_3/R_2) \cdot R_2 = 54.523 \text{ and}$$

$$L_{\theta 4} = d_i \text{ or } d_j = 131.24.$$



Plotting in Results 3D

In Results 3D a hybrid grid is represented by rectangular blocks designed for optimal visibility. It does not employ the hybrid-grid block sizes used by the simulator.

Changing Grid Refinement Later in the Run

Keyword *REFINE may appear also in a segment of recurrent data, in which case the refinement is applied at the immediately preceding *TIME or *DATE keyword.

Of the options related to *REFINE, those available depend on whether or not keyword ***DYNAGRID** appears anywhere in the (recurrent) data.

1. *DYNAGRID absent: Any refinement specified by *REFINE is present in the grid for the remainder of the run. Natural fracture options *DUALPOR and *DUALPERM are allowed.
2. *DYNAGRID present: Sub-keywords may remove (*DEREFINE) or modify (*REREFINE) any refinement specified by *REFINE. Natural fracture options are not allowed.

Block Group Example

Specify a small rectangular refinement area.

```
** Small rectangular refinement area
*REFINE 12:18 23:27 4:8 *INTO 3 3 3
```

Specify two large irregularly shaped refinement areas. The block groups can be used to assign properties, too.

```
*BLOCKGROUP 'Area A' *IJK ...  ** 650 members
*REFINE *BG 'Area A' *INTO 3 3 3
*BLOCKGROUP 'Area B' *IJK ...  ** 420 members
*REFINE *BG 'Area B' *INTO 3 3 3
```

Complex Fracture Example

Specify a small rectangular complex-fracture area.

```
** Small rectangular fractured area
*REFINE *INNERWIDTH 1.8
   12:18 23:27 4:8 *INTO 9 9 3
```

Specify two large irregularly shaped complex-fracture areas. The block groups can be used to assign properties, too.

```
*BLOCKGROUP 'Area A' *IJK ...  ** 650 members
*REFINE *INNERWIDTH 1.8 *WF2 1.0 *K2INT 1000
   *BG 'Area A' *INTO 9 9 3
*BLOCKGROUP 'Area B' *IJK ...  ** 420 members
*REFINE *INNERWIDTH 1.3 *WF2 1.2 *K2INT 1500
   *BG 'Area B' *INTO 7 7 3
```

Define Planar Fracture Local Grid (Conditional) *PLNR_REFINE, *MDPLNRBK

PURPOSE:

Create a local grid tailored toward a planar hydraulic fracture.

FORMAT:

*PLNR_REFINE *fracture_origin* *INTO n_{ir} n_{jr} n_{kr}
*BWHLEN L_{half} (*IDIR | *JDIR)
*INNERWIDTH W_{inner}
(*WF2 W_{f2}) (*K2INT k_{2int})
(*LAYERSUP $nlay_{up}$)
(*LAYERSDOWN $nlay_{down}$)
(*SBG_FZT '*sbg_fzt*')
(*FRAC_CON) (*SBG_FZL '*sbg_fzl*')
*BG_NAME '*bg_name*'
*MDPLNRBK *fund_blocks*

DEFINITIONS:

*PLNR_REFINE

Construct a local grid for planar (also known as bi-wing) fracture with the specified parameters. Local grid refinement is done in as many parent blocks as needed to contain the fracture. All the resulting refined blocks are gathered together into block group '*bg_name*' which can be used to reference all these refined blocks. In addition, the transverse fracture zone is a single plane of blocks that extends from the centre outward to the fracture tips. Grid-array properties can be assigned to these fracture zone via *BG '*bg_name*' *FZ or *SBG_FZT; blocks outside this fracture zone can be assigned via *BG '*bg_name*' *NFZ.

fracture_origin

User block address of the origin of the planar hydraulic fracture, where a well perforation connects to the fracture. This cell must be a member of the fundamental grid.

*INTO n_{ir} n_{jr} n_{kr}

Number of fine blocks in the I, J and K directions, for each fundamental cell containing the fracture.

*BWHLEN L_{half}

Specify half-length L_{half} (m | ft) of a planar hydraulic fracture. The fracture refinement extends this distance from *fracture_origin* in the specified direction until the half-length is completely contained, possibly into neighbouring cells.

(*IDIR | *JDIR)

Specify direction in which the planar hydraulic fracture is progagated from *fracture_origin*. The direction keyword is required.

*INNERWIDTH W_{inner}

Specify width W_{inner} (m | ft) of the inner-most planes of blocks for a complex fracture grid. These blocks are tagged as the fracture zone and can be referenced separately via grid-array qualifier *FZ or skipped via *NFZ. Outside of the fracture zone, block sizes normal to the fracture-zone plane increase logarithmically away from the fracture zone. The value of W_{inner} does not refer to the actual (intrinsic) width of the fracture (which is on the order of 0.001 m) but to the width of a fracture conduit on which an effective fracture permeability is calculated so as to maintain the conductivity of the original fracture. Normally values on the order of 2/3m to 1 m are used. The default is 2 ft (0.6096 m).

*WF2 W_{f2}

Specify secondary fracture width W_{f2} (m | ft).

*K2INT k_{2int}

Specify secondary intrinsic permeability k_{2int} (md).

*WF2 and *K2INT re-define the intrinsic fracture width and fracture permeability of the secondary fracture network (natural fractures) within the SRV or planar fracture region.

W_{f2} and k_{2int} along with fracture spacing (*DIFRAC, DJFRAC,*DKFRAC) are used to model the effective permeability enhancement of the naturally fractured network within the SRV or planar fracture region caused by the hydraulic fractures. If *WF2 and K2INT are not present, the effective natural fracture permeability in the SRV or planar fracture region is read from the input permeability arrays (*PERMI *FRACTURE, *PERMJ *FRACTURE, *PERMK *FRACTURE).

*WF2 and *K2INT must be defined together. The use of one of these keywords without the other is an error.

*LAYERSUP $nlay_{up}$

Specify the number of K layers to include above *fracture_origin*. The default is 0. Use this keyword only when *fracture_origin* corresponds to the completion of a horizontal well.

*LAYERSDOWN $nlay_{down}$

Specify the number of K layers to include below *fracture_origin*. The default is 0. Use this keyword only when *fracture_origin* corresponds to the completion of a horizontal well.

*SBG_FZT '*sbg_fzt*'

Specify the name of the structured block group that is identified with the transverse fracture zone blocks (see *PLNR_REFINE, above). Block group '*sbg_fzt*' is useful for specifying grid-array properties on a per-cell or I-J-K basis, which is not possible with *BG '*bg_name*' *FZ. If *FRAC_CON is

used, block group '*sbg_fz*" lets you distinguish the transverse from the longitudinal fracture zone.

*FRAC_CON

Extend the fracture zone from the fracture center to the edge of the refinements in the direction *perpendicular* to the fracture plane (along the well direction), over the entire fracture height. The resulting extension is called the longitudinal fracture zone. Grid array modifier *FZ will refer to both the transverse and longitudinal fracture zones.

*SBG_FZL '*sbg_fzl*'

Specify the name of the structured block group that is identified with the longitudinal fracture zone blocks created by *FRAC_CON. Block group '*sbg_fzl*' is useful for specifying grid-array properties in the longitudinal fraction zone on a per-cell or I-J-K basis, which is not possible with *BG '*bg_name*' *FZ. Block group '*sbg_fzl*' lets you distinguish the longitudinal from the transverse fracture zone. *SBG_FZL is valid only when *FRAC_CON is used. The transverse and longitudinal fracture zones will share the centre column of cells; if those cells are assigned values by both *SBG_FZT and *SBG_FZL, the last assignment overwrites any previous assignment.

*BG_NAME '*bg_name*'

Specify the name of the planar fracture whose members make up all the refined blocks created for this planar hydraulic fracture.

*MDPLNRBK *fund_blocks*

Specify the maximum number of fundamental grid blocks that each planar fracture may fully or partially cover. The default is 10. This is a primary keyword and should appear at most once. Builder will supply this keyword automatically. This dimensioning information must be supplied explicitly since the final actual number depends upon fracture length relative to local block sizes which are not known at storage allocation time.

DEFAULTS:

If *FRAC_CON is absent, there is no longitudinal fracture zone.

If *MDPLNRBK is absent then *fund_blocks* = 10.

CONDITIONS:

Fracture origin *fracture_origin* must be a cell in the fundamental grid.

Blockgroup '*bg_name*' must not have been defined previously.

The combination of the location of *fracture_origin*, the sizes of it and its neighbours and *BWHLEN must not cause the planar fracture to extend (1) beyond the boundary of the fundamental grid or (2) into an existing hydraulic fracture.

If *LAYERSUP is absent then $nlay_{up}$ = 0 is assumed.

If *LAYERSDOWN is absent then $nlay_{down}$ = 0 is assumed.

The number of fundamental grid blocks covered by any one planar fracture may not exceed *fund_blocks*.

*SBG_FZL is allowed only if *FRAC_CON is used.

EXPLANATION:

Example:

Create and assign permeability to a planar fracture with only transverse fracture zone.

```
*PLNR_REFINE 12 23 4 *INTO 9 9 3
   *BWHLEN 160  ** Fracture half-length, ft
   *IDIR
   *INNERWIDTH 1.8
   *WF2 1.0  *K2INT 1000
   *SBG_FZT 'W3_Stg8'
   *BG_NAME 'Well 3, Stg 8'

*MDPLNRBK 20
*PERMI *MATRIX   *BG 'Well 3, Stg 8' *NFZ 1270
*PERMI *FRACTURE *BG 'Well 3, Stg 8' *NFZ 3740
*PERMI *MATRIX   *BG 'Well 3, Stg 8' *FZ 1380 1290
*PERMI *FRACTURE *SBG 'W3_Stg8' *FZ *ALL ...
*PERMJ *FRACTURE *SBG 'W3_Stg8' *FZ *IDIR ...
*PERMJ . . .
```

Create and assign permeability to a planar fracture with transverse and longitudinal fracture zones. Note that the assignment for longitudinal fracture zone overwrites the assignment for transverse fracture zone at the centre where the zones overlap.

```
*PLNR_REFINE 12 23 4 *INTO 9 9 3
   *BWHLEN 160  ** Fracture half-length, ft
   *IDIR
   *INNERWIDTH 1.8
   *WF2 1.0  *K2INT 1000
   *SBG_FZT 'W3_Stg8_TRVS'
   *FRAC_CON
   *SBG_FZL 'W3_Stg8_LNGT'
   *BG_NAME 'Well 3, Stg 8'

*MDPLNRBK 20
*PERMI *MATRIX   *BG 'Well 3, Stg 8' *NFZ 1270
*PERMI *FRACTURE *BG 'Well 3, Stg 8' *NFZ 3740
*PERMI *MATRIX   *BG 'Well 3, Stg 8' *FZ 1380 1290
*PERMI *FRACTURE *SBG 'W3_Stg8_TRVS' *FZ *ALL ...
*PERMJ *FRACTURE *SBG 'W3_Stg8_TRVS' *FZ *IDIR ...
*PERMI *FRACTURE *SBG 'W3_Stg8_LNGT' *FZ *ALL ...
*PERMJ *FRACTURE *SBG 'W3_Stg8_LNGT' *FZ *IDIR ...
*PERMJ . . .
```

Planar Fracture Template *PLNRFRAC_TEMPLATE, *PLNRFRAC

PURPOSE:

Specify a template of data for the planar model and apply it in multiple locations.

FORMAT:

*PLNRFRAC_TEMPLATE ‘*Template_Name*’
*PLNR_REFINE *INTO n_{ir} n_{jr} n_{kr}
*BWHLEN L_{half}
(*IDIR | *JDIR)
*INNERWIDTH W_{inner}
(*WF2 W_{f2})
(*K2INT k_{2int})
(*LAYERSUP $nlay_{up}$)
(*LAYERSDOWN $nlay_{down}$)
{ *grid_array* (*MATRIX | *FRACTURE) (*FZ | *NFZ) ... }
(*FRAC_CON)
*END_TEMPLATE
*PLNRFRAC ‘*Template_Name*’ *fracture_origin* *BG_NAME ‘*bg_name*’
(*SBG_FZT ‘*sbg_fzt*’) (*SBG_FZL ‘*sbg_fzl*’)

*MDPLNRBK *fund_blocks*

*PLNRFRAC_TEMPLATE ‘*Template_Name*’

. . . *END_TEMPLATE

Define the planar-fracture template with unique name *Template_Name*. Multiple templates are allowed and will differ by their names and possibly the values of their parameters. The maximum allowed length of *Template_Name* is 80 characters.

*PLNRFRAC ‘*Template_Name*’ *fracture_origin* *BG_NAME ‘*bg_name*’ (*SBG_FZT '*sbg_fzt*') (*SBG_FZL ‘*sbg_fzl*’)

Apply template ‘*Template_Name*’ to *fracture_origin* with the blockgroup name '*bg_name*' and optional fracture zone block groups ‘*sbg_fzt*’ and ‘*sbg_fzl*’. These blockgroup names must not have been defined previously.

For each invocation of *PLNRFRAC (and hence *PLNR_REFINE) does the following:

1. creates the planar fracture using *fracture_origin* and the parameters in the template,
2. assigns the resulting new refined cells to blockgroup ‘*bg_name*’,
3. assigns the cells of the transverse fracture zone to ‘*sbg_fzt*’ (if *SBG_FZT is present),

4. assigns the cells of the longitudinal fracture zone to '*sbg_fzl*' (if *SBG_FZL and *FRAC_CON are present),
5. assigns the template grid-array data to blockgroup '*bg_name*'.
   Note that data cannot be assigned in *PLNRFRAC_TEMPLATE via *SBG_FZT and *SBG_FZL. These structured block groups can be used only after they are defined via keyword *PLNRFRAC.

*PLNR_REFINE *INTO n_{ir} n_{jr} n_{kr}

Construct a local grid for planar (also known as bi-wing) fracture with the specified parameters. Local $n_{ir} \times n_{jr} \times n_{kr}$ grid refinement is done in as many fundamental blocks as needed to contain the fracture. All the resulting fine blocks are gathered together into block group '*bg_name*' which can be used to reference them. In addition, the transverse fracture zone is a single plane of blocks that extends from the centre outward to the fracture tips and can be referenced with *BG '*bg_name*' *FZ or *SBG_FZT; blocks outside this fracture zone can be assigned via *BG '*bg_name*' *NFZ.

fracture_origin

User block address of the origin of the planar hydraulic fracture, where a well perforation connects to the fracture. This cell must be a member of the fundamental grid.

*BWHLEN L_{half}

Specify half-length L_{half} (m | ft) of a planar hydraulic fracture. The fracture refinement extends this distance from *fracture_origin* in the specified direction until the half-length is completely contained, possibly into neighbouring cells.

(*IDIR | *JDIR)

Specify direction in which the planar hydraulic fracture is progagated from *fracture_origin*. The direction keyword is required.

*INNERWIDTH W_{inner}

Specify width W_{inner} (m | ft) of the inner-most planes of blocks for a complex fracture grid. These blocks are tagged as the fracture zone and can be referenced separately via grid-array qualifier *FZ or skipped via *NFZ. Outside of the fracture zone, block sizes normal to the fracture-zone plane increase logarithmically away from the fracture zone. The value of W_{inner} does not refer to the actual (intrinsic) width of the fracture (which is on the order of 0.001 m) but to the width of a fracture conduit on which an effective fracture permeability is calculated so as to maintain the conductivity of the original fracture. Normally values on the order of 2/3m to 1 m are used. The default is 2 ft (0.6096 m).

*WF2 W_{f2}

Specify secondary fracture width W_{f2} (m | ft).

*K2INT k_{2int}

Specify secondary intrinsic permeability k_{2int} (md).

*WF2 and *K2INT re-define the intrinsic fracture width and fracture permeability of the secondary fracture network (natural fractures) within the SRV or planar fracture region.

W_{f2} and k_{2int} along with fracture spacing (*DIFRAC, DJFRAC,*DKFRAC) are used to model the effective permeability enhancement of the naturally fractured network within the SRV or planar fracture region caused by the hydraulic fractures. If *WF2 and K2INT are not present, the effective natural fracture permeability in the SRV or planar fracture region is read from the input permeability arrays (*PERMI *FRACTURE, *PERMJ *FRACTURE, *PERMK *FRACTURE).

*WF2 and *K2INT must be defined together. The use of one of these keywords without the other is an error.

*LAYERSUP $nlay_{up}$

Specify the number of K layers to include above *fracture_origin*. The default is 0. Use this keyword only when *fracture_origin* corresponds to the completion of a horizontal well.

*LAYERSDOWN $nlay_{down}$

Specify the number of K layers to include below *fracture_origin*. The default is 0. Use this keyword only when *fracture_origin* corresponds to the completion of a horizontal well.

*FRAC_CON

Extend the fracture zone across the center of the fracture origin to make longitudinal connection.

{ *grid_array* (*MATRIX|*FRACTURE) (*FZ | *NFZ...) ... }

Any number of grid-array assignments from the following list. For each grid array definition, array qualifier *BG '*bg_name*' is applied internally.

Keyword	Data Section	Recurrent?
*POR	Reservoir Description	No
*PERMI, *PERMJ, *PERMK, *DIFRAC, *DJFRAC, *DKFRAC	Reservoir Description	Yes
*RTYPE, *ROCKDEN, *SORM, *NDARCYCOR, *SWCON, *SWCRIT, *SGCRIT, *SORW, *SORMAX, *SOIRW, *SORG, *SGCON, *SLCON, *SOIRG, *KRWIRO, *KRGCL, *KRGRL, *KROCW, *KROCRW, *KROGCG, *KROGCRG, *PCWMAX, *JFWMAX, *PCGMAX, *JFGMAX, SRFTNW, *SRFTNG	Rock Fluid	Yes

*ADGMAXC, *ADAMAXC *ADGCSTC, *ADACSTC, *ROCKDEN, *COAL-DIF-COMP, *COAL-DIF-TIME, *LANGMULT	Rock Fluid	No
*PRES, *TEMPER, *SW, *SWINIT, *SWNEQ, *ZGLOBAL *ZGLOBALC *ITYPE	Initial Conditions	No

*SBG_FZT '*sbg_fzt*'

Specify the name of the structured block group that is identified with the transverse fracture zone blocks (see *PLNR_REFINE, above). Block group '*sbg_fzt*' is useful for specifying grid-array properties on a per-cell or I-J-K basis, which is not possible with *BG '*bg_name*' *FZ. If *FRAC_CON is used, block group '*sbg_fzt*' lets you distinguish the transverse from the longitudinal fracture zone.

*FRAC_CON

Extend the fracture zone from the fracture center to the edge of the refinements in the direction *perpendicular* to the fracture plane (along the well direction), over the entire fracture height. The resulting extension is called the longitudinal fracture zone. Grid array modifier *FZ will refer to both the transverse and longitudinal fracture zones.

*SBG_FZL '*sbg_fzl*'

Specify the name of the structured block group that is identified with the longitudinal fracture zone blocks created by *FRAC_CON. Block group '*sbg_fzl*' is useful for specifying grid-array properties in the longitudinal fraction zone on a per-cell or I-J-K basis, which is not possible with *BG '*bg_name*' *FZ. Block group '*sbg_fzl*' lets you distinguish the longitudinal from the transverse fracture zone. *SBG_FZL is valid only when *FRAC_CON is used. The transverse and longitudinal fracture zones will share the centre column of cells; if those cells are assigned values by both *SBG_FZT and *SBG_FZL, the last assignment overwrites any previous assignment.

*BG_NAME '*bg_name*'

Specify the name of the planar fracture whose members make up all the refined blocks created for this planar hydraulic fracture.

*MDPLNRBK *fund_blocks*

Specify the maximum number of fundamental grid blocks that each planar fracture may fully or partially cover. The default is 10. This is a primary keyword and should appear at most once. Builder will supply this keyword automatically. This dimensioning information must be supplied explicitly since the final actual number depends upon fracture length relative to local block sizes which are not known at storage allocation time.

DEFAULTS:

If *FRAC_CON is absent, there is no longitudinal fracture zone.

CONDITIONS:

The definition of any given '*Template_Name*' must appear at most once.

Any given *fracture_origin* must be used by *PLNR_REFINE or *PLNRFRAC at most once.

Any given '*bg_name*' must be defined by *PLNR_REFINE or *PLNRFRAC at most once.

A template must be defined before it can be used by *PLNRFRAC.

*SBG_FZL is allowed only if *FRAC_CON is used.

Structured block groups defined via*SBG_FZT and *SBG_FZL can be used only after they are defined via keyword *PLNRFRAC.

EXPLANATION:

Keyword *PLNRFRAC_TEMPLATE allows you to define a template of planar-fracture data that can be applied easily to any number of fundamental-grid cells. That template may be used via *PLNRFRAC in both the "*Reservoir Description*" and "*Well and Recurrent Data*" sections.

After *PLNRFRAC has created blockgroup '*bg_name*', the individual properties assigned by that template may be overwritten with conventional grid-array keywords only after the current stage. For example, I-direction permeability can be overwritten by grid-array *PERMI in the next date section or upon reentry grid module.

No geometry parameters (e.g., block size) associated with *PLNR_REFINE may be overwritten once a planar fracture grid is created.

Example:

Create and assign permeability for a number of planar fractures on one well. Add more fractures (stages) later in the run.

```
*PLNRFRAC_TEMPLATE 'Area C'
      *PLNR_REFINE *INTO 9 9 3
      *BWHLEN 160  ** Fracture half-length, ft
      *IDIR
      *INNERWIDTH 1.8
      *WF2 1.0  *K2INT 1000
   *POR   *MATRIX   *NFZ 0.3
   *POR   *MATRIX   *FZ  0.4 0.3
   *PERMI *MATRIX   *NFZ 1270
   *PERMI *FRACTURE *NFZ 3740
   *PERMI *MATRIX   *FZ 1380 1290
   *PERMI *FRACTURE *FZ 5000 3780
   *PERMJ . . .
*END_TEMPLATE
```

```
*PLNRFRAC 'Area C' 20 29 14 *BG_NAME 'Well 1, Stg 1'
*PLNRFRAC 'Area C' 20 35 14 *BG_NAME 'Well 1, Stg 2'
*PLNRFRAC 'Area C' 20 42 14 *BG_NAME 'Well 1, Stg 3'
*PLNRFRAC 'Area C' 20 48 14 *Bg_NAME 'Well 1, Stg 4'
*PLNRFRAC 'Area C' 20 53 14 *BG_NAME 'Well 1, Stg 5'
   *SBG_FZT 'Well1_Stg5_TRVS'
. . .
** Assign per-cell Sw values to fracture zone
*SW *CON . . .
*SW *FRACTURE *SBG 'Well1_Stg5_TRVS' *ALL ...
. . .
*RUN
*TIME 0
. . .
*TIME 740
*PLNRFRAC 'Area C' 20 58 14 *BG_NAME 'Well 1, Stg 6'
*PLNRFRAC 'Area C' 20 63 14 *BG_NAME 'Well 1, Stg 7'
. . .
```

Dual Porosity (Optional) *DUALPOR

PURPOSE:

*DUALPOR indicates the use of the dual porosity option.

FORMAT:

*DUALPOR

DEFAULTS:

Optional keyword. No default.

CONDITIONS:

This keyword must be located in the RESERVOIR DESCRIPTION keyword group before input of any reservoir property data. Only one of *DUALPOR or *DUALPERM may be specified.

Only one of *DUALPOR, *DUALPERM, *SUBDOMAIN or *MINC may be specified.

*DUALPOR option cannot be used with the *NINEPOINT or *WELLBORE options.

*DUALPOR may be used with local grid refinement *REFINE.

EXPLANATION:

This keyword indicates that a dual porosity option is to be used in the simulator. This option allows each simulator block to have up to two porosity systems, one called its matrix porosity and the other called its fracture porosity. Each porosity can have its own porosity value and its own permeabilities, as well as other distinct properties. Matrix properties are described using the *MATRIX qualifier while fracture properties are described using the *FRACTURE qualifier.

Inter-block flows are calculated in the same manner as they would be for a non-*DUALPOR model. These flows are governed by the fracture properties. An additional set of matrix-fracture flows is also calculated when *DUALPOR is specified. These flows are governed by the matrix properties. The calculation of this matrix-fracture fluid transfer is described more fully later (see also the *SHAPE keyword).

Thus, *DUALPOR allows one matrix porosity and one fracture porosity per grid block, where the matrix is connected only to the fracture in the same grid block. Fracture porosities are connected to other neighbouring fracture porosities in the usual manner. The presence of both fracture and matrix porosities in a block, or just a fracture porosity or a matrix porosity, is under user control (see the *POR and *NULL keywords). Of course, if *DUALPOR was set but no blocks were assigned a matrix porosity, results like that of the standard simulator (no *DUALPOR) would be obtained.

Property definition for *DUALPOR systems usually requires the use of pairs of definitions for most items, one carrying a *MATRIX qualifier and the other a *FRACTURE qualifier. Further details are explained in the descriptions for the individual properties.

See J. E. Warren and P. J. Root, "The Behaviour of Naturally Fractured Reservoirs," SPEJ, Sept. 1963, pp. 245-255 (Trans. AIME, 234) for further details.

Dual Permeability (Optional) *DUALPERM

PURPOSE:

*DUALPERM indicates the use of a dual porosity model (see *DUALPOR above), augmented by inter-block matrix to matrix flows.

FORMAT:

*DUALPERM

DEFAULTS:

Optional keyword. No default.

CONDITIONS:

This keyword must be located in the RESERVOIR DESCRIPTION keyword group before input of any reservoir property data. Only one of *DUALPOR, *DUALPERM, *SUBDOMAIN or *MINC may be specified.

*DUALPERM option cannot be used with the *NINEPOINT or *WELLBORE options.

*DUALPERM may be used with local grid refinement *REFINE.

EXPLANATION:

The keyword *DUALPERM specifies a dual porosity, dual permeability system. The model uses the same calculations as the *DUALPOR model described above, but it is augmented by additional inter-block matrix to matrix flows.

Thus, *DUALPERM allows one matrix porosity and one fracture porosity per grid block, where the matrix is connected to the fracture in the same grid block. Fracture porosities are connected to neighbouring fracture porosities, and the same holds true for neighbouring matrix porosities.

Property definition for *DUALPERM systems usually requires the use of pairs of definitions for most items, one carrying a *MATRIX qualifier and the other a *FRACTURE qualifier. Further details are explained in the descriptions for the individual properties.

Dual permeability has its greatest effect in reservoirs with free gas and large variations in depth for which only the vertical (K direction) matrix to matrix inter-block flows are important. If so, use zero *MATRIX transmissibility modifiers in the I and J directions; see the *TRANSI and *TRANSJ keywords following, and the following example:

```
*TRANSI *MATRIX *CON 0.0
*TRANSJ *MATRIX *CON 0.0
*TRANSK *MATRIX *CON 1.0
```

Dual Porosity Subdomain Method (Optional) *SUBDOMAIN

PURPOSE:

*SUBDOMAIN indicates the use of a dual porosity model using the subdomain method.

FORMAT:

*SUBDOMAIN idiv

DEFINITIONS:

idiv
: Number of subdivisions for each matrix blocks, typically chosen from the range of 2 to 5.

DEFAULTS:

Optional keyword. No default.

CONDITIONS:

This keyword must be located in the Reservoir Description keyword group, before the *REFINE and *RANGE keywords. Only one of *DUALPOR, *DUALPERM, *SUBDOMAIN, or *MINC may be specified. *SUBDOMAIN may not be used together with the *NINEPOINT, or *REFINE options.

The use of grid refinement is supported, but if the grid is refined vertically and the refined block thickness is less than *DKFRAC, *DKFRAC will be set to the refined block thickness. This will result in an altered capillary holdup for the refined region. The user should be aware of how vertical refinement will alter *DKFRAC and capillary holdup.

When using the Subdomain option, the Grid Module which initializes grid block properties and interblock transmissibilities must not be called from Recurrent Data. This prevents the user from setting transmissibility multipliers in recurrent data and defining any grid refinements/modifications in recurrent data.

EXPLANATION:

The description given above for the *DUALPOR keyword should be studied first, as this option is closely related.

This option allows each simulator block to have up to two porosity systems, one called the matrix porosity and the other called the fracture porosity. Moreover, the *SUBDOMAIN option splits up the matrix porosity vertically into "idiv" equal thickness pieces. Inter-block fracture to fracture, and matrix to fracture flows within a block, are calculated. Also, matrix to matrix flow between the matrix layers within a block are calculated.

The *SUBDOMAIN method allows the modelling of segregating flows within the matrix porosity of a block.

Each block will have its own porosity value and its own permeabilities, as well as other distinct properties. Matrix properties are described using the *MATRIX keyword while fracture properties are described using the *FRACTURE keyword. Property definition usually requires the use of pairs of definitions for most items, one carrying a *MATRIX identifier and the other a *FRACTURE identifier. Further details are explained in the descriptions for the various properties.

Within a subdomain block, *DKFRAC (not block thickness) is used to calculate capillary holdup. *DKFRAC may be less than the block thickness, but can be no greater than block thickness. If *DKFRAC greater than the thickness of a Subdomain block is encountered *DKFRAC for that block will be set equal to the block thickness.

For details of this method, please see J. R. Gilman, "An Efficient Finite-Difference Method for Simulating Phase Segregation in the Matrix Blocks in Dual-Porosity Reservoirs", SPERE, July 1986, pp. 403-413.

Assigning Fractional Volumes to Subdomain blocks (Optional) *FRACVOL

PURPOSE:

*FRACVOL allows the user to alter the size of individual subdomain blocks (*SUBDOMAIN) or nested rings (*MINC) relative to each other. In *SUBDOMAIN model, this is used to improve subdomain capillary holdup calculations and is crucial in the proper modeling of reinfiltration.

FORMAT:

*FRACVOL frac_1 frac_2 … frac_idiv

DEFINITIONS:

frac_1
: Fractional volume of first subdomain / outermost ring block

frac_2
: Fractional volume of second subdomain / ring block

frac_idiv
: Fractional volume of last subdomain / innermost ring block

DEFAULTS:

Optional keyword. No default. If omitted subdomain / ring volumes are all set equal to 1.0/idiv.

CONDITIONS:

This keyword must be located in the Reservoir Description keyword group. *SUBDOMAIN or *MINC must be specified. Values of *FRACVOL as low as 0.01 have been used without difficulty.

EXPLANATION:

It allows the user to adjust the size of stacked subdomain or nested ring blocks relative to each other according to *FRACVOL values. In order to obtain the correct capillary holdup when using the Subdomain model it is optimal to give the first and last division the smallest possible fractional volume which does not significantly influence CPU time. In the case of a very small number of subdomain divisions when *TRANSD > 0.0 and *SD_REINF > 0.0 it may be sufficient to reduce the size of the last subdomain division only. See Keywords *TRANSD and *SD_REINF.

Examples:

The following provides examples of *FRACVOL usage:

```
*SUBDOMAIN 5
*FRACVOL 0.02 0.32 0.32 0.32 0.02

** or if the number of subdomains is limited and
** TRANSD > 0.0 and SD_REINF > 0.0

*SUBDOMAIN 4
*FRACVOL 0.33 0.33 0.32 0.02
```

Subdomain to Subdomain Flow Transmissibility Multipliers (Optional) *TRANSD

PURPOSE:

*TRANSD indicates input of an array of Inter-Subdomain (vertical) transmissibility multipliers and enables the subdomain to subdomain vertical matrix connection.

ARRAY:

*TRANSD

DEFAULTS:

Optional keyword. Defaults: 0.0

CONDITIONS:

This keyword may only be in the Reservoir Description keyword group and is only valid when the *SUBDOMAIN dual porosity option is in effect.

Array qualifiers *MATRIX and *FRACTURE are not allowed.

The application of these multipliers depends upon *TRANSMULT.

EXPLANATION:

In the standard Subdomain dual porosity model, the fracture and matrix are modeled as two separate grid systems. For a single spatial block, the fracture is modeled using a single grid block while the matrix is modeled using a vertical stack of grid blocks.

Within a single spatial block, the matrix stack can only develop fluid flow within the stack and between each block of the stack and the fracture block.

The use of the TRANSD array allows the matrix stacks in each subdomain to be connected vertically together by an additional vertical connection per spatial block. The last matrix block in a subdomain is connected to the first matrix block in the block below it. This allows gravity drainage to occur directly from subdomain to subdomain.

There is capillary continuity between subdomains unless *SD_REINF is used. See keywords FRACVOL and SD_REINF.

Examples:

The following provides examples of TRANSD usage:

```
** Turn on direct gravity drainage between subdomains
** for all blocks

*TRANSD *CON 1.0

** Turn on direct gravity drainage between block 2,3,6
** and 2,3,7 only

*TRANSD *IJK 2:2 3:3 6:6 1.0
```

The suggested range of values for transmissibility multipliers is:

	SI	Field
min	0.0	0.0
max	1.0E+4	1.0E+4

Dual Porosity MINC Method (Optional) *MINC

PURPOSE:

*MINC indicates the use of a dual porosity model using the multiple-interacting-continua (MINC) approach.

FORMAT:

*MINC idiv

DEFINITION:

idiv
Number of subdivisions for each matrix blocks, typically chosen from the range of 2 to 5.

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be located in the Reservoir Description keyword group, before the *REFINE and *RANGE keywords.

Only one of *DUALPOR, *DUALPERM, *SUBDOMAIN, or *MINC may be specified. *MINC option cannot be used with the *NINEPOINT, or *REFINE options.

EXPLANATION:

The description given above for the *DUALPOR keyword should be studied first, as this option is closely related.

This option allows each simulator block to have up to two porosity systems, one called the matrix porosity and the other called the fracture porosity. Moreover, the *MINC option splits up the matrix porosity into "idiv" nested rings. Inter-block fracture to fracture, and matrix to fracture flows within a block, are calculated. Also, matrix to matrix flow between the matrix rings within a block are calculated.

The *MINC method allows the modelling of some transient behavior within the matrix porosity of a block.

Each porosity will have its own porosity value and its own permeabilities, as well as other distinct properties. Matrix properties are described using the *MATRIX keyword while fracture properties are described using the *FRACTURE keyword. Property definition usually requires the use of pairs of definitions for most items, one carrying a *MATRIX identifier and the other a *FRACTURE identifier. Further details are explained in the descriptions for the various properties.

For details of this method, please refer to K. Pruess and T. N. Narasimhan, "A Practical Method for Modelling Fluid and Heat Flow in Fractured Porous Media", SPEJ, Feb. 1985, pp. 14-26.

Shape Factor Calculation (Conditional) *SHAPE

PURPOSE:

*SHAPE describes the method (which type of shape factor) is to be used in calculating matrix to fracture flows within blocks for dual porosity models.

FORMAT:

```
*SHAPE    (*GK)
          (*WR)
```

DEFINITIONS:

*GK

This subkeyword indicates the use of a Gilman and Kazemi style formulation for the shape factor.

*WR

This subkeyword indicates the use of a Warren and Root style formulation for the shape factor.

DEFAULTS:

Conditional keyword. Default: *SHAPE *GK.

CONDITIONS:

This keyword must be located in the RESERVOIR DESCRIPTION keyword group and is only relevant when one of *DUALPOR or *DUALPERM is specified.

EXPLANATION:

As a general rule, fluid flow between small porous regions is proportional to a transmissibility that consists of a sum of terms that are each a surface area multiplied by a permeability and divided by a regional dimension perpendicular to the area. By multiplying the numerator and denominator of such terms by the regional dimension, the terms become volumes multiplied by permeabilities, which are divided by the squares of regional dimensions.

If the regions are now taken to be disconnected portions of the matrix porosity in a grid block, separated by thin fracture planes, the regional dimensions become the fracture spacings. Assuming that there are many similar such matrix regions connected to fractures that pervade the grid block, a total transmissibility for matrix to fracture flow within the block can be obtained by scaling up the disconnected matrix portion volumes to the total matrix porosity volume. The shape factor to be discussed here concerns what then happens to the other terms making up the total transmissibility.

The Warren and Root formulation has matrix to fracture flow within a grid block proportional to:

W & R Transmissibility = 20/3 * k * ((1/Lx + 1/Ly + 1/Lz)**2) * M-Vol;

the Gilman and Kazemi formulation uses:

G & K Transmissibility = 4 * k *(1/(Lx**2) + 1/(Ly**2) + 1/(Lz**2)) * M-Vol

The above assumes that fracture planes exist in all three directions with fracture spacings Lx, Ly, Lz (which are set using the *DIFRAC, *DJFRAC and *DKFRAC keywords following), the permeability is k, and the total matrix volume is M-Vol. Further details for this and other cases are given in the references noted before.

The *SHAPE keyword selects which of the above is to be used. The Gilman and Kazemi factor (*SHAPE *GK) is recommended due to the straightforward way it incorporates various anisotropies.

Note that the basic transmissibility formulas are used even when the fracture spacings exceed the grid block size. These cases correspond to dividing up the matrix to fracture flow over several grid blocks and modelling the usual matrix to fracture flows over the individual blocks. Also, the basic formulae are used even when the grid block containing the matrix regions is itself not cubic in form (such as for corner point, radial or hybrid refined grids).

Matrix-Fracture Transfer Calculation (Conditional) *TRANSFER

PURPOSE:

*TRANSFER indicates the type of matrix-fracture model for treating different phases in dual-porosity or dual-permeability reservoirs.

FORMAT:

*TRANSFER ntran

DEFINITIONS:

ntran

The matrix-fracture model number:

= 0; No special treatment;

= 1; Complete phase segregation model;

= 2; Pseudo-capillary pressure model;

= 3; Pseudo-capillary pressure model with corrections to contact areas between phases.

See Appendix A for more information.

DEFAULTS:

Conditional keyword. Default: *TRANSFER 3

CONDITIONS:

This keyword must be located in the RESERVOIR DESCRIPTION keyword group. This keyword is required only for naturally fractured reservoirs as specified by either the *DUALPOR or *DUALPERM keyword.

EXPLANATION:

Refer to Appendix A for more information.

Note that *TRANSFER 0, representing the usual dual porosity flow model derived from single phase considerations, does not accurately model all gravity drainage processes.

Fracture Spacing (Conditional) *DIFRAC, *DJFRAC, *DKFRAC

PURPOSE:

*DIFRAC indicates the input of fracture spacings for the I direction.

*DJFRAC indicates the input of fracture spacings for the J direction.

*DKFRAC indicates the input of fracture spacings for the K direction.

ARRAY:

*DIFRAC
*DJFRAC
*DKFRAC

DEFAULTS:

Conditional keywords. No defaults.

CONDITIONS:

These keywords must be in the RESERVOIR DESCRIPTION keyword group. These keywords are required with *DUALPOR or *DUALPERM.

Setting value(s) to zero for the *DIFRAC keyword means that there are no fracture planes perpendicular to the I axis in those blocks, and similarly for the other directions. (Setting values to 0 actually corresponds to an infinite, and hence ineffective, spacing in that direction.)

If a block is assigned a zero value for each of *DIFRAC, *DJFRAC, and *DKFRAC, then the block's fracture porosity will be declared null and will not participate in any simulator calculations.

EXPLANATION:

Fracture spacings are used to calculate the matrix to fracture transfer coefficient as detailed above for the *SHAPE keyword (see also Appendix A).

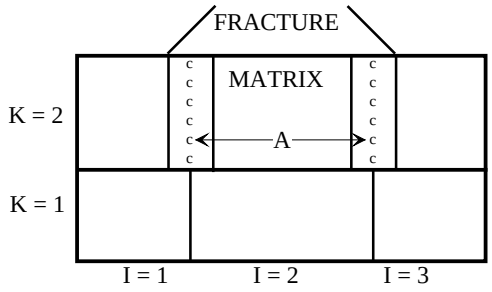
Fracture spacings should be measured from centre line to centre line in the appropriate direction. The basic transmissibility formulas (see the *SHAPE keyword) are valid even when the fracture spacings exceed the grid block size, so large fracture spacings should not be a concern (except that weak matrix to fracture coupling occurs).

*DIFRAC and *DJFRAC fracture spacings are required for *GRID *RADIAL situations. For such grids, *DIFRAC should be thought of as measuring spacings in the "x"-direction (corresponding to the 0 degree axis) and *DJFRAC to orthogonal spacings.

Spacings are generally inherited from parent blocks for *HYBRID grids and hence, are automatically available.

Any block with its DIFRAC = DJFRAC = DKFRAC = 0 will be assumed to have no fracture porosity.

The following diagram illustrates a system where there are parallel fractures. The fracture spacing is measured from centre of fracture to centre of fracture (line A):



Null Block Indicator (Optional) *NULL

PURPOSE:

*NULL indicates the input of an array of null block indicators which can be used to mark individual porosities as non-participating in dual porosity models, or entire blocks as non-participating.

ARRAY:

*NULL

DEFAULTS:

Optional keyword. Default: All blocks are active (participating), with no null blocks.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

The numerical values in the incoming array must be zeroes (0) for null or inactive blocks, or ones (1) for active or participating blocks.

EXPLANATION:

All blocks are considered active unless a block is specifically designated as being null. Null blocks do not participate in any of the simulator's calculations. Moreover, they are a barrier to any flow.

The *NULL keyword and array provides one method for making blocks null. Use the following flags in the *NULL array:

0 = null block
1 = active block.

Other methods are also available for making blocks null; see the *POR keyword, for instance. Note that if a block is marked null, setting nontrivial properties, including a nonzero porosity, will NOT cause the block to become active again.

If a dual porosity model is being used, selective participation of the two porosities can be controlled with the *NULL keyword. Using *NULL with no *MATRIX or *FRACTURE qualifier nulls the block (including both porosities) and makes it a barrier to flow. Nulling the block once with one of *MATRIX or *FRACTURE, and then again with the other qualifier has the same effect. Using *NULL *MATRIX and setting each of *DIFRAC, *DJFRAC, and *DKFRAC equal to 0 for a block accomplishes the same task. Using just *NULL *MATRIX or *NULL *FRACTURE makes only one of the porosities non-participating. Flow can occur to the other porosity as required.

Note that a pinched out status set by the *PINCHOUTARRAY keyword over-rides *NULL settings. See the description of *PINCHOUTARRAY following. *NULL settings over-ride pinch out setting generated by use of the *PINCHOUT-TOL keyword, or zero thickness situations.

Example: Cause certain blocks in a 10 x 10 x 3 system to be null:

```
*NULL *IJK
1:10   1:10   1:3   1
 1:5  10:10   1:3   0
 1:4    9:9   1:3   0
 1:1    6:6   1:3   0
```

Porosity (Required) *POR

PURPOSE:

*POR indicates input of porosities, where zero values can be used to mark individual porosities as null (non-participating) in dual porosity models, or entire blocks as null.

ARRAY:

*POR

DEFAULTS:

Required keyword. No defaults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. Porosity must be specified for each grid cell, either directly via *POR or indirectly via *PORVOL. If both keywords specify values for a cell, the *PORVOL value is used.

EXPLANATION:

The *POR array assigns blocks their porosity values. The fluid (pore) pressure at which these porosities are to apply is given using the *PRPOR (see following) or through use of the *CROCK keywords (see Rock-Fluid section).

Porosities are dimensionless. Thus, the *POR array should consist of values between 0.0 and 1.0. Percentages should NOT be entered. Zero values have special significance in setting blocks null.

Zero values can mark individual porosities as null (non-participating) in dual porosity models. This should be done using the *MATRIX and *FRACTURE qualifiers. An entire block becomes null when both its porosities are null.

Note that in dual porosity models, fracture porosity is the fraction of void space in the fracture system considered with respect to a volume of the bulk reservoir rock. Matrix porosity is the fraction of void space in a piece of unfractured matrix material examined independently of any fractures.

Example:

A grid with 5 blocks in the I direction, 3 blocks in the J direction, and a single layer is used to model a dual porosity system. Only those blocks with J=2 are fractured. Block (1,1,1) is not part of the reservoir. Porosities could be input as:

```
*POR  *MATRIX  *IJK
1:5   1:3  1   0.16
1     1    1   0.00
*POR  *FRACTURE  *IJK
1:5   1:3  1   0.00
1:5   2    1   0.04
```

The acceptable range of values for (dimensionless) porosities is:

	SI dimensionless	**Field dimensionless**
min	0.0	0.0
max	1.0	1.0

Pore Volume (Optional) *PORVOL

PURPOSE:

Specify grid cell porosity indirectly via pore volume.

ARRAY:

*PORVOL

DEFAULTS:

Optional keyword. For each grid cell for which *PORVOL is not specified, its pore volume is calculated from porosity and bulk volume.

CONDITIONS:

Porosity must be specified for each grid cell, either directly via *POR or indirectly via *PORVOL. If both keywords specify values for a cell, the *PORVOL value is used.

A cell's minimum allowed *PORVOL value is zero.

EXPLANATION:

See the manual page for keyword *POR, since PORVOL specifies porosity indirectly.

When a cell's pore volume V_p is calculated from keyword *POR, the formula is

$$V_p = V_b \cdot \phi$$

where

V_b	bulk volume (from geometry, e.g., *DI, *DJ, *DK, *VOLMOD)
ϕ	porosity (from *POR)

Keyword *PORVOL allows you to specify pore volume V_p directly, in which case the input porosity is calculated as

$$\phi = V_p / V_b$$

If the calculated ϕ exceeds 1, a warning message (*PORVOL exceeds available volume) is issued and ϕ is limited to 1.

After the porosity value is determined via *POR and *PORVOL, the pore volume used in the simulation is re-calculated from porosity.

A zero value of *PORVOL has the same effect as a zero value of *POR. See the *POR keyword manual page.

The unit of *PORVOL is volume (m^3 | ft^3).

Natural Fracture Systems

For natural fracture grids, pore volume can be specified for the matrix and fracture cells separately by use of the *MATRIX and *FRACTURE qualifiers.

Example:

```
*PORVOL     *KVAR 100 200 300
```

This will set the pore volume of each layer to be 100, 200 and 300 respectively.

Rock Compressibility *PRPOR, *CPOR, *DCPOR

PURPOSE:

*PRPOR signals the input of a reference pressure for the rock compressibility determination. *CPOR signals the input of rock compressibility. *DCPOR signals the input of pressure dependency of the rock compressibility, *CPOR

FORMAT:

*PRPOR	(*MATRIX)	ref_pressure
	(*FRACTURE)	
*CPOR	(*MATRIX)	cpor
	(*FRACTURE)	
*DCPOR	(*MATRIX)	dcpor
	(*FRACTURE)	

DEFINITIONS:

ref_pressure
: Reference pressure (kPa | psi).

cpor
: Pressure dependence of formation porosity (rock compressibility) (1/kPa | 1/psi).

dcpor
: Pressure dependence of the rock compressibility (1/kPa ** 2 | 1/psi ** 2).

DEFAULTS:

*CPOR will default to 0.0 and *PRPOR will default to 1 atmosphere (101.3 kPa | 14.7 psi).

*DCPOR is an optional keyword with a default of 0.0.

Note that GEM provides an alternate, independent way of entering this type of data using the *CROCKTYPE keyword (see later). Also, the *CROCK keyword is available (see the Rock-Fluid Data section).

<u>If *CROCKTYPE does appear, settings for *CPOR and *PRPOR will be mostly ignored</u>. (The default aquifer compressibility does use the *CPOR value.) Thus, blocks that are not assigned compressibilities using the *CCPOR/*CPRPOR subkeywords of *CROCKTYPE will end up with no compressibility at all.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

The functional form used in the calculation of porosity por in a formation containing fluid at pressure p is:

por(p) = por * [1 + cpor * (p - ref_pressure) + dcpor * (p - ref_pressure) ** 2]

where "por" is the initial porosity input using the *POR keyword and p is the calculated oil phase pressure calculated internally by GEM.

These cpor, dcpor and ref_pressure will be used to calculate the porosities for all rock types unless they have their own compressibilities specified via *CROCK in the rock-fluid property section (see following).

In dual porosity systems both matrix and fracture reference porosities and rock compressibilities should be input. The *MATRIX and *FRACTURE subkeywords are used to indicate the matrix and fracture rock compressibilities.

For example, the input porosity values for a dual porosity system, where the reservoir has uniform porosities of .11 and .02 for the matrix and the fractures at the reference pressure of 14.7 psia. The rock compressibilities are 3.0E-6 for both the matrix and the fracture.

```
*PRPOR *MATRIX     14.7
*CPOR *MATRIX    3.0E-6

*PRPOR *FRACTURE   14.7
*CPOR *FRACTURE  3.0E-6

*POR *MATRIX   *CON .11
*POR *FRACTURE *CON .04
```

The acceptable range of values for reference pressure is:

	SI	**Field**
	kPa	**psi**
min	0.0	0.0
max	1.0E+5	14,504.0

The acceptable range of values for the formation rock compressibility is:

	SI	**Field**
min	0.0	0.0
max	0.001	0.001

Pore Volume Modifiers (Optional) *VOLMOD

PURPOSE:

*VOLMOD indicates input of an array of pore volume modifiers.

ARRAY:

*VOLMOD

DEFAULTS:

Optional keyword. Default: 1.0

In dual porosity models, pore volume modifiers can be applied to the matrix and fracture pore volumes separately by use of the *MATRIX and *FRACTURE qualifiers. However, the default for fracture multipliers is still 1.0, even if *VOLMOD *MATRIX (or even just *VOLMOD) appeared, and the default for matrix multipliers is still 1.0, even if *VOLMOD *FRACTURE appeared.

It is recommended that *VOLMOD *MATRIX and *VOLMOD *FRACTURE be used together, or not at all, for dual porosity models.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

Values for pore volume modifiers must be non-negative, and they may exceed 1.

EXPLANATION:

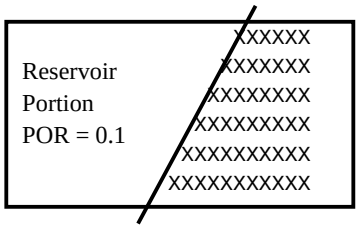
Pore volumes are calculated in the simulator based on the reservoir rock bulk volume, which is calculated from the external grid block dimensions, and the porosity, which depends on pore-pressure and compressibility data. The pore volumes accumulate fluids in the simulator and are basic to all its calculations.

Pore volume modifiers are multiplicative and are used to modify the pore volume calculation given above. These modifiers can be used to adjust pore volumes for a reservoir boundary that passes through a block, thereby leaving a portion of the grid block out of the reservoir, or for a flow unit boundary that may be encountered when simulating pattern floods. It is recommended that pore volume modifiers be used for these situations, rather than adjusting the porosities directly, so that unusual porosity values are avoided.

Note that multipliers larger than 1 may be used to associate volume external to the reservoir with a grid block. For instance, total wellbore volume for a well can be associated with the grid block through which the well penetrates by using a multiplier for that block.

Example:

The region on the right side of the following block is not part of the reservoir and constitutes .4 of the volume of the grid block shown. Input the true average porosity for the portion of the grid block that lies in the reservoir and assign a block volume multiplier of .6 to the block.



Block (2, 3, 4)

```
*POR     *IJK ...
              2 3 4 0.1
              ...
*VOLMOD  *IJK 2 3 4 0.6
```

Multipliers for the other blocks will default to 1.

The acceptable range of values for pore volume modifiers is:

	SI **m**	**Field** **ft**
min	0.0	0.0
max	1.0E+4	32,808.0

Block Geometry Modifiers (Optional) *VAMOD, *VATYPE

PURPOSE:

Describes modification to grid block volumes and face areas.

FORMAT:

*VAMOD key v ai aj ak (ai- aj- ak-) (*9P aij+ aij-)

ARRAY:

*VATYPE

DEFINITIONS:

key

Integer key associated with this geometry type, to be used with *VATYPE. You do not need to define a key for the unmodified type or null-block type. It is suggested that you define your own modifier types using key = 2 and up, leaving predefined key = 0 for null blocks and predefined key = 1 for unmodified (whole) blocks.

v

Block volume modification factor, equal to (desired gross volume) / (product of block sizes *DI, *DJ and *DK). It is needed even for zero-porosity blocks to correctly account for energy in rock. A value of zero denotes a true null block, with no pore volume and no rock volume.

ai

Area modifier factor in the I direction, equal to (desired area) / (area from block sizes *DJ and *DK). A zero value will result in no flow.

aj

Area modifier factor in the J direction, equal to (desired area) / (area from block sizes *DI and *DK). A zero value will result in no flow.

ak

Area modifier factor in the K direction, equal to (desired area) / (area from block sizes *DI and *DJ). A zero value will result in no flow.

ai-

Area modifier factor in the -I direction, used in situations where ai varies along the I direction.

aj-

Area modifier factor in the -J direction, used in situations where aj varies along the J direction.

ak-

Area modifier factor in the -K direction, used in situations where ak varies along the K direction.

aij+

Area modifier factor in the I+J+ direction for *NINEPOINT *IJ and in the I+K+ direction for *NINEPOINT *IK. This is needed only along the diagonal boundaries of symmetry patterns.

aij-

Area modifier factor in the I+J- direction for *NINEPOINT *IJ and in the I+K- direction for *NINEPOINT *IK. This is needed only along the diagonal boundaries of symmetry patterns.

VATYPE

Assign a modifier type key to the grid, including refined blocks. A key value of 0 denotes a null block. A key not defined using *VAMOD refers to the unmodified type. See 'key', above.

DEFAULTS:

If *VAMOD and *VATYPE are absent, all blocks are active and their full volumes and areas are used.

If ai-, aj- and ak- are absent then ai- = ai, aj- = aj and ak- = ak. This is appropriate when the factor does not vary along its direction. When it does, the + and - face of a block will have a different factor, and both ai and ai- must be given values.

EXPLANATION:

Typical Uses of Geometry Modifiers

Typical uses for block geometry modifiers are:

1. Place centres of outer blocks on the reservoir boundary,
2. Model symmetry elements of repeating patterns, and
3. Model a reservoir with an irregular shape.

In any case, the technique is the same:

- Define initial grid with keywords *GRID, *DI, *DJ, *DK and *DTOP,
- Trim grid with geometry modifiers to get desired volumes, etc.,
- Enter rock and fluid properties as for whole blocks, and
- Apply well and completion fractions to calculate well indices.

Once the geometry modifiers are defined, enter properties on the usual per-gross-volume basis. The geometry factors will be applied during initialization to quantities derived from volumes and areas. For example, the derived quantity Block Pore Volume will include the "v" factor, but the user input property Porosity will not.

Referencing Grid Block Faces

An area modifier applies to the interface between the current block and the adjacent block in the "+" coordinate axis direction. The "+" direction is the direction that takes you away from the origin of the coordinate system. For a cylindrical grid, apply this idea after "unrolling" the grid into a Cartesian grid.

Figure 1 illustrates this rule. Grid block numbering starts at the origin which is at the lower left corner. Let block (4,1,2) be the current block, that is, area modifiers are assigned to block (4,1,2). Area modifier "ai" is applied to the interface in the +I direction linking (5,1,2) to (4,1,2). Area modifiers for the "-" directions are rarely needed since value for those interfaces default to the value of the "+" direction modifier of the adjacent block. Therefore, the interface between (3,1,2) and (4,1,2) is assigned via "ai" for (3,1,2).

A "-" direction area factor is needed only when the value of the factor varies in that direction, as mentioned in the above DEFAULTS section. For an example, consider the bottom row of blocks in Figure 1. Suppose each block interface in the I direction has a different volume and area factors V1, A1, V2, A2, etc. Do the following to get the area factors consistent, assuming ak = v and aj = 1 for each geometry type:

*VAMOD key1 V1 A1 1 V1 ** factors for (1,1,1)
*VAMOD key2 V2 A2 1 V2 A1 1 V2 ** factors for (2,1,1)
*VAMOD key3 V3 A3 1 V3 A2 1 V3 ** factors for (3,1,1)
etc.

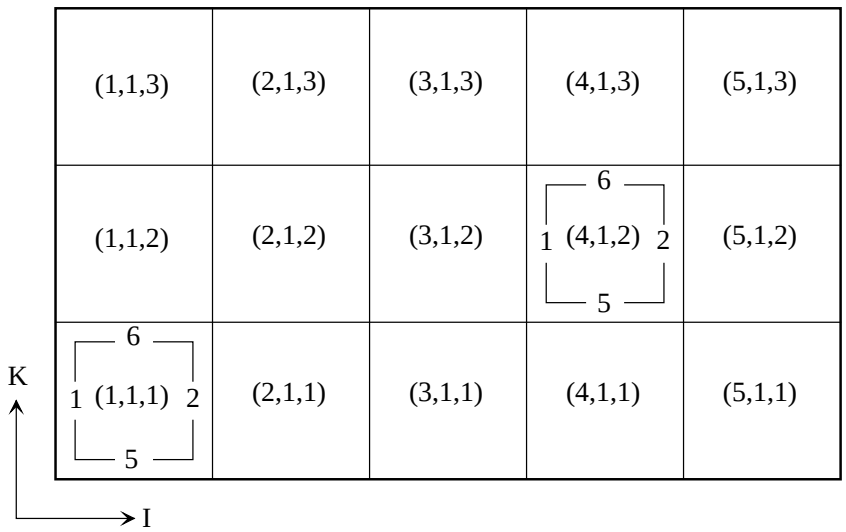


Figure 1: Referencing Grid Block Faces

Definitions of Geometry Factors

Figure 2 illustrates graphically the concept behind geometry modifier factors. Suppose we wished to place a block node (located at the block centre) on the reservoir YZ boundary with the desired block size DX, as shown on the left of Figure 2. To do this, assign an I-direction size of DX' = 2*DX for this boundary block, and trim the block with *VAMOD to get the desired volume and flow areas. The meanings of the factors are

v = [desired volume] / [volume from block sizes]
= [DX * DY * DZ] / [DX' * DY * DZ]
= 0.5
ai = [desired area] / [area from block sizes]
= [DY * DZ] / [DY * DZ]
= 1
aj = [desired area] / [area from block sizes]
= [DX * DZ] / [DX' * DZ]
= 0.5
ak = [desired area] / [area from block sizes]
= [DX * DY] / [DX' * DY]
= 0.5

and the keywords are

```
*DI *IVAR DX' . . .        ** Assign DX' to boundary block
*VAMOD key 0.5 1 0.5 0.5   ** Split block in half in X
                           ** direction
```

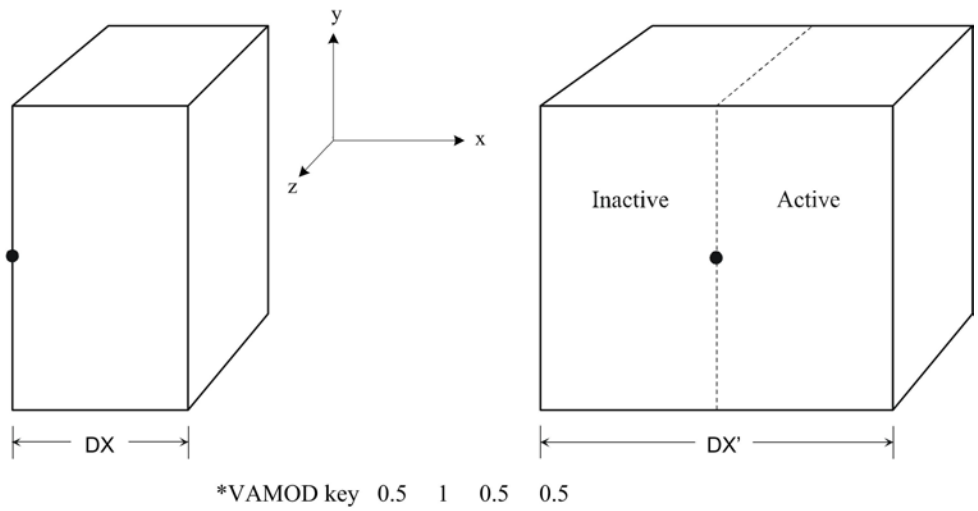


Figure 2: Grid Node on a Side Block Boundary

Figure 3 shows how to place a block node on the reservoir corner. Both the X and Y directions are extended, that is, DX' = 2*DX and DY' = 2*DY, and then trimmed by 1/2. The meaning of the volume factor is

v = [desired volume] / [volume from block sizes]
= [DX * DY * DZ] / [DX' * DY' * DZ]
= 0.25

and the keywords are

```
*DI *IVAR DX' . . .          ** Assign DX' to boundary block
*DJ *JVAR DY' . . .          ** Assign DY' to boundary block
*VAMOD key 0.25 1 0.5 0.25   ** Split block in half in X
                             ** direction
```

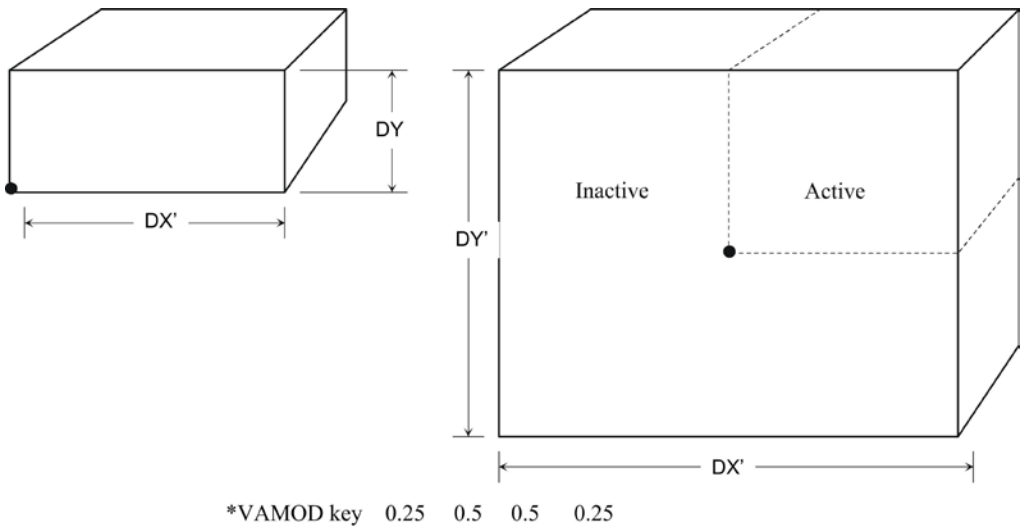


Figure 3: Grid Node in a Block Corner

Figure 4 shows how to place a block node on a diagonal boundary. Both the X and Y directions are extended, but the geometry factors come more from inspection. Here, the nine-point option is illustrated. The meaning of the geometry factors is

v	=	[desired volume] / [volume from block sizes] = 0.5 by inspection
ai	=	1 since this whole face is on the active part of the block,
aj	=	0 since this face is on the inactive part of the block, and
ak	=	v = 0.5,
aij+	=	[desired diagonal "area"] / [existing diagonal "area"] = 0.5 by inspection
aij-	=	[desired diagonal "area"] / [existing diagonal "area"] = 1 by inspection

and the keywords are

```
*NINEPOINT *IJ
*DI *IVAR DX' . . .          ** Assign DX' to boundary block
*DJ *JVAR DY' . . .          ** Assign DY' to boundary block
*VAMOD key 0.5 1 0 0.5 *9p 0.5 1
                             ** Split block in half in X dir
```

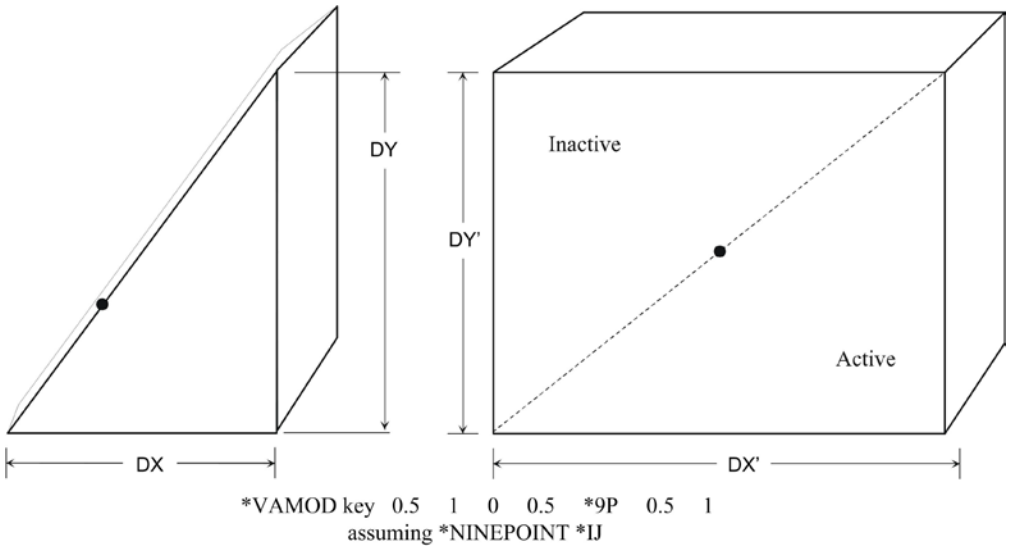


Figure 4: Grid Node on a Diagonal Boundary

Null Blocks

You can use *VATYPE to specify null blocks instead of *NULL. In fact, this is preferred if any geometry modifiers will be assigned via *VAMOD. Use key value 0 for null blocks, just as for *NULL.

There is no inter-block connection to null blocks or beyond the reservoir boundary, and area modifiers corresponding to such connections are not needed internally. When a number is required to satisfy the syntax of the *VAMOD keyword, but you know it will not be used, enter 0.

Example: One-eighth of a 5-spot Symmetry Element

Apply the 9-point option to a 9x5 grid of square blocks, and then trim to 1/8 of a 5-spot pattern.

```
*GRID *CART 9 5 1 *NINEPOINT *IJ

*DI *CON 10
*DJ *EQUALSI

**    key  v      ai  aj    ak        aij+  aij-
*VAMOD 2  0.5    0.5  1.0  0.5                   ** like Fig 6
*VAMOD 3  0.5    1.0  1.0  0.5   *9P  0.5   1.0  ** like Fig 8
*VAMOD 4  0.5    1.0  1.0  0.5   *9P  1.0   0.5  ** like Fig 8
*VAMOD 5  0.125  0.5  1.0  0.125 *9P  0.5   1.0
*VAMOD 6  0.25   1.0  1.0  0.25  *9P  1.0   0.5
*VATYPE *ALL 5 2 2 2 2 2 2 2 5
            0 3 1 1 1 1 1 4 0   ** ---- i
            0 0 3 1 1 1 4 0 0   ** |
            0 0 0 3 1 4 0 0 0   ** |
            0 0 0 0 6 0 0 0 0   ** j
```

The only difference between keys 3 and 4 is the *9P values; these keys could be merged if *NINEPOINT were not used. Array-reading option *ALL was used with *VATYPE so that the data itself can make a picture of the grid when arranged in rows of ni, columns of nj and planes of nk.

Refined Grid

By default, all blocks of a refined grid have the same values for a quantity or property (except block size) as the grid's parent block. This applies also to block modifiers. Geometry modifiers may be entered for specific refined blocks using the subkeyword *RG.

Suppose that the block in Figure 2 is to be refined 3x3 areally. Of the nine finer blocks, three fall entirely in the inactive zone, three fall entirely in the active zone and three are split in half just as the parent block was. In addition to the keywords indicated above for Figure 2, the following are needed for this refined grid case:

```
*REFINE block_address *INTO 3 3 1
*VATYPE *RG block_address *IVAR 0 key 1
```

where "key" is the same one used for the parent block. We divided the I direction into an odd number of fine blocks so that the new block nodes fall on the reservoir boundary, as it did for the parent block.

If the external faces of a refined grid are connected to an unrefined block in the "+" direction, then the area modifiers of the refined blocks are used. If the external faces of a refined grid are connected to another refined grid, then the smallest effective area is used.

Hybrid Grids

The specification of area modifiers for hybrid grid blocks is more complicated because the method of referring to the radial, angular or axial directions differs from the fundamental grid's I, J or K system. The correspondence between these direction systems is shown in the section "Direction Dependent Data" of keyword *REFINE. In general, the following can be used for partial hybrid blocks (*IDIR, etc., denote the hybrid grid's orientation):

```
v ai aj ak
0.5 0.5 1.0 0.5      ** hybrid half-block *IDIR & *KDIR
0.5 0.5 0.5 1.0      ** hybrid half-block *JDIR
0.25 0.25 0.25 0.25  ** hybrid innermost quarter-block
```

The following data fragment shows how to place the center of a hybrid grid on a reservoir boundary edge. See the sample testbed data files for more examples of typical cases.

```
** Vertical hybrid grid on reservoir boundary in
** column I=3, J=1 refine 3 1 1:4 into 3 4 1 hybrid kdir
**    key  v   ai   aj   ak
vamod  2  0.5  1.0  0.5  0.5  ** I=1 plane
vamod  3  0.5  0.5  1.0  0.5  ** hybrid half-block *KDIR

** Assign geometry types to fundamental I=1 plane
vatype con 1
  mod 1 1:4 1:4 = 2
** Assign geometry types to hybrid blocks using diagram in
** section "Hybrid Grid Orientations" of *REFINE description.
```

```
** Hybrid's j'=1 & 3 are in fund. J-K plane (split in half),
** j'=2 is on inner (full) side of reservoir boundary (next
**    to J=2), and
** j'=4 is on outer (null) side of reservoir boundary.
vatype rg 3 1 1:4 jvar 3 1 3 0
```

Well Completion in a Partial Block

If a well is completed in a partial block, it may be necessary to modify the well index specified via keyword *PERF. Most “fraction” issues of wells in symmetry element grids are handled by the *FRAC suboption of *WELL. However, *FRAC usually accounts only for the fractional area normal to the well direction.

A fractional well length corresponding to a partial block must be specified via the *ff* option of *PERF *GEO or must be included in the well index entered via the *PERF *WI option. For standard areal symmetry elements, no block is partial in the vertical direction and so no completion fraction is needed for a vertical well. However, a horizontal well completed in a block that is partial in the well direction will have a completion fraction less than one.

For example, consider completing a well in the partial blocks in Figures 2 and 3. The wellbore enters the block from the right and runs horizontally to the block node. In the case of each figure, the completion length is DX instead of DX', so the completion fraction for this block is *ff* = 0.5.

Consider the grid specified by the keyword data in section **Null Blocks**, above. A horizontal well through blocks (1:9,1,1) would be specified as

```
*WELL wn 'Horz Well' *FRAC 0.5  ** On symmetry boundary
*PERF *GEO wn
    1 1 1  0.5   ** Partial block in I direction
  2:9 1 1
   10 1 1  0.5   ** Partial block in I direction
```

Permeabilities (Required) *PERMI, *PERMJ, *PERMK

PURPOSE:

*PERMI indicates input of an array of I direction permeabilities.

*PERMJ indicates input of array of J direction permeabilities.

*PERMK indicates input of an array of K direction permeabilities.

ARRAY:

*PERMI
*PERMJ
*PERMK

DEFAULTS:

Required keywords. No defaults. Units are (md | md), ALWAYS MILLIDARCIES.

CONDITIONS:

This keyword must appear in the RESERVOIR DESCRIPTION keyword group, but may also be specified in the RECURRENT DATA section for modelling hydraulic fractures

EXPLANATION:

Permeabilities are used to compute inter-block fluid flows and must be available for any direction for which fluid may enter or leave a block. Note that values must be available for both grid blocks in a communicating pair, as permeability averaging is carried out. Also, a block's permeabilities are used for both the flows to its high side neighbours as well as its low side. For instance, if block (I,J,K) has an I direction permeability "permi", this value enters into calculating the flow to block (I+1,J,K) as well as to block (I-1,J,K).

If a dual porosity model is being used, values for permeabilities are required for both *MATRIX and *FRACTURE. Matrix permeabilities are required for calculating the fluid flow between the matrix and fracture systems, and for computing matrix to matrix flow for *DUALPERM, while fracture permeabilities are required for calculating fracture to fracture flows between blocks.

*MATRIX permeabilities should be the values measured from a piece of unfractured matrix material, while *FRACTURE permeabilities should be entered as effective fracture permeabilities; that is, the permeability of the fracture system with respect to a volume of reservoir rock. Effective fracture permeabilities are what is usually measured during a well test or what would be computed from an open channel flow model with multiplication by a fracture porosity.

*PERMJ and *PERMK may be specified using the *EQUALSI array input option if *PERMI is already input. For example, for a dual porosity/dual permeability system:

```
** Specify horizontal permeability
*PERMI *FRACTURE *ALL
2500.   2200.  2150.   2300.   2200.
...
*PERMI *MATRIX *ALL
340.   315.   280.   260.   240.
...
** J direction permeabilities are equal to I
** direction values.
*PERMJ *EQUALSI
** Vertical permeability is one tenth of the
** horizontal permeability.
*PERMK *EQUALSI * 0.10
```

The example demonstrates that you can specify only the values for the I direction and then, if necessary, alter the values in the other directions or equate the I direction values to the remaining directions.

The acceptable range of values for permeability is:

	SI	**Field**
	md	**md**
min	0.0	0.0
max	1.0E+10	1.0E+10

Netpay (Optional) *NETPAY

PURPOSE:

*NETPAY indicates input of an array of net pay thicknesses which are converted internally to an array of net-to-gross multipliers.

ARRAY:

*NETPAY

DEFAULTS:

Optional keyword. The default net-to-gross multiplier is 1.0 for grid blocks that are not supplied with net pay values, or equivalently, net pay equals gross pay, the latter being defined by *DK or derived from corner point input.

In dual porosity models, net pay values can be applied to the matrix and fracture pore volumes separately by use of the *MATRIX and *FRACTURE qualifiers. However, the default for fracture multipliers is still 1.0, even if *NETPAY *MATRIX (or just *NETPAY) appeared, and the default for matrix multipliers is still 1.0, even if *NETPAY *FRACTURE appeared.

It is recommended that *NETPAY *MATRIX and *NETPAY *FRACTURE be used together, or not at all, for dual porosity models.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

Values for net pays must be non-negative, and may exceed the values input for gross pays.

EXPLANATION:

The keyword *NETPAY allows input of net thicknesses (m | ft) which are converted to net-to-gross ratios by dividing the input array values by the respective gross block thicknesses, the latter being obtained from *DK or corner point input. These net-to-gross ratios are then used to modify the porosities, and the permeabilities in the I and J directions (but not the K direction).

The net-to-gross ratios are used as multiplication modifiers as follows:

(a)	por is replaced by	por*ntg
(b)	permi is replaced by	permi*ntg
(c)	permj is replaced by	permj*ntg

where "por" denotes the grid block's porosity as set using the *POR keyword, "permi" and "permj" are the block's permeabilities as set using the *PERMI and *PERMJ keywords, and "ntg" is the net- to-gross ratio as derived from the net pay value assigned by this *NETPAY keyword divided by the appropriate thickness obtained from the *DK array or corner point input. These modifications are used internally and do not appear in the output.

Note that the permeability in the K direction is not altered. Transmissibility multipliers (see the *TRANSK keyword) are available for vertical flow adjustments due to full or partial barriers caused by shales or other geological features. Transmissibility multipliers can still be used to further alter the flows in the I and J directions.

The acceptable range of values for any derived net-to-gross ratio is:

	SI	**Field**
min	0.0	0.0
max	1.0E+4	1.0E+4

Net to Gross Multipliers (Optional) *NETGROSS

PURPOSE:

*NETGROSS indicates input of an array of net-to-gross multipliers.

ARRAY:

*NETGROSS

DEFAULTS:

Optional keyword. The default net-to-gross multiplier is 1.0.

In dual porosity models, net-to-gross multipliers can be applied to the matrix and fracture pore volumes separately by use of the *MATRIX and *FRACTURE qualifiers. However, the default for fracture multipliers is still 1.0, even if *NETGROSS *MATRIX (or just *NETGROSS) appeared, and the default for matrix multipliers is still 1.0, even if *NETGROSS *FRACTURE appeared.

It is recommended that *NETGROSS *MATRIX and *NETGROSS *FRACTURE be used together, or not at all, for dual porosity models.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. This keyword should not be used if *NETPAY was used (see above).

EXPLANATION:

The keyword *NETGROSS allows input of net-to-gross which are used to modify the porosities and permeabilities in the I and J directions. The net-to-gross ratios are used as multiplication modifiers as follows:

(a)	por is replaced by	por	*ntg
(b)	permi is replaced by	permi	*ntg
(c)	permj is replaced by	permj	*ntg

where “por” denotes the grid block’s porosity as set using the *POR keyword, “permi” and “permj” are the block’s permeabilities as set using the *PERMI and *PERMJ keywords, and “ntg” is the incoming net-to-gross ratio for the block set using *NETGROSS. These modifications are used internally and do not appear in the output.

Note that the permeability in the K direction is not altered. Transmissibility multipliers (see the *TRANSK keyword) are available for vertical flow adjustments due to full or partial barriers caused by shales or other geological features. Transmissibility multipliers can still be used to further alter the flows in the I and J directions.

The acceptable range of values is:

	SI	Field
min	0.0	0.0
max	1.0E+4	1.0E+4

Transmissibility Multipliers (Optional) *TRANSI, *TRANSJ, *TRANSK

PURPOSE:

*TRANSI indicates input of an array of I direction transmissibility multipliers.

*TRANSJ indicates input of an array of J direction transmissibility multipliers.

*TRANSK indicates input of an array of K direction transmissibility multipliers.

ARRAY:

*TRANSI
*TRANSJ
*TRANSK

DEFAULTS:

Optional keyword. Defaults: 1.0

In dual porosity models, transmissibility multipliers can be applied to both the matrix and fracture pore volumes separately by use of the *MATRIX and *FRACTURE qualifiers. The defaults for *TRANSI *MATRIX is 1.0, regardless of values set for *TRANSI *FRACTURE, and the same holds if *MATRIX and *FRACTURE are reversed. The same comments hold for *TRANSJ and *TRANSK.

Transmissibility multipliers may be altered in recurrent (well) data. Multipliers that are altered will take on their assigned values, while all others will retain their existing values; that is, the values set by any appearance of *TRANSI, *TRANSJ, and *TRANSK keywords in the RESERVOIR DESCRIPTION, other values being defaulted to 1.0

CONDITIONS:

These keywords may be in the RESERVOIR DESCRIPTION keyword group. They may also appear in recurrent (well) data.

The application of these multipliers depends upon *TRANSMULT.

EXPLANATION:

Flow between grid blocks is proportional to a cross-sectional inter-block flow area, an averaged permeability value, and a divisor equal to the inter-block distance. These terms combine to form a transmissibility which is calculated in the simulator. Before it is used, a transmissibility multiplier, as set by a *TRANSI, *TRANSJ or *TRANSK in conjunction with *TRANSMULT keyword, is applied.

The multiplier defaults to 1.0, equivalent to applying no multiplier, if no data is present. Transmissibility multipliers are dimensionless.

*TRANSJ and *TRANSK may be specified using the *EQUALSI array input option, provided that a *TRANSI array is entered first.

Since two blocks enter into any inter-block flow calculation, a method is required for deciding which block will contribute the multiplier.

Inter-block flow between blocks on a single grid:

This rule applies whether the grid is the (main) fundamental grid (grid 1), or any refined grid. If flow between a pair of blocks is considered, and they both lie on the same grid, then it is the block with the lowest I index for an I direction pair, or the lowest J index for a J direction pair, or the lowest K index for a K direction pair, that supplies the multiplier; that is, a directional multiplier applies to a block's interface with its neighbour with the higher index in that direction. These rules apply even when faults are present (see *FAULT following). Note that fault considerations only affect lateral (I and J direction) calculations.

This rule is altered for *GRID *RADIAL and *GRID *HYBRID when connecting block nj to block 1 in direction J when nj exceeds 1; that is, when a subdivided ring is being closed. In this case, the multiplier from block nj is used for the closure. Also, flow perpendicular to the wellbore in *HYBRID grids uses a multiplier averaged over the two directions perpendicular to the well.

Note that refined grids inherit the multipliers from their parent block, unless special values are read for the refined grid directly (*RG qualifier).

Note that except for the special case of zero transmissibility multipliers, all refined blocks in a locally refined block inherit the multipliers from their parent block. Interior blocks and those at a refined grid – refined grid interface would inherit the parent's multipliers.

In the special case of a zero multiplier, the multiplier is inherited only by those refined blocks on the appropriate interface.

If the user needs to modify transmissibilities of an interface to a value other than zero when refined grids are involved, the use of the *RG keyword is required to explicitly refer to refined blocks at the interface.

Flow between a refined grid and the fundamental:

Basically the same rules apply as for fundamental blocks, except when determining the I, J, or K index of a refined block at a refined block fundamental block interface, refer to its parent's I, J, or K index. If the refine blocks parent has the lowest I (J, or K) index then the multiplier or the refined block is used. If the adjoining fundamental block has the lowest index then the multiplier of the adjoining fundamental block is used. This also applies to the *TRANLI (J, K) keywords except that the fundamental block with the highest index is used.

Use of this rule and *TRANLI (J, K) make it possible to only refer to fundamental blocks when defining non zero transmissibility multipliers between refined and fundamental blocks.

Flow between two refined grids:

Again the same rules apply. Refer to the I, J, or K index of the parent blocks for both refined blocks.

Dual porosity models:

*MATRIX transmissibilities are applied to matrix to fracture flows for all double porosity models, except for *DUALPERM (when no such multipliers exist).

If a *DUALPERM model is being used, the *MATRIX transmissibility multipliers are used for modifying inter-block matrix to matrix flow in the same manner that single porosity multipliers operate. There are no matrix to fracture flow multipliers available for this case.

It is the I and J direction multipliers that are often zeroed with the *DUALPERM model, leaving the K direction multipliers nonzero. This choice is made because the most important dual permeability effects are usually in the vertical direction, arising due to phase density differences.

Examples:

The following provides an example of standard transmissibility multiplier usage:

```
** Specify horizontal transmissibility multipliers
*TRANSI *FRACTURE *ALL
1.4  2*1.2 1.4  1.5  1.4
...

*TRANSI *MATRIX *ALL
1.2  1.3   1.4   1.1  1.2  1.4
...

*TRANSJ *EQUALSI

** Vertical transmissibility is one tenth of the
** horizontal transmissibility.
*TRANSK *EQUALSI * 0.10
```

Suppose block (2,2,2) contains a 3 x 2 x 1 refined grid. Then I direction transmissibility multipliers can be applied to flows going in or out of the refined grid as follows:

```
*TRANSI *RG 1 1 1 *ALL
.8 1 .8  .8 1 .8
```

The acceptable range of values for transmissibility multipliers is:

	SI	Field
min	0.0	0.0
max	1000.0	1000.0

Diagonal Transmissibility Multipliers (Optional) *TRANSIJ+, *TRANSIJ-, *TRANSIK+, *TRANSIK-

PURPOSE:

Specify transmissibility multipliers in the diagonal grid directions (*NINEPOINT only).

ARRAY:

*TRANSIJ+
*TRANSIJ–
*TRANSIK+
*TRANSIK–

DEFAULTS:

If the keyword does not appear, the corresponding multiplier is assumed to be one. If a keyword appears but a block is not referenced, its corresponding multiplier is one.

CONDITIONS:

Keywords *TRANSIJ+, *TRANSIJ–, *TRANSIK+ and *TRANSIK– must be located in the Reservoir Description section or in recurrent data.

*TRANSIJ+ and *TRANSIJ– are applicable only with *NINEPOINT *IJ, and indicate the diagonal directions I+J+ and I+J–, respectively. Similarly, *TRANSIK+ and *TRANSIK– are applicable only for *NINEPOINT *IK to indicate directions I+K+ and I+K–, respectively.

EXPLANATION:

See the explanation for *TRANSI for general comments on transmissibility multipliers.

Transmissibility multipliers can be changed at any time in recurrent data.

Diagonal transmissibility multipliers are used only with the nine-point option, to modify the flow for diagonal interblock connections as follows.

This keyword	Refers to this direction	Between This Cell	And This Cell	For *NINEPOINT
*TRANSIJ+	I+J+	(i,j,k)	(i+1,j+1,k)	*IJ
*TRANSIJ–	I+J–	(i,j,k)	(i+1,j–1,k)	*IJ
*TRANSIK+	I+K+	(i,j,k)	(i+1,j,k+1)	*IK
*TRANSIK–	I+K–	(i,j,k)	(i+1,j,k–1)	*IK

An interblock connection in a diagonal direction has a transmissibility calculation that is similar to a normal five-point method, but the effective cross-sectional area and internode separation are particular to the nine-point formulation.

Flow Barrier in Nine-Point Plane

You can use transmissibility multipliers to specify a fluid-flow barrier normal to the nine-point plane, but you must take care with addressing cells for diagonal directions.

For example, suppose that you have a 10x10x5 grid with standard five-point discretization. To specify a fluid-flow barrier between grid planes K=3 and K=4 (over the entire I-J area) you would need only *TRANSK:

```
*TRANSK *IJK 1:10 1:10 3  0.0
```

Note that the multiplier is assigned using cell address K=3 since *TRANSK refers to the connection between K and K+1.

Now suppose that we apply *NINEPOINT *IK to the same grid, that is, nine-point discretization is done in the I-K plane. To implement a fluid-flow barrier between planes K=3 and K=4 you would need additional data:

```
*TRANSIK+ *IJK 1:10 1:10 3  0.0
*TRANSIK- *IJK 1:10 1:10 4  0.0
```

Note that the *TRANSIK+ multiplier is assigned using cell address K=3 since *TRANSIK+ refers to the connection between K and K+1. On the other hand, the *TRANSIK– multiplier is assigned using cell address K=4 since *TRANSIK– refers to the connection between K and K–1.

Similar comments apply to *TRANSIJ+ and *TRANSIJ– when constructing a fluid-flow boundary between two J planes when *NINEPOINT *IJ is used.

Transmissibility Multipliers for Lower Indexed Cell Blocks (Optional) *TRANLI, *TRANLJ, *TRANLK

PURPOSE:

*TRANLI indicates input of an array of I direction transmissibility multipliers for faces contacting lower indexed blocks.

*TRANLJ indicates input of an array of J direction transmissibility multipliers for faces contacting lower indexed blocks.

*TRANLK indicates input of an array of K direction transmissibility multipliers for faces contacting lower indexed blocks.

ARRAY:

*TRANLI
*TRANLJ
*TRANLK

DEFAULTS:

Optional keyword. Defaults: 1.0

CONDITIONS:

These keywords may be in the RESERVOIR DESCRIPTION keyword group or they may be in recurrent (well) data.

The application of these multipliers depends upon *TRANSMULT.

EXPLANATION:

Flow between grid blocks is proportional to a cross-sectional inter-block flow area, an averaged permeability value, and a divisor equal to the inter-block distance. These terms combine to form a transmissibility which is calculated in the simulator. Before it is used, a transmissibility multiplier is applied. The multiplier can be set using the *TRANSI, *TRANSJ, or *TRANSK keywords (as described elsewhere) or the *TRANLI, *TRANLJ, or *TRANLK keywords described here.

All transmissibility multipliers are dimensionless.

Transmissibility multipliers can be specified for any grid block. A default value of 1.0 will be used for unspecified multipliers.

When transmissibility multipliers appear in recurrent (well) data, any block's multipliers may be (re–)assigned, BUT unreferenced blocks retain their values assigned earlier (which will be 1.0's if no other values were ever assigned).

Since two blocks enter into any inter-block flow calculation, a method is required for deciding how blocks will contribute multipliers.

If flow between a pair of blocks is considered, it is the block with the highest I index for an I direction pair, or the highest J index for a J direction pair, or the highest K index for a K direction pair, that supplies multiplier values set by the *TRANLI, *TRANLJ, or *TRANLK keywords, respectively. This behavior is the opposite of the assignment of multipliers based on the *TRANSI, *TRANSJ, or *TRANSK keywords, where the lower indexed block in the pair supplies the multiplier.

If both types of multipliers have been defined for a face, one coming from a *TRANLI, *TRANLJ or *TRANLK value assigned to the higher indexed block, and a *TRANSI, *TRANSJ or *TRANSK value assigned to the lower indexed block, then the following rules are applied, in the order shown, to determine the final transmissibility:

1. if both values are 1, then no modification is performed (multiplier is 1);
2. if either value is 0, then no fluid flow is allowed (multiplier is 0);
3. if one value is 1, and the other is not 1, then modification is based on the non-unity value (multiplier is the non-unity value);
4. if both values are not 1, then the arithmetic average of the two values is used (multiplier is the average of the two values).

Thus, setting a zero *TRANLI, *TRANLJ, *TRANLK, *TRANSI, *TRANSJ, or *TRANSK cuts off all fluid flow at a face.

These multipliers can be used to control flow between refined grids, or from the fundamental grid to a refined grid. These multipliers apply even when faults are present (see *FAULT following). Faults only use the *TRANLI and *TRANLJ multipliers.

The rules of how these “low side” multipliers apply to refined blocks is essentially the same as those for *TRANSI (J, or K). The user should always refer to the I, J, or K indices of the parent blocks and apply the same rules as he would on a fundamental grid to determine which blocks multipliers are used.

These multipliers have no effect on flow between matrix and fracture in dual porosity models (*DUALPOR and *DUALPERM). *MATRIX flow values are used for matrix to matrix flow between different blocks in a DUAL PERMEABILITY model.

*TRANLI, *TRANLJ, or *TRANLK should not be used with *HYBRID grids.

*TRANLJ and *TRANLK may be specified using the *EQUALSI array input option, providing that *TRANLI is entered first.

See also the previous description for the *TRANSI, *TRANSJ and *TRANSK keywords.

The acceptable range of values for transmissibility multipliers is:

	SI	**Field**
min	0.0	0.0
max	1000.0	1000.0

Transmissibility Multiplier for Matrix-Fracture Flow (Optional)

***TRANSMF**

PURPOSE:

*TRANSMF specifies transmissibility multiplier for fluid flow between matrix and fracture.

ARRAY:

*TRANSMF

DEFAULTS:

For each spatial block for which *TRANSMF is not specified, the multiplier is 1.

CONDITIONS:

This keyword may appear in both the Reservoir Description section and Well and Recurrent Data section.

Array qualifiers *MATRIX and *FRACTURE are not allowed.

The application of these multipliers depends upon *TRANSMULT.

EXPLANATION:

The *TRANSMF keyword specifies a single multiplier per spatial cell that is applied to the flow of fluid between the matrix block and adjacent fracture block in that cell in a dual porosity setting. Because *TRANSMF applies to the spatial cell, array qualifiers *MATRIX and *FRACTURE are not allowed.

All transmissibility multipliers are dimensionless.

When transmissibility multipliers appear in recurrent data, any block's multiplier may be (re–)assigned, BUT unreferenced blocks retain their values assigned earlier (which will be 1.0's if no other values were ever assigned).

A zero transmissibility multiplier cuts off all fluid flow between the affected blocks.

Transmissibility Multiplier Action (Optional) *TRANSMULT

PURPOSE:

Specify action to take for successive transmissibility multipliers.

FORMAT:

*TRANSMULT (*REPLACE | *ACCUMULATE)

DEFINITIONS:

*REPLACE
The current transmissibility multiplier for an interblock connection is replaced by the value specified via input data.

*ACCUMULATE
The current transmissibility multiplier for an interblock connection is multiplied by the value specified via input data.

DEFAULTS:

If *TRANSMULT is absent, *REPLACE is assumed.

CONDITIONS:

Keyword *TRANSMULT may appear in the Reservoir Description and the Recurrent Data sections.

EXPLANATION:

The transmissibility used in a fluid flow equation is $T_{base} \cdot M_{save}$. T_{base} is the base value of transmissibility from the input geometry and absolute permeability; permeability may vary during the run because of dilation, etc. M_{save} is an arbitrary multiplier that is saved separately for each interblock connection. At the start of data input each connection's M_{save} is 1.

The following keywords specify new multipliers M_{input}:

*TRANSI, *TRANSJ, *TRANSK
*TRANLI, *TRANLJ, *TRANLK
*TRANSMF
*TRANSD

Keyword *TRANSMULT specifies what action is taken to get M_{save} from M_{input}.

Subkeyword	**New M_{save} is**
*REPLACE	M_{input}
*ACCUMULATE	M_{input} times old M_{save}

For example, assume that *TRANSI specifies a value of 2 early in the run and another *TRANSI specifies a value of 5 later in the run, for the same connection. The multiplier value after the second *TRANSI will be 5 for *REPLACE and $5 \cdot 2 = 10$ for *ACCUMULATE.

Because a run starts with M_{save} = 1, the two *TRANSMULT actions give the same result for the first change of a connection's M_{save}. The two actions give different results only for subsequent changes to a connection's M_{save}.

Keyword *TRANSMULT may be used any number of times in a data set to switch between actions *REPLACE and *ACCUMULATE. Data reading starts with action *REPLACE in effect. Whenever a transmissibility multiplier keyword is encountered, the multiplier is interpreted according to the current *TRANSMULT action in effect.

Because a run starts with M_{save} = 1, the two *TRANSMULT actions give the same result for the first change.

Example:

The following data changes the *TRANSMULT action several times.

```
*GRID *CART 10 1 10
. . .
*TRANSI *IJK 2:8 1 3  2.0
*TRANSI *IJK 2:8 1 3  2.5     ** Multiplier is 2.5
. . .
*RUN
. . .
*TIME 100.
   *TRANSMULT *ACCUMULATE
   *TRANSI *IJK 2:8 1 3  2.0
*TIME 200.
   *TRANSI *IJK 2:8 1 3  5.0  ** Multiplier is 25.
*TIME 300.
   *TRANSMULT *REPLACE
   *TRANSI *IJK 2:8 1 3  1.0  ** Reset multipliers
*TIME 400.
   *TRANSMULT *ACCUMULATE
   *TRANSI *IJK 2:8 1 3  5.0  ** Multiplier is 5.
*TIME 500.
   *TRANSI *IJK 2:8 1 3  3.0  ** Multiplier is 15.
```

Transmissibility Multiplier Regions (Optional)

***TRANS_MULT_REGION**

PURPOSE:

*TRANS_MULT_REGION specifies transmissibility multiplier regions to be used with *INTER_REGION_TM transmissibility multipliers

ARRAY:

*TRANS_MULT_REGION

DEFAULTS:

There is no default, if this option is used every block must be given a region number.

CONDITIONS:

This keyword must appear in the Reservoir Description section. Region numbers are positive integers which must be greater than or equal to 1. A value of 0 is not allowed.

Array qualifiers *MATRIX and *FRACTURE are allowed.

EXPLANATION:

The *TRANS_MULT_REGION keyword specifies region numbers. Flow between blocks in different regions will have their transmissibilities multiplied by the transmissibility multiplier specified on the *INTER_REGION_TM keyword.

Inter Region Transmissibility Multiplier (Optional)

***INTER_REGION_TM**

PURPOSE:

*INTER_REGION_TM specifies that the transmissibility at the interface between different *TRANS_MULT_REGION regions (see TRANS_MULT_REGION keyword) should be multiplied by a constant.

FORMAT:

*INTER_REGION_TM	iregion1	iregion2	tmult
	iregion1	*ALL	

DEFINITIONS:

tmult

The value of the inter region multiplier. The acceptable range is greater than or equal to 0.0. This multiplier is applied in addition to all other multipliers. A tmult of 0.0 actually eliminates the connection between blocks.

*ALL

*ALL means that flow between region iregion1 and all adjoining regions will have flow between those regions (at the interface between those regions) altered by the value of tmult entered on the row.

iregion1

First region defining the interface between regions. If *ALL is read after iregion1 (instead of iregion2) it is assumed that the following transmissibility multiplier is used for all regions which bound iregion1

iregion2

Second region defining the interface between regions.

DEFAULTS:

No Defaults

CONDITIONS:

Optional keyword. It must appear in the Reservoir Description Section. Region numbers must be greater than 0. Refined blocks can be defined using *TRANS_MULT_REGION or allow the grid module to do the inheritance (default). Naturally fractured grids are supported. Matrix and fracture cells can be assigned to separate regions.

EXPLANATION:

The use of *INTER_REGION_TM and *TRANS_MULT_REGION allow the user to control flow between user defined *TRANS_MULT_REGION regions. Iregion1 and iregion2 can be any region defined in the *TRANS_MULT_REGION array. The *ALL syntax can be used to represent flow between iregion1 and any other region, which may be used to completely isolate a region.

Note that conductive heat transfer is not affected by these multipliers.

When defining *INTER_REGION_TM multipliers, multiple definition of regions are allowed, the last definition always overrides/replaces any previous inter-region multiplier definitions.

Example 1:

```
** Inter Region multiplier for region 1 and 2 is 0.0,
** the Inter Region multiplier for all other regions
** and region 1 is 0.1

*INTER_REGION_TM  1  *ALL    0.1
                  1  2       0.0
```

Example 2:

```
** Inter Region multiplier for region 2 and other regions is 0.25,
** The Inter Region Multiplier for region 1 and all regions
** except 2 is 0.1,
*INTER_REGION_TM  1  *ALL    0.1
                  2  *ALL    0.25
```

Example 3:

```
** Inter Region multiplier for region 4 and all regions is 0.0
** The first Inter Region multiplier is overridden

*INTER_REGION_TM  2  4    0.1  ** overridden by next line
                  4  *ALL 0.0
```

Example 4:

```
** Inter Region multiplier for region 2 and all regions is 0.0
** The first Inter Region multiplier is overridden
*INTER_REGION_TM  2  4    0.1  ** overridden by next line
                  2  *ALL 0.0
```

Transmissibility Values (Optional) *TRANVI, *TRANVJ, *TRANVK

PURPOSE:

Specify transmissibility values in the various positive directions.

ARRAY:

*TRANVI
*TRANVJ
*TRANVK

DEFAULTS:

Optional keywords. If a keyword is absent for a block, the corresponding transmissibility value is calculated internally.

In dual porosity models, transmissibility values can be applied to the matrix and fracture pore volumes separately via array qualifiers *MATRIX and *FRACTURE.

CONDITIONS:

These keywords may appear in the Reservoir Description data section as well as the Well and Recurrent Data section.

Transmissibility value (*TRANVI, etc.) and variable-permeability models (e.g., compaction and dilation) should not be used together for the same block. Variable-permeability models update block permeabilities and subsequent interblock transmissibilities during the run. Any static transmissibility changed via *TRANVI, etc., may be overwritten by a variable-permeability model.

EXPLANATION:

Normally interblock fluid-flow transmissibility is calculated internally from grid-based geometry parameters and permeabilities. Keywords *TRANVI, *TRANVJ and *TRANVK allow you to over-ride these internal calculations with user-specified values.

The unit of transmissibility is permeability times length, i.e., [md-m | md-ft]. For example, if input permeability unit is “md” and input length unit is “ft”, then input transmissibility unit is “md-ft”.

Inter-block Flow in Same Grid

Generally, grid-array keywords apply a value to a block with (I,J,K) indices in a fundamental grid or a locally refined grid. Keyword *TRANVI applies a transmissibility value to the interblock connection between block (I,J,K) and block (I+1,J,K), and keywords *TRANVJ and *TRANVK act similarly in their respective directions. This implies that values specified at the upper index boundaries (e.g., I = NI) are not used.

For a radial grid (*GRID *RADIAL or *GRID *HYBRID) whose angular (J) direction defines a full closed circle, a *TRANVJ value for J=NJ is applied to the interblock flow between blocks (I,NJ,K) and (I,1,K).

This adjacent-index rule applies even when faults are present (see the description for keyword *FAULT). For example, neither *TRANVI nor *TRANVJ can refer to the fault-created interblock connection between (I,J,K) and (I+1,J+1,K).

Inter-block Flow in Different Grids

Transmissibility values may not be specified between blocks in two different grids (fundamental grid and locally refined grid, or two different locally refined grids). Use *IRCONNECT or *SCONNECT to modify the internal transmissibility calculation.

Natural Fracture models

Array qualifier *FRACTURE specifies transmissibility value for Fracture-Fracture flow in Dual Porosity and Dual Permeability models.

Array qualifier *MATRIX specifies transmissibility value for Matrix-Matrix flow of Dual Permeability models. Note that Dual Porosity models have no Matrix-Matrix flow in fractured blocks, so specifying transmissibility values with *MATRIX will have no effect for fractured blocks.

Zero Transmissibility Value

For an isothermal run, specifying a transmissibility value of zero will remove the interblock connection completely. Specifying a non-zero value later in the run will restore that interblock connection.

For a thermal run (see *THERMAL *ON in Component Properties Section), specifying a transmissibility value of zero will prevent interblock fluid flow but will not prevent interblock heat conduction. Specifying a non-zero value later in the run will restore that interblock fluid flow.

Interaction with Transmissibility Multipliers

Care must be taken when specifying both value (e.g., *TRANVI) and multiplier (e.g., *TRANSI) for the same connection. Non-zero value and multiplier will work together, that is, the specified value is the base transmissibility value which is multiplied by the specified multiplier at run time. In addition, these two data may be re-specified independently later in recurrent data.

However, a specified non-zero value cannot restore an interblock connection that was removed via a specified zero multiplier.

Examples:

The following provides an example of direct transmissibility input. Here, transmissibilities values are specified for the I and J directions, whereas for the K direction a multiplier is applied with the K-direction internal calculations.

```
** Specify horizontal transmissibility multipliers
*TRANVI *FRACTURE *ALL
1.4  2*1.2 1.4  1.5  1.4
...
*TRANVI *MATRIX *ALL
1.2  1.3   1.4   1.1  1.2  1.4
...
*TRANVJ *EQUALSI
** Vertical transmissibility is one tenth of the
** horizontal transmissibility.
*TRANSK *EQUALSI * 0.10
```

Suppose block (1,1,1) contains a 3 x 2 x 1 refined grid. Then I-direction transmissibility can be applied to fluid flow within the refined grid as follows.

```
*TRANVI *RG 1 1 1 *ALL
.8 1 .8  .8 1 .8
```

Faults (Optional) *FAULT

PURPOSE:

*FAULT indicates the input of designations for grid blocks whose flow connections are to take into account their vertical position in the reservoir with respect to lateral neighbours.

Each *FAULT keyword describes a group of grid blocks that together form a geological fault block. Edge members of fault blocks will receive special consideration when flows are calculated.

FORMAT:

```
*FAULT    throw    i1:i2    j1:j2
                     :        :
```

DEFINITIONS:

throw

Geologically speaking, "throw" is the difference in depth between a geological fault block and neighbouring reservoir rock. (m | ft).

In the simulator, throws provide modifications to depth data given earlier through use of the *DEPTH, *DTOP, *DEPTH-TOP or *PAYDEPTH keywords. A zero throw is valid if the depth information is correct and only the identification of the grid blocks involved in the geological fault block is required.

i1:i2 j1:j2

The indices, i1, i2, j1, and j2 locate grid block columns whose first index (I index) lies between i1 and i2 inclusive, whose second index (J index) lies between j1 and j2 inclusive, and whose third index (K index) lies between 1 and nk inclusive.

DEFAULTS:

Optional keyword. Default: no faults.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

These keywords should not be used with *GRID *CORNER. Fault data can be entered directly through corner positioning with these grids, as was discussed earlier.

EXPLANATION:

Geologic faults are formed when a portion of the reservoir is dislocated with respect to another portion. These dislocated portions form geological fault blocks. Lateral flow between fault blocks does not follow the usual geological strata. To take this into account when modelling a reservoir, it is necessary to be able to group grid blocks into fault blocks, and to take account of these fault blocks when developing inter-block communication.

The fault model described here assumes that each geologic fault block can be described by the grid blocks in a collection of grid block ranges. Each *FAULT keyword begins the description of a fault block. Member grid blocks are listed by the I's and J's, and the list is assumed ended when another keyword is found, which may be another *FAULT keyword.

Note that fault blocks extend vertically throughout the entire reservoir.

A "throw" value can be applied to the depths of all blocks in a fault block. Throws can be of any sign, or 0.0. They will be added directly to the already existing depth values. Thus, comments given earlier regarding depth measurements (see *DEPTH, *DTOP, *DEPTH-TOP and *PAYDEPTH keywords) will apply. If a full and correct depth array was introduced earlier (using the *PAYDEPTH option, for instance), the throw values can be set to 0.0. (If a grid block is assigned to more than one fault block, throws will accumulate.)

When it comes time to compute transmissibilities for lateral inter-block flow, and fault blocks are present, special checking is carried out. If lateral flow into grid block (I,J,K) from its positive I-direction is being considered, and block (I,J,K) is on the edge of some fault block that was defined using a *FAULT keyword, then a search for overlap is made with respect to all its high-side neighbours (I+1,J,KK) for KK=1,...,NK. Any neighbour that physically overlaps with block (I,J,K) will be connected to it. Note that connections other than the standard connection to (I+1,J,K) are possible. Multiple connections may even result.

A similar search is carried out when block (I,J,K) has not been marked as being on the edge of a fault block, but some of the neighbours in the (I+1,J,KK) stack are.

The J direction is treated in a similar fashion.

Vertical transmissibility calculations are not affected by these fault considerations. Neither are flows between blocks that are members of the same fault block.

The acceptable range of values for throws area:

	SI	**Field**
	m	**ft**
min	-1.0E+3	-3,280.0
max	1.0E+3	3,280.0

Aquifers (Optional) *AQUIFER, *AQPROP, *AQMETHOD

PURPOSE:

*AQUIFER indicates the input of aquifer location information.

*AQPROP indicates the input of aquifer properties.

*AQMETHOD selects the numerical method to be used for calculating water influx from the aquifer.

*AQLEAK determines the ability of water to leak back into the aquifer when under the influence of a higher reservoir pressure.

FORMAT:

```
*AQUIFER    (*BOUNDARY)
            (*RESBND)
            (*BOTTOM)
            (*REGION)      i1:i2 j1:j2 k1:k2      (*IDIR)
                                                  (*JDIR)
                                                  (*KDIR)
*AQPROP
Thickness Porosity Permeability Radius Angle (R-Ratio)
*AQMETHOD   (*CARTER-TRACY)
            (*FETKOVITCH)
            (*OLD)
*AQLEAK     (*ON)
            (*OFF)
            (*OLD)
```

DEFINITIONS:

*BOUNDARY

The aquifer is to be connected to all fundamental grid edge blocks (either I equals 1 or ni, or J equals 1 or nj, or both) in the appropriate I or J direction (ni = last block in x direction, nj = last block in y direction). Aquifer connections are not made through null blocks, so a reservoir boundary that is not at the active grid block boundary will not get an aquifer connection.

*RESBND

The behavior of *AQUIFER *BOUNDARY is to place each aquifer connection on the grid boundary; if a grid boundary block is null then no aquifer connection is made. *AQUIFER *RESBND places aquifer connections on the reservoir boundary instead. The reservoir boundary is defined as the first non-null block encountered when scanning in from the grid boundary. *AQUIFER *BOTTOM already operates in this way through pinched out blocks.

*BOTTOM

The aquifer is to be connected to the bottom of the reservoir; that is, the bottom of all fundamental grid blocks in the deepest layer. Aquifer connections are made through pinched-out (zero thickness) blocks but are not made through null blocks with non-zero thickness.

*REGION

The aquifer is to be connected to the region indicated by the following range (i1:i2 j1:j2 k1:k2) and direction keywords (*IDIR, *JDIR or *KDIR) keywords.

i1:i2

Indicates the I beginning and ending indices of the fundamental grid region where the aquifer is to be connected.

j1:j2

Indicates the J beginning and ending indices of the fundamental grid region where the aquifer is to be connected.

k1:k2

Indicates the K beginning and ending indices of the fundamental grid region where the aquifer is to be connected.

*IDIR

The direction of inflow between the aquifer and the reservoir boundary is to be in the I direction.

*JDIR

The direction of inflow between the aquifer and the reservoir boundary is to be in the J direction.

*KDIR

The direction of inflow between the aquifer and the reservoir boundary is to be in the K directions.

thickness

Aquifer thickness (m | ft).

porosity

Aquifer porosity (fraction).

permeability

Aquifer permeability (md | md).

radius

Effective reservoir radius (m | ft).

angle

Aquifer angle of influence (expressed as a fraction of a circle).

R-Ratio

Ratio of the aquifer's external radius to that the reservoir's effective radius (fraction).

*CARTER-TRACY

Calculate water influx from the aquifer using the Carter-Tracy approximation.

*FETKOVITCH

Calculate water influx from the aquifer using a formulation based on work by Fetkovitch.

*AQMETHOD *OLD

Selects the aquifer calculation for simulator versions numbered "97.xx" and earlier.

*AQLEAK

Indicates whether or not water is allowed to leak from the reservoir into the aquifer at a block where the block pressure exceeds the adjacent aquifer pressure. The behavior of the aquifer is modelled more accurately when *ON follows *AQLEAK, and leakage is allowed, rather than when *OFF is used, when no leakage is allowed. If *AQLEAK is not specified, this is equivalent to *AQLEAK *OFF.

*AQLEAK *OLD

Selects the aquifer leak off calculation for simulator versions numbered "98.xx" and earlier. The *OLD method, in some instances allowed aquifer leakage even when leakage was not allowed. *AQLEAK *OFF (the default) does not under any condition allow aquifer leakage when none was specified.

DEFAULTS:

Optional keywords. No defaults for *AQUIFER.

Defaults are assigned as follows:

Thickness:

average thickness for *BOUNDARY, *RESBND, and *REGION

-or-

square root of the contact area (see below), for *BOTTOM

Porosity:

average porosity

Permeability:

average reservoir permeability, averaged over the aquifer flow directions

Radius:

the radius of the circle whose circumference, which when multiplied by the Thickness, gives the contact area, for *BOUNDARY, or *RESBND -or- the square root of the (typical) area of a side of the reservoir, divided by pi (see below), for *BOTTOM -or- the square root of the contact area divided by pi, for *REGION

Angle:

full circle (1.0) for all grids except *RADIAL, when the sum of the outer ring angular extents is used (after division by 360.0), for *BOUNDARY,

*RESBND, and *REGION -or- angular extent of the bottom of the reservoir (after division by 360.0; see below), for *BOTTOM

R-Ratio:

defaults to 100.0 for *AQMETHOD *FETKOVITCH

That is:

$$\frac{(\text{aquifer external radius})}{(\text{reservoir effective radius})} = 100.0$$

In the above, "average" refers to a pore-volume weighted average taken over aquifer-contacting cells, and "contact area" means the sum of the areas of all cell faces that are defined to contact the aquifer, as specified by the *AQUIFER keyword.

*AQMETHOD defaults to *CARTER-TRACY using a dimensionless pressure influence function for an infinite aquifer.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

The aquifer modelling described here allows water influx (and reverse flow if *AQLEAK *ON is enabled) to a reservoir from an aquifer, or several aquifers. Using an analytical aquifer model can be more economical for simulation purposes than using grid blocks filled with water. If great accuracy is required in modelling aquifers however, water filled blocks should be used.

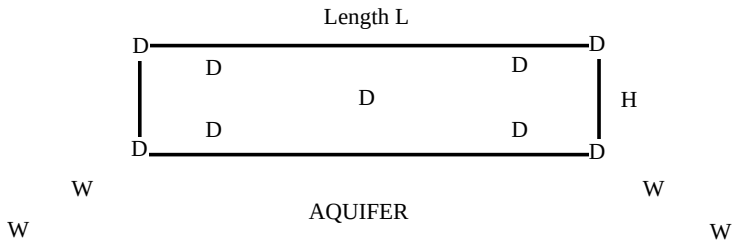
Since outflow from the reservoir into the aquifer is modelled analytically when *AQLEAK *ON is used, it is possible to have water outflow from a grid block attached to the aquifer that has no mobile water. Thus, care must be exercised when using analytical aquifer models. It is recommended that aquifers be modelled using grid blocks in cases where the reservoir pressure is expected to increase significantly during the course of the simulation.

When dual porosity models are used, the aquifers will be connected to the fracture porosity of grid blocks whenever possible. Otherwise, matrix porosity connections will be used.

The subkeyword *REGION indicates direct input of the aquifer boundary blocks. The subkeywords *IDIR, *JDIR, and *KDIR indicate which face of the reservoir is in contact with the aquifer. If more than one face is in contact with the aquifer, list each region on subsequent lines.

*AQPROP specifies the aquifer properties for calculation of water influx from the aquifer. *AQPROP defaults for *BOUNDARY or *RESBND aquifers envision a cylindrical reservoir with the aquifer contacting the reservoir around the full cylinder. The defaults are constructed so that the average thickness multiplied by the circumference gives the contact area calculated using the *AQUIFER keyword.

*AQPROP defaults for *BOTTOM aquifers envision a square contact area with a side dimension of L. Thus, L is equal to the square root of the bottom contact area. An average reservoir thickness H is also used in these calculations. The aquifer is assumed to be bounded by the edges of a wedge coming up to the bottom of the reservoir, as shown in the following side view of the reservoir. The "D"'s define diagonals drawn through the corners and their extensions, the "W"'s, define the wedge:



where the wedge angle is 2*atan(L/H). Then, the theory for radial aquifers is applied in this side view, with the value for Angle taken as 2*atan(L/H)/360, Thickness taken as L (viewed as a dimension into the page), and Radius taken as the square root of (L*H/pi). This latter choice is required as the reservoir must be circular to apply the theory. This defines defaults for *AQUIFER *BOTTOM.

*AQPROP defaults for *REGION aquifers are much like those for *BOUNDARY except that Radius is taken to be the square root of the contact area divided by pi.

If *AQMETHOD *CARTER-TRACY is specified, the simulator calculates water influx using a Carter-Tracy approximation. For more information, refer to R. D. Carter and G. W. Tracy, "An Improved Method for Calculating Water Influx", Trans., AIME, Vol. 219, (1960), 415-417. This method uses a dimensionless pressure influence function P(td), expressed as a function of dimensionless time td. The function is defined for using a table (see *AQFUNC below), along with an extrapolation method for dimensionless times that go beyond the end of the table. If the internal infinite extent aquifer table is used, an analytical expression is used for the extrapolation (see the Van Everdingen and Hurst reference mentioned in the *AQFUNC section). Otherwise, linear extrapolation in dimensionless time is used, which is appropriate for finite aquifers.

If *AQMETHOD *FETKOVITCH is specified, the simulator calculates water influx from the aquifer using a formulation based on work by Fetkovitch (see, for instance, M. J. Fetkovitch, "A Simplified Approach to Water Influx Calculations - Finite Aquifer Systems", JPT, July 1971, 814-828). This approach is able to model finite aquifers without having to generate dimensionless pressure functions. Only a "R-Ratio" parameter is required.

If *AQMETHOD *OLD is specified, older calculations (from versions numbered "97.xx" and earlier) are used. These versions used Carter-Tracy with a linear extrapolation for the dimensionless aquifer pressure influence function. When the default infinite extent aquifer function was being used, the linear extrapolation could result in an aquifer pressure drop that was too large, which could result in an aquifer that could seem limited in extent. As noted above, a standard analytical expression in ln(td) is now used for large time extrapolation of the infinite aquifer case. Some modifications of the defaults for *AQPROP parameters are also introduced when *AQMETHOD *OLD is specified.

Block transmissibility multipliers affect the aquifer inflow. *BOUNDARY aquifers for *RADIAL grids use *TRANSI multipliers. It the aquifer is of *BOTTOM type with layer NK generally deeper than layer 1, then *TRANSK-defined multipliers are used. If layer 1 is deeper than layer NK, then *TRANLK-defined multipliers are used. If the aquifer is of *BOUNDARY or *REGION type, and I-direction contact is made on a cell with an I-index I=1, then a *TRANLI-defined value is used. Similarly, if I-direction contact is made on a cell with an I-index I=NI, then a *TRANSI-defined value is used. If aquifer contact is made on a cell with an intermediate I-index, it is assumed that that cell has a null cell lying on one side

of it in the I-direction. For this case, the aquifer uses the *TRANSI/*TRANLI multiplier that would have governed inter-block flow if the neighbour was not null. Similar comments hold for the J- and K-directions.

Examples:

```
*AQUIFER *BOUNDARY
*AQPROP ...

-or-

*AQUIFER *BOTTOM
*AQPROP ...
```

Multiple use of *REGION is allowed to define complex aquifer connections.

```
*AQUIFER *REGION   3  4:6
7:9 *IDIR
 *REGION 1:34   7:9 *JDIR
 *REGION 4:7  5:7   9   *KDIR
*AQPROP
 0.  0.3   150.0  0.0 .5
```

The acceptable range of values for the aquifer thickness is:

	SI **m**	**Field** **ft**
min	0.0	0.0
max	1.0E+4	3.28E+4

Aquifer Pressure Influence (Conditional) *AQFUNC

PURPOSE:

*AQFUNC indicates input of a cumulative pressure drop function.

TABLE:

```
*AQFUNC
  dimtime        dimpdrop
     :               :
```

DEFINITIONS:

dimtime
: Dimensionless time.

dimpdrop
: Nondimensionalized cumulative pressure drop in the aquifer.

DEFAULTS:

Conditional keyword. See explanation below.

CONDITIONS:

If this keyword appears, it must be in the RESERVOIR DESCRIPTION keyword group.

If *AQUIFER is encountered without *AQFUNC, then a default function is used (see below).

EXPLANATION:

The simulator calculates water influx from the aquifer using the Carter-Tracy approximation. For more information, refer to R. D. Carter and G. W. Tracy, "An Improved Method for Calculating Water Influx", Trans., AIME, Vol. 219, (1960), pp. 415-417. The method uses a dimensionless pressure influence function P(td), expressed as a function of dimensionless time td.

The default dimensionless pressure function used is the one given in A. F. Van Everdingen and W. Hurst,

"The Application of the Laplace Transform to Flow Problems in Reservoirs", AIME Dec. 1949, pp.305-324, for a constant terminal rate solution in an infinite radial aquifer. Influence functions for limited extent aquifers can also be found in this reference.

The default function is tabulated internally as follows:

DIMENSIONLESS TIME	CUMULATIVE DIMENSIONLESS PRESSURE DROP FUNCTION
0.01	0.112
0.05	0.229
0.10	0.315
0.15	0.376
0.20	0.424
0.30	0.503
0.50	0.616
0.70	0.702
1.00	0.802
1.50	0.927
2.00	1.020
3.00	1.169
5.00	1.362
7.00	1.500
10.00	1.651
15.00	1.829
20.00	1.960
30.00	2.147
40.00	2.282
50.00	2.388
60.00	2.476
70.00	2.550
80.00	2.615
90.00	2.672
100.00	2.723
200.00	3.064
300.00	3.263
300.00	3.263
400.00	3.406
500.00	3.516
600.00	3.608
700.00	3.684
800.00	3.750
900.00	3.809
1000.00	3.860

An extrapolation method for dimensionless times that go beyond the end of the table is required. If the above infinite extent table is used, an analytical expression is used for the extrapolation (see the Van Everdingen and Hurst). Otherwise, linear extrapolation in dimensionless time is used, which is appropriate for finite aquifers.

Further discussion of some compatibility issues relating to older versions of GEM are given above under the *AQUIFER section.

Pore Volume Cut-Off Threshold (Optional) *PVCUTOFF, *PVCUTFR

PURPOSE:

*PVCUTOFF controls the level at which a matrix block will be set null due to a small pore volume.

*PVCUTFR controls the level at which a fracture block will be set null due to a small pore volume.

FORMAT:

*PVCUTOFF *pvcut*

*PVCUTFT *pvfcut*

DEFINITIONS:

pvcut, pvfcut

Pore volume (raw block volume multiplied by porosity multiplied by net-to-gross ratio) below which a block will be considered to be null. *pvcut* applies to matrix blocks whereas *pvfcut* applies to fracture blocks.

The units of *pvcut* or *pvfcut* are (m^3 | ft^3).

DEFAULTS:

Optional keywords. If *PVCUTOFF is absent, the pore-volume cut-off criterion is not used to change a matrix cell's type. If *PVCUTFR is absent, the pore-volume cut-off criterion is not used to change a fracture cell's type.

CONDITIONS:

These keywords, if present, must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

This option ensures that matrix (via *PVCUTOFF), or fracture (via *PVCUTFR) blocks with small pore volumes can be systematically removed from the fluid flow simulation. Such small pore volume blocks can hinder convergence.

Null versus Pinched Out

A cell that is changed to type Null by *PVCUTOFF or *PVCUTFR is not pinched out, that is, there is no vertical inter-cell connection through it. Use keyword *PINCHOUTARRAY or *PINCHOUT-TOL to force a vertical connection through a cell.

Pinch Out Array (Optional) *PINCHOUTARRAY

PURPOSE:

*PINCHOUTARRAY defines pinch outs using an array input format. (See also *PINCHOUT-TOL.) (This keyword replaces the older keyword *PINCHOUT.)

FORMAT:

*PINCHOUTARRAY

DEFAULTS:

Optional keyword. Default: No pinch outs.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group.

If *PINCHOUTARRAY and *PINCHOUT-TOL are used together, a block satisfying either condition will be assigned pinched-out status. Note that, regardless of the technique chosen, blocks with true zero thickness (*DK value of 0.0, or corner point cells entered with top corner points that are equal to bottom corner points) will be treated as pinched out, unless they had been flagged null using the *NULL keyword (see later).

All array qualifiers and array reading options are allowed for specifying the required ni * nj * nk values. The array values should consist of 0's to indicate blocks that are pinched out and 1's to indicate blocks that are not pinched out. (This keyword is similar to the *NULL keyword in that 1's are used to indicate active blocks and 0's are used to indicate special blocks that do not fully participate in the simulation.)

Note that if a vertical stack of one or more corner point cells are pinched out, and they are surrounded above and below by active corner point cells, then the blocks above and below will not connect to each other UNLESS the pinched out cells form an <u>uninterrupted</u> stack of contacting blocks. The presence of a gap will break the connection, where *CORNER-TOL gives the tolerance describing how close cells need to be before they are deemed to make contact.

Pinching out blocks on *HYBRID refined grids is not recommended.

This keyword replaces the older *PINCHOUT keyword. Use of this older keyword is no longer recommended.

EXPLANATION:

*PINCHOUTARRAY indicates the modelling of pinched out layers. Such blocks will not participate in any of the simulator's flow calculations; that is, they will be inactive. However, fluid will be permitted to pass through them in the vertical direction (only).

Pinched out blocks are used to remove layers from the simulator's calculations in certain regions on a grid. Such layers may be required to model geological strata that exist in other portions of the grid but are not present in the pinched out region. The *PINCHOUTARRAY keyword corresponds to true geological pinch outs.

Blocks that are designated as pinched out allow fluid to pass through them vertically, but not laterally, and fluid can pass through a stack of one or more pinched out blocks on a grid. Pinched out blocks can also lie between active blocks on a grid and a refined grid region, allowing fluid to pass through between the grids. Two *HYBRID grids can even be connected vertically through intervening pinched out layers on their parent grid.

Pinched out blocks should have relatively small thicknesses (a *DK array value of near 0.0 or matching, or nearly matching, top and bottom corner points) since the transmissibility calculations between cells above and below the pinched out stack do not use the intervening stack's thickness.

A pinched out status set with *PINCHOUTARRAY over-rides an inactive setting using *NULL. This means that if a block has been designated in a *PINCHOUTARRAY list, fluid will pass through it regardless of it having been also designated inactive in a *NULL list. However, a *NULL setting overrides zero thickness; that is, a zero thickness block will not allow vertical fluid passage if it is flagged inactive using the *NULL keyword. Note that zero thickness overrides zero pore volume; that is, a block that is pinched out due to having zero thickness will allow fluid passage regardless of whether it was assigned non-zero porosity values or not.

The simulator uses the following hierarchy when determining whether a block is pinched out (allows vertical fluid passage) or is completely inactive. Note that Rule (1) overrides Rule (2), which overrides Rule (3), which overrides Rule (4).

1. The block has been flagged as pinched out using the *PINCHOUTARRAY keyword (a 0 value was assigned). This block will always be pinched out and this state will not be overridden.
2. The block has been flagged as inactive using the *NULL keyword (a 0 value was assigned). This block will not allow vertical fluid passage, unless overridden by Rule (1).
3. A zero thickness block will be pinched out (allows vertical fluid passage) if not overridden by Rules (1-2).
4. A zero pore volume block will be inactive and not allow any fluid passage unless overridden by Rules (1-3).

Example:

To pinch out the second layer of a 100 X 100 X 9 model use the following:

```
*PINCHOUTARRAY *IJK 1:100 1:100 2:2  0
```

Note that the remaining blocks in the model need not be referred to and their state remains unaltered.

Pinchout Tolerance (Optional) *PINCHOUT-TOL

PURPOSE:

*PINCHOUT-TOL controls the minimal thickness required to initiate an automatic pinched out connection (see also *PINCHOUTARRAY).

FORMAT:

*PINCHOUT-TOL pnctol

DEFINITIONS:

pnctol

Minimal thickness required under which a block is removed from the simulation and the block above it is connected directly to the block below. Dimensions are (m | ft).

DEFAULTS:

Optional keyword. The defaults are:

- 0.0010 (m | ft) for Corner Point grids <u>on the fundamental grid only</u>, if *PINCHOUTARRAY does not appear;
- 0.0002 (m | ft) for non-Corner Point grids <u>on the fundamental grid only</u>, if *PINCHOUTARRAY does not appear;
- (m | ft) otherwise.

Values at computer round-off levels are considered to be identically 0.0.

CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group.

If *PINCHOUTARRAY and *PINCHOUT-TOL are used together, a block satisfying either condition will be assigned pinched-out status. Note that, regardless of the technique chosen, blocks with true zero thickness (*DK value of 0.0, or corner point cells entered with top corner points that are equal to bottom corner points) will be treated as pinched out, unless they had been flagged null using the *NULL keyword.

Note that if a vertical stack of one or more corner point cells are pinched out, and they are surrounded above and below by active corner point cells, then the blocks above and below will not connect to each other UNLESS the pinched out cells form an <u>uninterrupted</u> stack of contacting blocks. The presence of a gap will break the connection, where *CORNER-TOL gives the tolerance describing how close cells need to be before they are deemed to make contact.

Having pinching out blocks on *HYBRID refined grids is not recommended.

EXPLANATION:

Blocks whose thickness are less than pnctol are considered to be pinched out. When this occurs, blocks above and below the pinched out blocks are connected as if the pinched out blocks did not exist. The minimum thickness “pnctol” at which this occurs is controlled using *PINCHOUT-TOL.

Pinched out blocks are used to remove layers from the simulator's calculations in certain regions on a grid. Such layers may be required to model geological strata that exist in other portions of the grid but are not present in the pinched out region. The *PINCHOUT-TOL and *PINCHOUTARRAY keywords correspond to true geological pinch outs.

Blocks that are designated as pinched out allow fluid to pass through them vertically, but not laterally, and fluid can pass through a stack of one or more pinched out blocks on a grid. Pinched out blocks can also lie between active blocks on a grid and a refined grid region, allowing fluid to pass through between the grids. Two *HYBRID grids can even be connected vertically through intervening pinched out layers on their parent grid.

*NULL settings override *PINCHOUT-TOL-generated pinch outs; that is, a small thickness block will not allow vertical fluid passage if it is flagged inactive using the *NULL keyword. Note that small thickness overrides zero pore volume; that is, a block that is pinched out due to having small thickness will allow fluid passage regardless of whether it was assigned non-zero porosity values or not.

Corner Point Tolerance (Optional) *CORNER-TOL, *PARTOL-TOL

PURPOSE:

*CORNER-TOL controls the minimal spacing required to separate corner points (see above for descriptions of Corner Point grids). It is also used for miscellaneous tolerance checking for corner point applications.

FORMAT:

*CORNER-TOL *cptol*
*PARTOL-TOL *partol*

DEFINITIONS:

cptol
Minimal spacing required to separate corner points and related quantities; that is, corner points that are closer than *cptol* are deemed to be the same. Dimensions are (m | ft).

partol
Minimal spacing required to separate corner points and related quantities. When a corner point grid has refinements with small cells, and a run terminates with a Fatal Error from subroutine GRCNC8, setting *partol* to a value below its default may remove those messages. Dimensions are (m | ft).

DEFAULTS:

If *CORNER-TOL is absent then *cptol* = 0.050 is assumed for all length units.

If *PARTOL-TOL is absent then *partol* = 0.005 is assumed for all length units.

CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

Corner points that lie within a distance of *cptol* are considered to be in the same place. If two corner points that belong to the same cell lie within a distance of *cptol*, then, either:

- one point belongs to the top of the cell and the other to the bottom, and that corner is pinched out;
- the cell is squeezed in the I or J direction and an error has occurred.

Points from neighbouring cells (four points from the top of one cell and four points from the bottom of the other) that are supposed to be touching to make a standard flow connection will be regarded as making contact if they lie within a distance of *cptol* (in top-bottom pairs).

If the average thickness of the cell as measured through its centre is less than a certain tolerance (see *PINCHOUT-TOL), then that cell will be designated as pinched out. However, for the Corner Point cells above and below to make a connection, those cells' top and bottom corner points must match to within the tolerance *cptol*.

Block Groups (Optional) *BLOCKGROUP

PURPOSE:

Define a block group.

ARRAY:

*BLOCKGROUP '*block_group*'

DEFINITIONS:

'*block_group*'
Quoted name of the block group. The maximum allowed number of characters is 80.

DEFAULTS:

If *BLOCKGROUP is absent, there are no block groups.

Each cell not referenced explicitly in the '*block_group*' definition is not a member of that block group.

CONDITIONS:

Array value of a cell should be either 0 (not a member of '*block_group*') or 1 (member of '*block_group*').

A block group may not be modified after it has been used via *BG.

EXPLANATION:

A block group is an arbitrary collection of grid blocks which can be used for addressing frequently a particular group of blocks. A block may belong to any number of block groups, and a block group may contain any non-zero number of blocks.

A block group is useful for assigning values to any grid array via sub-keyword *BG. See **Input of Grid Property Arrays** in chapter "Keyword Data Entry System". In addition, a block group can be applied to certain block-based data-input keywords (e.g., *REFINE).

When keyword *BLOCKGROUP appears multiple times with the same group name, the resulting group is the union of each definition. This lets you define complex regions on multiple grid levels. A block group's definition may not be modified after it has been used.

A second block group may be defined using another previously defined block group. This may be done either (1) using *BG array qualifier, or (2) no array qualifier but *MOD with *BG.

Similar to other property arrays, an LGR block that is not explicitly assigned to a block group automatically inherits the membership status of its parent block.

A block group is similar to a reporting sector (keywords *SECTOR, etc.). A reporting sector is used only for reporting simulation results whereas a block group is a more general block addressing tool. For example, a block group can define a reporting sector via keyword *SECTORARRAY '*Sector_Name*' *BG '*block_group*' *CON 1.

A block group may be defined in the Recurrent Data section. In addition to blocks defined in the Reservoir Description data section, a block group definition may include child blocks that have been created at that point by keyword *REFINE in recurrent data. A block group may not include child blocks created by keyword *DYNAGRID.

Example:

The following data defines a block group named 'Sand-1' consisting of a few cells in a 5x5x1 grid and part of the locally refined 3x3x2 grid in (3,3,1). The whole-grid porosity is overwritten with another value in the 'Sand-1' block group. A second block group 'Sand-2' is defined as equal to 'Sand-1' minus block (2,2,1). A third block group 'Sand-3' is the same as 'Sand-2' but is assigned in an alternative manner using *MOD.

```
*BLOCKGROUP 'Sand-1' *IJK  4 4 1  1
                           2 2 1  1
*BLOCKGROUP 'Sand-1' *RG 3 3 1 *IJK 1 1 1:2  1
                                    2 2 1:2  1
                                    3 3 1:3  0
*POR *CON 0.28
   *MOD *BG ‘Sand-1’= 0.34
. . .
*BLOCKGROUP 'Sand-2' *BG 'Sand-1' *CON 1
*BLOCKGROUP 'Sand-2' *IJK 2 2 1  0
. . .
*BLOCKGROUP 'Sand-3' *CON 0
   *MOD
      *BG 'Sand-1' = 1
      2 2 1 = 0
```

Structured Block Groups (Optional) *SBLOCKGROUP

PURPOSE:

Define a structured block group.

ARRAY:

*SBLOCKGROUP *'str_block_group'* n_i n_j n_k
{ uba_range }
*END_SBLOCKGROUP

DEFINITIONS:

'str_block_group'
Quoted name of the structured block group. The maximum allowed number of characters is 80.

n_i n_j n_k
Number of blocks in the I, J and K directions for the local structured block group, on the same line as '*str_block_group*'. Any references to individual I, J or K indices (e.g., *IVAR, *MOD *ARRAY) must correspond to this grid structure.

{ uba_range }
List of block references which together make up the structured block group. Each *uba_range* must appear on a separate line. The resulting list must define a set of blocks that index-wise has a rectangular shape consistent with *ni*, *nj* and *nk*. See "User Block Address" in chapter **Keyword Data Entry System.**

DEFAULTS:

Each cell not referenced explicitly in the '*str_block_group*' definition is not a member of that structured block group.

CONDITIONS:

Keyword *SBLOCKGROUP must appear only once for each '*str_block_group*'.

EXPLANATION:

A structured block group is a convenient tool for making multiple references in a consistent manner to a particular cell-wise rectangular area of the grid. The definition of a structured block group does not itself create grid blocks.

A structured block group is a collection of grid blocks which together can be collocated with a local I-J-K (structured) grid. This local grid's I, J and K indices increase in the same sense as the fundamental grid, and the local grid's origin is located at its corner closest to the fundamental grid's origin.

A block may belong to any number of structured block groups, and a structured block group may contain any non-zero number of blocks which form a (n_i,n_j,n_k) system.

A structured block group may be defined for any grid type.

A structured block group is useful for assigning values to any grid array via sub-keyword *SBG. See **Input of Grid Property Arrays** in chapter “Keyword Data Entry System”. In addition, a structured block group can be applied to certain block-based data-input keywords (e.g., *REFINE).

A structured block group is similar to the more generic block group defined by keyword *BLOCKGROUP. The only difference between them is that *SBLOCKGROUP defines a Cartesian (I-J-K) structure that allows local use of all the grid-array sub-keywords like *ALL and *IVAR.

A structured block group may be defined in the Recurrent Data section. In addition to blocks defined in the Reservoir Description data section, a structured block group definition may include child blocks that have been created at that point by keyword *REFINE in recurrent data. A structured block group may not include blocks created by keyword *DYNAGRID.

Example:

The following data defines two structured block groups. The first group named 'Sand-1' consists of a rectangular sub-region of the fundamental grid. The second group named 'Sand-2' consists of a collection of child cells resulting from local grid refinement in a number of adjacent parent cells. The whole-grid porosity is overwritten with another value in the 'Sand-1' block group.

```
*GRID *CART 50 50 10
. . .
** Structured Block group #1 is region (25:35,4:27,7:9)
*SBLOCKGROUP 'Sand-1'  11 24 3  ** 792 members
   25:35 4:27 7:9
*END_SBLOCKGROUP

** Apply 7x1x7 LGR to region (3:7,35,3:5)
*REFINE 3:7 35 3:5 *INTO 7 1 7

** Structured Block group #2 is LGR from multiple parents
*SBLOCKGROUP 'Sand-2'  30 1 21  ** 630 members
   3   35 3:5 / 3:7 1 1:7  ** -I partial
   4:6 35 3:5 / 1:7 1 1:7  ** Full
   7   35 3:5 / 1:4 1 1:7  ** +I partial
*END_SBLOCKGROUP

*POR *CON 0.28
   *MOD *SBG 'Sand-1'= 0.34
. . .
*SWINIT *SBG 'Sand-2'
   *KVAR . . .  ** 21 values
. . .
*SWCON *SBG 'Sand-2'
      *ALL . . .  ** 630 values
```

Sectors (Optional) *SECTOR

PURPOSE:

*SECTOR controls the output of reservoir data summarized on a regional basis.

FORMAT:

```
*SECTOR     'Sector_Name'     i1:i2     j1:j2     k1:k2
```

DEFINITIONS:

'Sector_Name'
: Sector name. (16 characters maximum).

i1:i2
: Indicates the I beginning and ending indices of the grid region where the sector is to be located.

j1:j2
: Indicates the J beginning and ending indices of the grid region where the sector is to be located.

k1:k2
: Indicates the K beginning and ending indices of the grid region where the sector is to be located.

DEFAULTS:

Optional keyword. Default: Full FIELD sector, with MATRIX and FRACTURE sectors if dual porosity or dual permeability is used.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

Sectors are collections of grid blocks defined using the grid block ranges following the sector name. The simulator prints various quantities by sector, which makes sectors useful for obtaining regional summaries. A grid block can belong to different sectors.

The sector 'FIELD' is reserved internally to represent all grid blocks. Also, 'MATRIX' is used for all matrix porosities, and 'FRACTURE' is used for all fracture porosities, for dual porosity reservoirs. As a sector is identified by its name, do not use 'FIELD', 'MATRIX', nor 'FRACTURE' as input sector names for *SECTOR.

The surface volumes of the various produced streams from each sector are printed in a summary table at the end of the simulation run. They are calculated by flashing the sector fluids through the standard separator declared in the initial conditions section. Separate flash calculations are performed for each phase in each sector and the results from these calculations are summed to obtain the total surface volumes.

Production splits are also given by sector. An estimate of how each stream arises from the oil and gas phases in the reservoir is provided by flashes done for each well, and is reported by sector.

Note that the oil and gas recoveries in GEM are calculated based on the actual volumes of oil and gas produced. This may be different from the decrease in the oil and gas in sector 'FIELD'. For example, consider an under-saturated oil reservoir with a value of original oil in place of OOIP, and original gas in place of OGIP. Injection of a stock tank volume of dry gas Q, with no production, results in zero oil recovery. However, the gas volume in the sector 'FIELD' will be greater than OGIP + Q, and the oil volume will be less than OOIP. This is because the dry gas is able to cause intermediate components to move to the gas phase. As a result, the sector stock tank gas volume will increase and it will be richer in intermediates than it was initially.

Example: Consider the following 7 x 6 x 1 grid with three sectors:

J = 6	S1	S1	S1	S1/S3	S3	S3	S3
J = 5	S1	S1	S1	S3	S3	S3	S2/S3
J = 4	S1	S1	S1			S2	S2
J = 3					S2	S2	S2
J = 2				S2	S2	S2	S2
J = 1			S2	S2	S2	S2	
	I = 1	2	3	4	5	6	7

The following input is needed to define these 2 sectors:

```
*SECTOR     'S1'      1:3  4:6   1
*SECTOR     'S1'        4    6   1
*SECTOR     'S2'      3:6    1   1
            'S2'      4:7    2   1
            'S2'      5:7    3   1
                      6:7    4   1
                        7    5   1
*SECTOR     'S3'      4:7  5:6   1
```

Note the different formats used to define the sectors; see *SECTORARRAY following for an alternate method.

Sector Array (Optional) *SECTORARRAY

PURPOSE:

*SECTORARRAY defines Sectors (see *SECTOR) using an array input format. Sectors permit output to be printed on a regional basis.

FORMAT:

*SECTORARRAY 'Sector_Name'

DEFINITIONS:

'Sector_Name'
Sector name. (16 characters maximum)

DEFAULTS:

Optional keyword. The full FIELD sector, and MATRIX and FRACTURE sectors if dual porosity or dual permeability is used, are always defined internally.

CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. All array qualifiers and array reading options are allowed for specifying the required ni * nj * nk values. The qualifiers and array values should come after 'Sector_Name'. The array values should consist of 0's (no sector membership for that cell) or 1's (indicating sector membership for that cell).

EXPLANATION:

Sectors are collections of grid blocks that the simulator uses to summarize various quantities. A grid block can belong to different sectors. See the description of *SECTOR above for further information about sectors.

The *SECTORARRAY keyword provides an array-based alternative to the *SECTOR range-based input format.

Example:

To set sector membership in a sector named 'Sector-1' for a few cells in the reservoir, use the following

```
*SECTORARRAY 'Sector-1' *IJK 1:5 1:5 1:1  0
                              4   4   1  1
                              2   3   1  1
```

where it is assumed the grid is dimensioned 5x5x1. Note that the string "1:5 1:5 1:1 0" is not actually required as "no membership" is the default state.

Sector Assignment via Names and an Array (Optional)

***ISECTOR, *SECTORNAMES**

PURPOSE:

*SECTORNAMES introduces sectors via a list of sector names and corresponding sector numbers (see *SECTOR for a discussion of sectors). This keyword should only appear once in the data set.

*ISECTOR assigns these sector numbers to cells using the standard array concepts.

FORMAT:

*SECTORNAMES 'Sector_Name_1' i1 'Sector_Name_2' i2 'Sector_Name_3' i3 …
*ISECTOR

DEFINITIONS:

'Sector_Name_1' i1

'Sector_Name_2' i2

…

Sector names. (16 characters maximum) and their associated numbers

DEFAULTS:

Optional keyword. The full FIELD sector, and MATRIX and FRACTURE sectors if dual porosity or dual permeability is used, are always defined internally.

CONDITIONS:

These keywords must be in the RESERVOIR DESCRIPTION keyword group.

*SECTORNAMES should appear before *ISECTOR. All array qualifiers and array reading options are allowed for specifying the required ni×nj×nk values for *ISECTOR. The values assigned by *ISECTOR should consist of the values that appear in the *SECTORNAMES list.

EXPLANATION:

Sectors are collections of grid blocks that the simulator uses to summarize various quantities. (See the description of *SECTOR for further information about sectors.) The *ISECTOR keyword provides an array-based alternative to assigning sector numbers that have been defined using *SECTORNAMES. Several instances of *ISECTOR can appear in the data set.

Example:

To set sector membership in two sectors named 'LAYER-1' and 'LAYER-2', use the following:

```
*SECTORNAMES ‘LAYER-1’ 1 ‘LAYER-2’ 2

*ISECTOR  *IJK 1:5 1:5 1:1  1
               1:5 1:5 2:2  2
```

where it is assumed that the grid is dimensioned 5x5x2. Provided that this not a dual porosity problem, then three sectors will be defined in the simulation: 'LAYER-1', 'LAYER-2' and ‘FIELD’, where the latter is the default sector consisting of all the active cells.

Flux Sectors (Optional) *FLUX_SECTORNAMES, *FLUX_ISECTOR

PURPOSE:

Specify flux sectors via sector number and grid array.

FORMAT:

*FLUX_SECTORNAMES fsector_defn_list
where
fsector_defn_list is any number of ('fsector_name' fsector_number) pairs

ARRAY:

*FLUX_ISECTOR

DEFINITIONS:

'fsector_name'
: Quoted name of a flux sector (maximum of 16 characters). All the flux-sector names must be unique.

fsector_number
: Positive integer uniquely associated with 'fsector_name', starting from 1. fsector_number is assigned to cells via grid-array keyword *FLUX_ISECTOR.

*FLUX_ISECTOR
: Assign flux-sector numbers to grid cells. This keyword may appear at most once, for the fundamental grid. This enforces the condition that each cell belongs to at most one flux sector. The result is that there is no overlap between flux sectors, so that flux between sectors is well defined.

DEFAULTS:

If no flux sectors are defined, no flux-sector reports are issued.

If flux sectors are defined, any cell not assigned to a flux sector is not included in any flux-sector report.

CONDITIONS:

If *FLUX_SECTORNAMES appears, it must appear at most once and before *FLUX_ISECTOR which itself must appear exactly once.

Any value assigned by *FLUX_ISECTOR must appear in the *FLUX_SECTORNAMES list.

Array qualifiers *MATRIX and *FRACTURE may not be used with *FLUX_ISECTOR.

The flux-sector option may not be used together with dynamic gridding.

EXPLANATION:

Flux sectors are collections of grid blocks, similar to in-place sectors defined via *SECTOR. Unlike in-place sectors, flow can be determined **between** flux sectors. If two flux sectors share a boundary, then flow from one flux sector to another will be calculated. In addition, total flow (Influx) from all surrounding blocks into a flux sector will be calculated.

The *FLUX_ISECTOR keyword provides a grid-array-based assignment for positive flux sector numbers that have been defined using *FLUX_SECTORNAMES. Since *FLUX_ISECTOR may appear at most once, each grid cell is assigned to at most one flux-sector number. Thus there is no overlap between individual flux sectors. This is different from in-place sectors, of which more than one may be assigned to a grid cell.

Cells not assigned a flux-sector number are not included in the flux-sector reports and will appear in flux-sector maps with flux-sector number 0.

A flux sector does not report in-place statistics. In order to get in-place statistics for a flux sector, defined an in-place sector via *SECTOR (or equivalent) with the same cell map as the flux sector.

The particular information included in each flux-sector report is determined by *OUTPRN *FLUX_SECTOR and *OUTSRF *FLUX_SECTOR.

The frequency of flux-sector reporting is determined by *WPRN *SECTOR and *WSRF *SECTOR (the same as in-place sector reports).

The flux-sector option may be used with a natural-fractured grid type, in which case each fracture cell has the same flux-sector number as its associated matrix cell(s).

The flux-sector option may be used with local grid refinement, in which case each child cell has the same flux-sector number as its parent cell.

Example:

Use the following to set flux sector membership in four sectors named 'Area-I', 'Area-I', 'Area-III' and 'Area-IV':

```
*GRID *CART 13 14 6
. . .
*FLUX_SECTORNAMES
   'Area-I'   1
   'Area-II'  2
   'Area-III' 3
   'Area-IV'  4
*FLUX_ISECTOR
   *IJK 1:6  1:7  1:6  1
        7:13 1:7  1:6  2
        1:6  8:14 1:6  3
        7:13 8:14 1:6  4
```

Irregular Connection (Optional) *IRCONNECT

PURPOSE:

Specify irregular interblock connection data.

FORMAT:

*IRCONNECT { uba_1 uba_2 L_1 L_2 A_{12} (*I | *J | *K) }

DEFINITIONS:

*IRCONNECT { … }
*IRCONNECT may be followed by one or more sets of data, with one set per line. One set of data specifies one connection.

uba_1 uba_2
User Block Addresses of two active cells. The order of appearance is not significant. Each cell must not be null, pinched out or the parent cell of a locally refined grid. Index ranges are not allowed.

L_1 L_2
Distance (m | ft) between each cell's center (pressure reference location) and their common face. L_1 is the distance in cell uba_1, and L_2 is the distance in cell uba_2.

A_{12}
Area (m^2 | ft^2) of the face common to cells uba_1 and uba_2. A_{12} includes any area modification factor (see *VAMOD), so any area factor specified by *VAMOD data will not be applied to this connection.

(*I | *J | *K)
Nominal direction of connection. This direction will be used when building an interblock quantity from direction-dependent data, e.g., fluid transmissibility from permeabilities.

DEFAULTS:

If *IRCONNECT is absent, no irregular connection data is assigned.

CONDITIONS:

Keyword *IRCONNECT must appear at most once in a data set. Specification of data for multiple connections must be done in table form, with data for each connection on its own line.

EXPLANATION:

A **regular connection** is an interblock connection that is generated automatically between adjacent active blocks. For *GRID *CART and *GRID *RADIAL, those blocks have adjacent I-J-K indexing (same index in two directions and index value in the third direction different by 1). For *GRID *CORNER and *VARI, those blocks share some contact area.

An **irregular connection** is an interblock connection that is not generated automatically. Often an irregular connection is the result of a fault, where a block column shifts relative to neighbouring columns and the strict stair-step relationship between blocks is lost.

Irregular connection data is interblock connection data for a regular connection that is not generated automatically. When uba_1 and uba_2 correspond to a regular connection, *IRCONNECT overwrites the regular interblock connection data.

An irregular connection has no restrictions with respect to which physical processes can be modeled for that connection. This is unlike *SCONNECT whose simpler form results in some restrictions.

Building Interblock Transmissibility

Interblock transmissibility is based on resistance in series. Consider blocks 1 and 2 with common contact area A_{12} and half-block sizes L_1 and L_2, respectively and absolute permeabilities K_{m1} and K_{m2}, respectively, in direction m (I, J or K). The resistance to flow in block 1 (center to common face) is $L_1/(A_{12} \cdot K_{m1})$ and the resistance to flow in block 2 (common face to center) is $L_2/(A_{12} \cdot K_{m2})$. The interblock transmissibility is the inverse of the total resistance between block centers:

$$1 / T_{12} = [\, L_1 / (A_{12} \cdot K_{m1})] + [\, L_2 / (A_{12} \cdot K_{m2}) \,]$$

or $$T_{12} = A_{12} / [\, (L_1 / K_{m1}) + (L_2 / K_{m2}) \,]$$

Quantities A_{12}, L_1, L_2 and direction m are obtained from the grid specification (regular connection) or from *IRCONNECT (irregular connection). Permeabilities K_{m1} and K_{m2} can be static, obtained from keywords *PERMI/J/K and not changed during the run. However, some options (e.g., dilation) involve variation of permeabilities which causes frequent recalculation of interblock transmissibilities. In this case T_{12} is recalculated internally from updated K_{m1} and K_{m2} for both regular and irregular connections. This capability is missing for the alternate special-connection keyword *SCONNECT.

Other Interblock Calculations

Transmissibility is specific to convective fluid flow in a porous medium. Other physical processes have corresponding interblock flows based on resistance in series, whose calculations are similar to T_{12} with a property analogous to permeability.

Physical Process	Analogous Property
Fluid convection	Permeability
Molecular diffusion	Diffusivity
Thermal conduction	Thermal conductivity

Some options require a representative block value of fluid velocity which is calculated from interblock fluid velocity of the block connections. Theoretically this interblock velocity transmissibility is T_{12}/A_{12} but the general formula is

$$T_{V12} = 1 / [\, (L_1 / K_{m1}) + (L_2 / K_{m2}) \,]$$

Irregular Geometries

For irregular grid or cell geometries L_1, L_2 and A_{12} can be regarded as general parameters that may not necessarily correspond to actual block half-sizes and common contact area. For example, cross-sectional area may vary significantly between the block centers, as it does in the radial direction of a radial grid. In the radial case, A_{12} is chosen as the area at the contact location and quantities L_1 and L_2 are integrals of $A_{12} \cdot dx/A(x)$ over the respective distances.

Irregular geometry includes grids which may be based on blocks that do not necessarily have six sides and eight corners. The only restriction is that interblock flow must depend upon the flow property (absolute permeability for convection) of only the two blocks in question.

Dynagrid Models

Unlike *SCONNECT, use of *IRCONNECT is supported in models with *DYNAGRID.

Disallowed Connection Types

Each of the following connection types has a transmissibility calculation that cannot be written in the form of the T_{12} formula, above, and so cannot be modified with *IRCONNECT:

- matrix-fraction connection
- nine-point connections

Example:

The following data specifies an irregular connection in a fault situation.

```
*GRID *VARI 10 1 10
. . .
*IRCONNECT
   1,1,1  2,1,2  5.2  7.3  83.44 *I  ** Irregular
   1,1,2  2,1,3  4.6  8.3  57.2  *I  ** Irregular
   1,1,3  2,1,3  5.1  7.1  60.2  *I  ** Modified
```

Special Connections (Optional) *SCONNECT

PURPOSE:

*SCONNECT allows the definition of flow connections between blocks that would not otherwise be generated by the simulator. Also, *SCONNECT allows existing flow connections to be overridden.

FORMAT:

```
*SCONNECT    1st_cell_desc_1  2nd_cell_desc_1 trms_1
             1st_cell_desc_2  2nd_cell_desc_2 trms_2
             1st_cell_desc_3  2nd_cell_desc_3 trms_3
                                   …
```

DEFINITIONS:

1st_cell_d..., 2nd_cell_d...

Descriptions of active cells. If the cell is on the 2nd_cell_d... fundamental grid, then its cell descriptor consists of 3 integers, i j k, separated by blanks. If the cell is on a refined grid, then its cell descriptor consists of 3 integers, giving the location of the fundamental grid parent cell, a slash (/), and 3 integers giving the location of the cell on the refined grid. An *M or *F can follow indicating Matrix or Fracture if the simulation is using *DUALPOR or *DUALPERM. All items should be separate by blanks. Two such descriptors are required, describing the two cells between which flow is to occur. The cells cannot be null or pinched out, and they cannot be parent cells.

trms

The numerical value for the flow transmissibility. The value is expected to be the result of taking a cross-sectional area for the flow, multiplying it by a permeability, and dividing it by a centre-to-centre inter-block distance. No fluid mobility contribution is expected, as it will be supplied internally by the simulator.

Dimensions are (md-m | md-ft).

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group. *SCONNECT is not allowed in models using *DYNAGRID. Use *IRCONNECT instead of *SCONNECT for establishing connection between two blocks in presence of *DYNAGRID.

EXPLANATION:

*SCONNECT allows the definition of special flow connections and the alteration of flow connections generated by the simulator.

If the two descriptors describing a connection in a *SCONNECT line are a pair that the simulator has NOT already generated connection between, then the connection given by the *SCONNEC input is added into the simulator's connection list. These connections are treated like all the others.

If the two descriptors describing a connection in a *SCONNECT line are a pair that the simulator has already generated a connection between, then the connection given by the *SCONNECT input overwrites the simulator-generated connection. This has the effect of replacing the existing transmissibility value by the "trms" value provided by the *SCONNECT keyword. This can be used to set transmissibilities to specific values in a way that avoids printing out lists of inter-block connections and using multipliers to adjust values.

Example:

Connect the first and last blocks in a 1 dimensional problem:

```
...
*GRID *CARTESIAN 10 1 1
...
*SCONNECT  1 1 1  10 1 1  200.
...
```

Interblock Matrix-Fracture Connection (Optional) *NOINTERMF

PURPOSE:

*NOINTERMF indicates that the inter-block matrix-fracture connection will not be established when a fracture block is null but its collocated matrix block is non-null in a dual-continua model.

FORMAT:

*NOINTERMF

DEFAULTS:

Optional keyword.

Default: Inter-block matrix-fracture connection is established when a fracture block is null but its collocated matrix block is non-null in a dual-continua model.

CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group.

EXPLANATION:

In a dual-continua model inter-block matrix-fracture connection is established when a fracture block is null but its collocated matrix block is non-null. Keyword *NOINTERMF, if present, reverses this behavior.

An example below shows how the connection pattern gets modified in presence of this keyword.

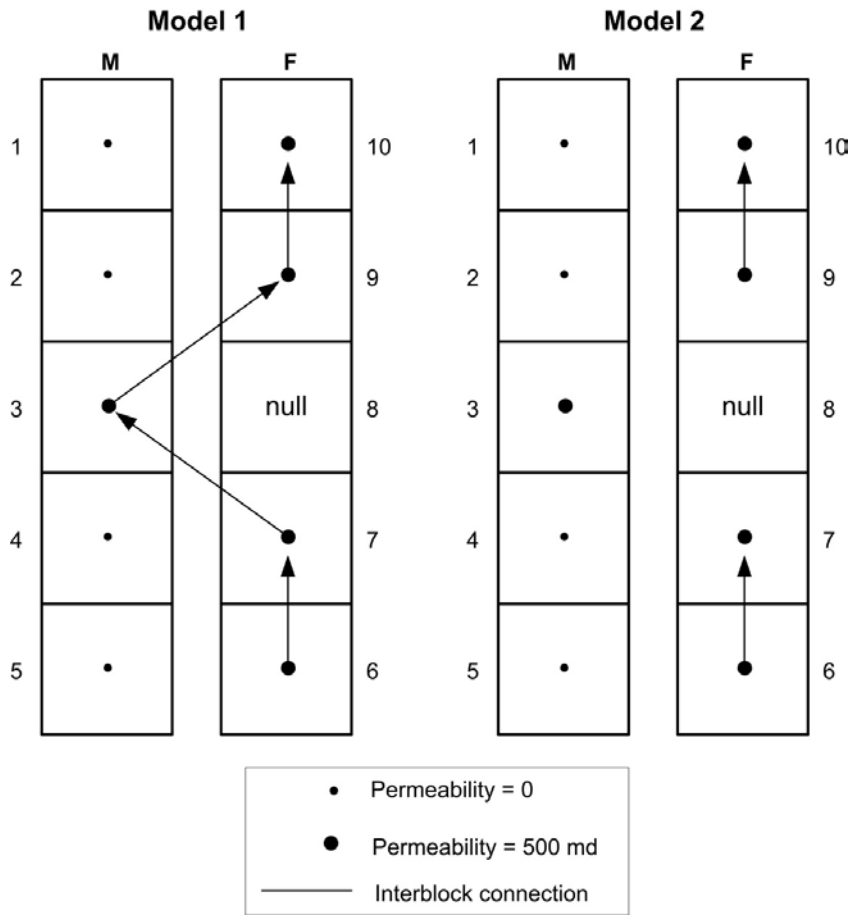
Example:

Consider a simple 1-D, 5 grid-blocks, dual-permeability, isothermal model, as shown in the following figure. The matrix and collocated fracture grid blocks are shown side-by side along with block numbers for illustration purposes. The blocks with smaller dots have zero-permeability in all the three directions while the blocks with larger dots have a permeability of 500 md in all the three directions.

Figure on the left hand side (Model-1) shows the default connectivity; where due to presence of null fracture block (8), inter-block M-F connections are established between block numbers (7-3), and (3-9).

Figure on the right hand side shows Model-2 that uses keyword *NOINTERMF. In this case the inter-block M-F connections between block numbers (7-3) and (3-9) are not established.

Note that due to the above permeability distribution no intra-block (M-F) connections are made, since either matrix permeability is zero for the collocated fracture block, or the fracture block is null for the collocated matrix block with non-zero permeability.



Related keywords: *SCONNECT, *IRCONNECT.

Fault Array (Optional) *FAULTARRAY

PURPOSE:

*FAULTARRAY signals the input of an array of binary flags which controls whether individual block faces are connected using standard connections or fault connections.

ARRAY:

*FAULTARRAY

DEFAULTS:

Standard connections assumed.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group. *FAULTARRAY is not necessary with corner-point options as the block corners determine connections directly. For Cartesian grids, Builder will automatically generate this array if faults exist in the topmost structure map. Care must be taken if the user overrides the automatically generated values.

All array reading options are valid. The most commonly used array reading subkeyword used with this option would be *CON.

EXPLANATION:

The *FAULTARRAY values consist of a single integer which defines how all of a grid blocks connections are made. A standard connection does not account for depth as it connects two blocks. It only takes layer number into account. In other words, blocks are connected even if the difference in the two block depths make a physical connection impossible. A fault connection accounts for depth when creating a connection and would create connections between blocks which physically touch. This is the default for all corner-point options.

The value of *FAULTARRAY controls how each of the four areal connections are made. The four connections are labeled nilow, nihigh, njlow, njhigh where i refers to the i direction and j refers to the j direction. Low refers to flow between block i (or j) and i-1 (or j-1). High refers to flow between block i (or j) and block i+1 (j+1).

The *FAULTARRAY binary integer flag uses the following convention:

nilow, nihigh, njlow, njhigh = 0 if the connection is a standard connection

nilow, nihigh, njlow, njhigh = 1 if the connection is a fault connection

The value of *FAULTARRAY for a block is:

IVAL = nilow + 2*nihigh + 4*njlow + 8*njhigh

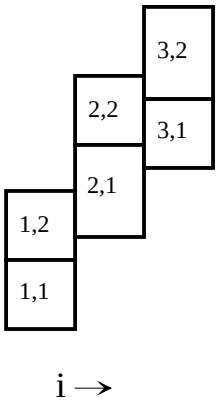
Thus if all connections are standard IVAL = 0, and if all connections take into account block depths (are fault connections), IVAL = 15.

Use:

```
**all connections areally are fault connections
 *FAULTARRAY  *CON  15
**all i connections are fault connections,
**all J connections are standard
 *FAULTARRAY  *CON   3
```

Example:

Standard Connections: i Connections	Fault Connections: j Connections
1,1 is connected to 2,1	2,1 is connected to 3,1
2,1 is connected to 3,1	1,2 is connected to 2,1
1,2 is connected to 2,2	2,2 is connected to 3,2
2,2 is connected to 3,2	2,2 is connected to 3,1



Compaction/Dilation Rock Type (Optional) *CROCKTYPE

PURPOSE:

This keyword is used when more than a single rock type is assigned in different regions of a reservoir or when a table of porosity and permeability multipliers vs. pressure is required.

FORMAT:

*CROCKTYPE rock_number

DEFINITIONS:

*CROCKTYPE:
Indicates the use of compressibility/compaction/dilation rock type regions in the reservoir.

rock_number
Rock_number for this compressibility/compaction/dilation rock type.

DEFAULTS:

Optional keyword. No default value.

CONDITIONS:

This keyword must be located in the Reservoir Description section. Its subkeywords have to be input in formats as explained on consecutive pages. If this keyword does not appear, the simulator will use a constant rock compressibility and its corresponding reference pressure which are entered by using the *CPOR and *PRPOR keywords, or by use of the *CROCK keyword (see the Rock-Fluid section).

Note that GEM provides an alternate, independent way of entering this type of data using the *CPOR and *PRPOR keywords (see earlier). Also, the *CROCK keyword is available (see the Rock-Fluid Data section).

If *CROCKTYPE does appear, settings for *CPOR and *PRPOR will be mostly ignored. (The default aquifer compressibility does use the *CPOR value however; see below.) Thus, blocks that are not assigned compressibilities using the *CCPOR/*CPRPOR subkeywords of *CROCKTYPE will end up with no compressibility values at all.

EXPLANATION:

This option allows the user to define different rock compressibility options in various parts of the reservoir. In addition to the standard relation which calculates por(p) as a function of a constant rock compressibility, a reference pressure and reference porosity:

$$por(p) = por_input*[1+ccpor*(p-cprpor)]$$

where:

por_input	:	initial porosity
ccpor	:	constant rock compressibility
cprpor	:	constant reference pressure
p	:	pressure
por(p)	:	porosity at pressure p

the user can input a table of porosity and permeability multipliers vs. pressures, which exhibits reversibly, irreversibly or hysteresis to model compaction processes. When table options are used, please note that mul_por(p) (below) must be set up to account for both normal compressibility and compaction/dilation effects.

If the constant rock compressibility and reference pressure option is used, the equation above is used to calculate the porosity at pressure p. If the table of multipliers vs. pressure option is used, the relation below is applied to calculate porosity at pressure p:

por(p) = por_input * mul_por(p)

where:

por_input	:	initial porosity
mul_por(p)	:	porosity multiplier at a pressure p
por(p)	:	porosity at pressure p

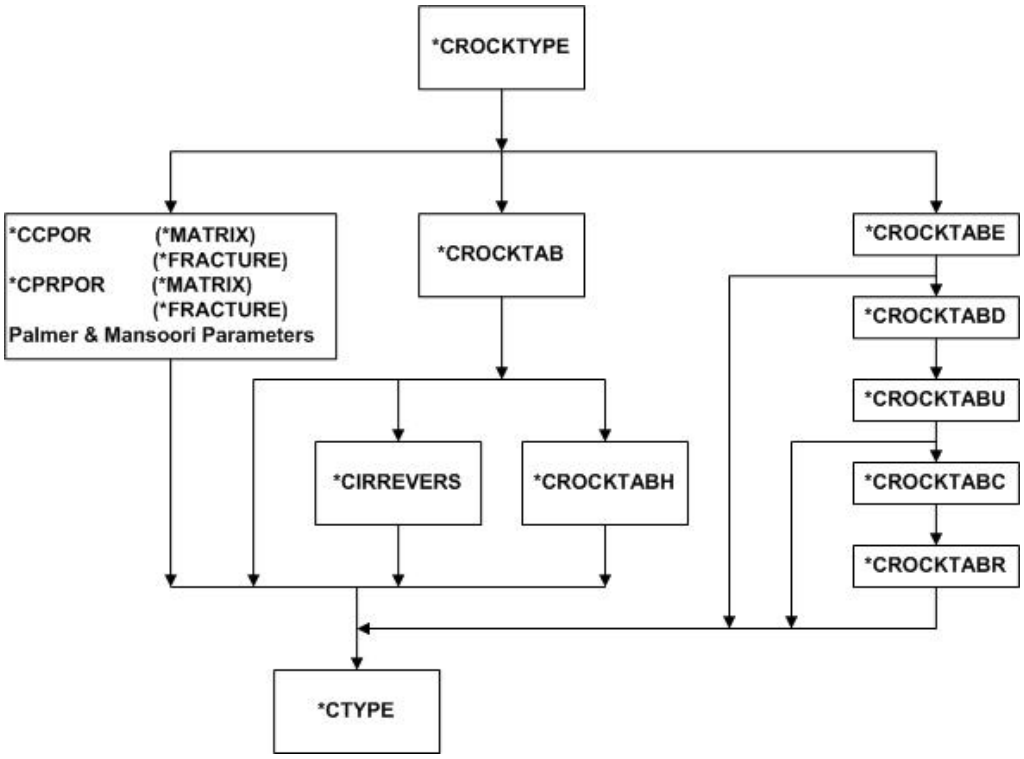
It should be noted that the analytical aquifer only uses values of rock compressibility and its corresponding reference pressure entered with keywords *CPOR and *PRPOR. When the aquifer option is applied and these values are not given (i.e. omitted from the data file), the aquifer model will default rock compressibility in all aquifers to 0.0. Hence *CPOR and *PRPOR must always be included in data sets which use an analytical aquifer, even if the reservoir compressibility is being obtained from data entered under *CROCKTYPE.

Parameters for a specific coal-based geomechanics model due to Palmer and Mansoori can also be entered following *CROCKTYPE as follows:

*POISSR	ν
*YOUNGM	E
*STRINF	ε_L
*PRESLN	p_L
*EXPPM	a

Further details concerning these parameters can be found in a later section.

The figure below shows the relationship among keywords being used in the rock compaction/dilation model. *CROCKTYPE is the main keyword to signal when the model is used. Without this keyword, the other keywords shown are invalid. Descriptions and usage of all the compaction/dilation keywords can be found on pages following. Note that keywords used for the rock compaction model cannot be mixed with those used for the rock dilation model and vice versa. Compaction and dilation may be used together in different *CROCKTYPE regions of a reservoir model however, but the two regions must not be in communication with each other (for instance, they must be separated by null blocks).



Keyword Relation in Rock Compaction/Dilation Model

Compaction/Dilation Rock Compressibility (Optional)

***CCPOR, *CPRPOR**

PURPOSE:

*CCPOR signals the input of a rock compressibility value that will be used in a *CROCKTYPE region of the reservoir defined on the *CTYPE card.

*CPRPOR signals the input of a reference pressure for the corresponding rock compressibility under the keyword *CCPOR.

These keywords are equivalent to *CPOR and *PRPOR but can be defined in different regions of the reservoir.

FORMAT:

*CCPOR	(*MATRIX)	ccpor
	(*FRACTURE)	
*CPRPOR	(*MATRIX)	cprpor
	(*FRACTURE)	

DEFINITIONS:

ccpor
: Pressure dependence of formation porosity; that is, rock compressibility. (1/kPa | 1/psi)

cprpor
: Reference pressure for calculating the effect of rock compressibility ccpor. (kPa | psi)

DEFAULTS:

Optional keywords. No default values.

CONDITIONS:

This keyword must be located in the Reservoir Description section.

The keywords *CCPOR and *CPRPOR must be located under the keyword *CROCKTYPE.

The *MATRIX and *FRACTURE subkeywords are used to indicate the matrix and fracture dependence.

When using dual porosity models, both matrix and fracture rock compressibilities and reference pressures should be input. If no fracture rock compressibility and its corresponding reference pressure are input, their values will be assumed to be the same as those of the matrix blocks, however; a warning message will be given. If no matrix rock compressibility and its reference pressure are input, an error message is given and the simulator stops.

EXPLANATION:

These keywords are equivalent to the keywords *CPOR and *PRPOR but can be applied to individual rock type regions. See definition of *CPOR and *PRPOR in the Reservoir Description section.

Compaction Rock Table (Optional) *CROCKTAB

PURPOSE:

*CROCKTAB indicates a data table for porosity and permeability multipliers versus pressure lookup. This keyword replaces the keywords *CCPOR and its reference pressure when the relation between rock compressibility and pressure is not linear or when permeability is changing with pressure. The porosity and permeability multipliers will be determined by linear interpolation of the *CROCKTAB table. Please note that *CPOR is still required by the analytical aquifer model to calculate aquifer compressibilities.

FORMAT:

```
*CROCKTAB
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

press

Minimum pressure is 101 kPa (14.7 psia). (kPa | psi).

por_mult

porosity multiplier (dimensionless) = V/Vi

where:

V is the pore volume measured at the pressure “press”.

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

horizontal permeability multiplier (dimensionless) = permh/permhi

where:

permh is the horizontal permeability measured at the pressure “press”.

permhi is the input (initial) horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

vertical permeability multiplier (dimensionless) = permv/permvi

where:

permv is the vertical permeability measured at the pressure “press”.

permvi is the input (initial) vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default values for ‘press’ or ‘por_mult”.

Default value of "ver_perm_mult" is "hor_perm_mult".

Default value of "hor_perm_mult" is 1.00.

CONDITIONS:

When the keyword *CROCKTAB appears, some restrictions apply:

1. This keyword must be located in Reservoir Description section.
2. This keyword must be located under the keyword *CROCKTYPE.
3. This keyword can not be used with subdomain, MINC and hybrid grids.
4. When using the keyword *CROCKTAB, keywords *CCPOR and *CPRPOR should not appear in the same *CROCKTYPE region. *CCPOR and *CPRPOR are ignored for the *CROCKTYPE which uses *CROCKTAB. Compressibilities in the reservoir are determined using table lookup. *CPOR may still be required if an analytical aquifer model is used.
5. The first two columns in the table, pressure and porosity multipliers always exist. The permeability multipliers (third and fourth columns) are optional.
6. If a current pressure falls outside the range of the table, linear extrapolation is applied to compute the multipliers at that pressure.
7. Pressures in the table should increase monotonically down the column.
8. Porosity multipliers should remain same or increase with the increase in pressure.
9. The *INT table input option does not apply to this table.

EXPLANATION:

Under the keyword *CROCKTAB, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods. Normally the pressure range of this table should encompass the minimum and maximum pressures expected during a simulation. The minimum producer bottomhole pressure and maximum injector injection pressure would normally define this range. If the pressure falls outside the table range, linear extrapolation is applied to compute values of multipliers outside the range. When the porosity multiplier is computed, the porosity at that pressure is given by:

por(p) = por_input * mul_por(p)

where:

por_input	:	initial input porosity (*PORI)
mul_por(p)	:	porosity multiplier at a pressure p
por(p)	:	porosity at pressure p

Please note that the compressibility *CPOR is not used to determine por(p), but is required if the analytical aquifer model is employed.

The use of *CROCKTAB keyword allows the user to input non-linear relationships between porosity/permeability multipliers and pressures. Without the use of keywords *CROCKTABH and *IRREVERS (as discussed later), the rock behavior is reversible (a default process in the simulator) i.e. the relationship between porosity/permeability multipliers vs. pressures always remains on the main path defined by this single table for this region.

Compaction Hysteresis Rock Table (Optional) *CROCKTABH

PURPOSE:

*CROCKTABH signals entry of hysteresis subtables for a rock type.

FORMAT:

```
*CROCKTABH
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

pressure

Minimum pressure is 101 kPa (14.7 psia) (kPa | psi).

por_mult

porosity multiplier (dimensionless) = V/Vi

where:

V is the pore volume measured at pressure "press".

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

horizontal permeability multiplier (dimensionless) = permh/permhi

where:

permh is the horizontal permeability measured at the pressure "press'.

Permhi is the input initial horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

vertical permeability multiplier (dimensionless) = permv/permvi

where:

permv is the vertical permeability measured at the pressure "press".

permvi is the input initial vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default values for 'press' or 'por_mult".

Default value of "ver_perm_mult" is "hor_perm_mult".

Default value of "hor_perm_mult" is 1.0.

CONDITIONS:

When the keyword *CROCKTABH appears, the following restrictions apply:

1. This keyword must be located in the Reservoir Description section.
2. This keyword is optional and only used under the keyword *CROCKTAB
3. This keyword can not be combined with the keyword *CIRREVERS.
4. There must be at least two subtables (two hysteresis curves).
5. The first two columns in the table, pressure and porosity multipliers must always exist. The permeability multipliers (third and fourth columns) are optional.
6. The pressure and multipliers in the first row of each subtable of the corresponding rock compaction hysteresis curve must lie on the main path curve. Further, to avoid potential numerical problems, the first row of each subtable must be identical to a corresponding row on the main path from which the hysteresis path starts.
7. Pressures in each subtable must increase monotonically down the column.
8. Porosity multipliers should remain same or increase with the increase in pressure.
9. First pressures in subtables must increase monotonically from the first subtable to the last subtable.
10. To do interpolation based on the data subtables of hysteresis curves the pressure on the main path (called a rebound pressure) at which a hysteresis phenomenon starts occurring must lie between a pressure range defined by the first pressures in the first and last subtables. If the rebound pressure is less than the first pressure in the first subtable or is greater than the first pressure in the last subtable, the rock behaves as if it was on the main path and there is no hysteresis occurring.
11. The *INT table input option does not apply to this table.

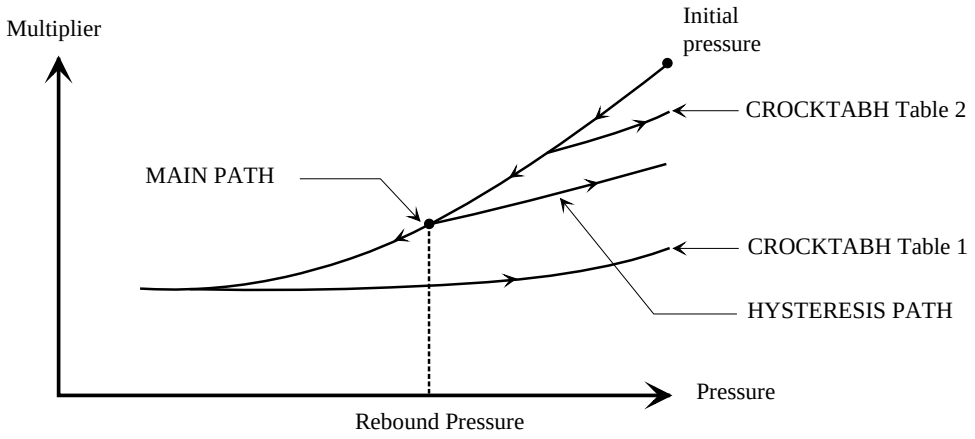
EXPLANATION:

This keyword is designed to model porosity and permeability multipliers versus pressure behavior that exhibits hysteresis effects. In the explanation below, porosity behavior will be described but the behavior also applies to permeability multipliers.

While pressure decreases, porosity multiplier vs. pressure behavior is defined by the main path (non-hysteresis) input on the *CROCKTAB table. When the pressure increases while on the main path, a rebound pressure is set. The porosity multiplier vs. pressure behavior is now free to follow a path defined by *CROCKTABH relationships.

If the rebound pressure lies within the range defined by the first pressures on the first and last *CROCKTABH tables, the porosity multiplier vs. pressure relationship is defined by interpolating bounding *CROCKTABH curves to obtain a hysteresis porosity multiplier vs. pressure relationship for the specific rebound pressure. The porosity multiplier is calculated using the rebound pressure and bounding *CROCKTABH tables until the pressure is reduced below the rebound pressure. When this occurs, the *CROCKTAB table is used to define the porosity multipliers vs. pressures relationship.

If the rebound pressure is less than the first pressure in the first *CROCKTABH table or greater than the last pressure in the first *CROCKTABH table, the *CROCKTAB table is used to define the porosity multiplier vs. pressure behavior. Hysteresis is only allowed to occur if the rebound pressure lies between the first pressures in the first and last *CROCKTABH tables.



Hysteresis Process for a Rock Type with Two *CROCKTABH Tables

Compaction Irreversibility Flag (Optional) *CIRREVERS

PURPOSE:

This keyword indicates the behavior of a rock type to be irreversible.

FORMAT:

*CIRREVERS

DEFAULTS:

If this keyword does not appear, the rock behavior is reversible.

CONDITIONS:

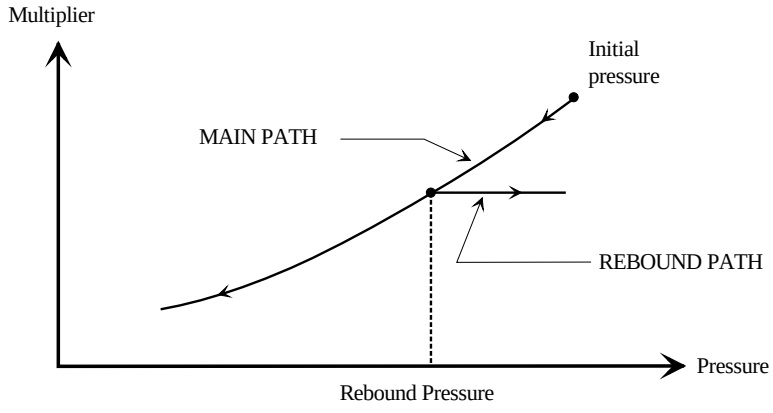
This keyword must be located in Reservoir Description section.

This keyword is optional and located under the keyword *CROCKTAB (CROCKTAB data entries follow *CIRREVERS)

This keyword can not be combined with the keyword *CROCKTABH. It should not be input when the keyword *CROCKTABH appears.

EXPLANATION:

This keyword is designed to allow the rock behavior to be irreversible. As long as pressure decreases the main path (defined by *CROCKTAB) is followed. When the pressure increases while on the main path defined by the *CROCKTAB table a rebound pressure is set at this point. The multiplier versus pressure behavior no longer is defined by the main path but by a rebound path where multipliers are kept constant with pressure. On the horizontal path (or rebound path), if the pressure is higher than the rebound pressure, the multipliers are kept at the values defined by the rebound pressure. When the pressure drops below the rebound pressure, the rock behavior will again resume the multiplier versus pressure behavior defined by the main path until the next increase in pressure, at which a new rebound pressure and path is set.



Irreversible Process

Dilation Rock Table in Elastic Zone (Optional) *CROCKTABE

PURPOSE:

This keyword is used for the rock dilation model only. The keyword indicates that rock behavior is acting within an elastic zone when pressure in the rock is below its yield point pressure. If an analytical aquifer is applied, *CPOR and *PRPOR are still required.

ARRAY:

```
*CROCKTABE
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

press

Minimum pressure is 101 kPa (14.7 psia). (kPa | psi).

Maximum pressure is a yield point pressure of a rock type.

por_mult

Porosity multiplier (dimensionless) = V/Vi

where:

V is the pore volume measured at the pressure " press"

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

Horizontal permeability multiplier (dimensionless) = permh/permhi

where:

permh is the horizontal permeability measured at the pressure " press".

permhi is the input (initial) horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

Vertical permeability multiplier (dimensionless) = permv/permvi

where:

permv is the vertical permeability measured at the pressure " press".

permvi is the input (initial) vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default value for 'press' or 'por_mult' .

Default value of 'ver_perm_mult' is 'hor_perm_mult'.

Default value of ‘hor_perm_mult’ is one.

If a value of hor_per_mult is less than zero, it is replaced by a value of one.

If a value of ver_per_mult is less than zero, it is replaced by a value of hor_perm_mult on the same row.

CONDITIONS:

When the keyword *CROCKTABE appears, some conditions apply:

1. This keyword must be located in Reservoir Description section.
2. This keyword must be located under the keyword *CROCKTYPE.
3. This keyword can not be used with subdomain, MINC and hybrid grids.
4. When keyword *CROCKTABE is used, keywords *CCPOR and *CPRPOR should not appear under the same *CROCKTYPE. Keywords *CPOR and PRPOR may still be required if an analytical aquifer model is used.
5. The first two columns in the table such as pressure and porosity multiplier always exist. The permeability multipliers (third and fourth columns) are optional.
6. Pressure in the table should increase monotonically down the column.
7. Porosity multipliers should remain same or increase with the increase in pressure.
8. Minimum pressure in the table must be greater than zero.
9. There is only one keyword *CROCKTABE for one rock type.
10. This keyword can not be combined with any other keywords related to the rock compaction model.
11. The *INT table input option does not apply to this table.

EXPLANATION:

Under the keyword *CROCKTABE, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods. The maximum pressure in the table must be a yield point pressure of that rock type. When pressure in a rock is lower than its yield point pressure, the rock behaves elasticity or reversibility on that path.

Dilation Rock Table in the Plastic Zone (Optional)

***CROCKTABD**

PURPOSE:

This keyword is used for the rock dilation model only. It is optionally entered and located after keyword *CROCKTABE. The keyword indicates that rock behavior is acting in the plastic region when pressure in the rock is higher than its yield point pressure.

FORMAT:

```
*CROCKTABD
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

press

Minimum pressure is the yield point pressure of a rock type (kPa | psi) which is equal to maximum pressure in the *CROCKTABE table.

por_mult

Porosity multiplier (dimensionless) = V/Vi

where:

V is the pore volume measured at the pressure " press"

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

Horizontal permeability multiplier (dimensionless) = permh/permhi

where:

permh is the horizontal permeability measured at the pressure " press".

permhi is the input (initial) horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

Vertical permeability multiplier (dimensionless) = permv/permvi

where:

permv is the vertical permeability measured at the pressure " press".

permvi is the input (initial) vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default value for 'press' or 'por_mult' .

Default value of ‘ver_perm_mult’ is ‘hor_perm_mult’.

Default value of ‘hor_perm_mult’ is one.

If a value of hor_per_mult is less than zero, it is replaced by a value of one.

If a value of ver_per_mult is less than zero, it is replaced by a value of hor_perm_mult on the same row.

CONDITIONS:

When the keyword *CROCKTABD appears, some conditions apply:

1. This keyword must be located in Reservoir Description section.
2. This keyword is optional and located under the keyword *CROCKTABE.
3. The first two columns in the table such as pressure and porosity multiplier always exist. The permeability multipliers (third and fourth columns) are optional.
4. Pressure in the table should increase monotonically down the column.
5. Porosity multipliers should remain same or increase with the increase in pressure.
6. First pressure in the table must be the same as that of the last pressure in the *CROCKTABE table for the same rock type.
7. There is only one keyword *CROCKTABD for one rock type.
8. This keyword can not be combined with any other keywords related to the rock compaction model.
9. The *INT table input option does not apply to this table.

EXPLANATION:

Under the keyword *CROCKTABD, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods. The minimum pressure in the table must be a yield point pressure of that rock type and that pressure must be the same as that of the maximum pressure in the *CROCKTABE table. When pressure in a rock is continuously increasing, the rock behaves plasticity on the path given by the *CROCKTABD table. When pressure in the rock goes beyond the maximum pressure in the table, an extrapolation method would be applied. However, it is strongly recommended that the maximum pressure in the table should be large enough so that extrapolation does not occur.

Dilation Rock Table in Unloading Zone (Optional) *CROCKTABU

PURPOSE:

This keyword is used for the rock dilation model only. It must be entered after

1. keyword *CROCKTABD, if it is the first *CROCKTABU table in a *CROCKTYPE, after
2. *CROCKTABU or
3. after *CROCKTABR. The keyword indicates that rock behavior is irreversible in the plastic region when pressure in the rock is decreasing.

FORMAT:

```
*CROCKTABU
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

press

Maximum pressure is one of the pressures in the *CROCKTABD table (kPa | psi). Minimum pressure must be greater than zero.

por_mult

Porosity multiplier (dimensionless) = V/Vi where:

V is the pore volume measured at the pressure " press"

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

Horizontal permeability multiplier (dimensionless) = permh/permhi where:

permh is the horizontal permeability measured at the pressure " press".

permhi is the input (initial) horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

Vertical permeability multiplier (dimensionless) = permv/permvi where:

permv is the vertical permeability measured at the pressure " press".

permvi is the input (initial) vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default value for 'press' or 'por_mult' .

Default value of 'ver_perm_mult' is 'hor_perm_mult'.

Default value of 'hor_perm_mult' is one.

If a value of hor_per_mult is less than zero, it is replaced by a value of one.

If a value of ver_per_mult is less than zero, it is replaced by a value of hor_perm_mult on the same row.

CONDITIONS:

This keyword must be entered and located under the keyword *CROCKTABE.

1. When the keyword *CROCKTABU appears, some conditions apply:
2. This keyword must be located in Reservoir Description section.
3. The first two columns in the table such as pressure and porosity multiplier always exist. The permeability multipliers (third and fourth columns) are optional.
4. Pressure in the table should decrease monotonically down the column.
5. Porosity multipliers should remain same or decrease with the decrease in pressure.
6. First pressure in each table must be the same as one of pressures in the *CROCKTABD table for the same rock type.
7. There must be at least two *CROCKTABU tables for one rock type.
8. Each subsequent *CROCKTABU table must start at a pressure higher than the last.
9. The minimum pressure in the table must be greater than zero.
10. This keyword can not be combined with any other keywords related to the rock compaction model.
11. The *INT table input option does not apply to this table.

EXPLANATION:

Under the keyword *CROCKTABU, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods applied to two unloading paths given by two *CROCKTABU tables for one rock type. On the dilation path, when pressure in the rock suddenly decreases, the rock behavior would follow an arbitrary unloading path that lies between two given unloading tables. If the pressure keeps on decreasing, the rock follows the unloading path until it reaches the minimum pressure in the table. When pressure in the rock increases, the rock will follow the same path until it reaches the dilation path. When on the unloading path and the pressure increases above the pressure at the intersection between unloading path and dilation path the rock will follow the dilation path. It is valid to model dilation using one *CROCKTABE table, one *CROCKTABD table and two or more *CROCKTABU tables. Recompaction and reloading may also be modeled by using the *CROCKTABC and *CROCKTABR tables which will be described on their own keyword.

If pressure in the rock decreases but it is less than the first pressure in the first *CROCKTABU table or greater than the last pressure in the first *CROCKTABU table, the rock behavior is still following the dilation path and the CROCKTABD table is used to compute multiplier vs. pressure. The rock only follows an arbitrary unloading path when pressure in the rock lies between the first pressure in the first and last *CROCKTABU tables.

Dilation Rock Table in Recompacting Zone (Optional)

***CROCKTABC**

PURPOSE:

This keyword is used for the rock dilation model only and is optionally entered after keyword *CROCKTABU. The keyword indicates that rock behavior is recompaction in the unloading zone when pressure in the rock is decreasing below the minimum pressure in the unloading *CROCKTABU table.

FORMAT:

```
*CROCKTABC
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

press

Maximum pressure is equal to minimum pressure in *CROCKTABU table (kPa | psi).

por_mult

Porosity multiplier (dimensionless) = V/Vi

where:

V is the pore volume measured at the pressure " press"

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

Horizontal permeability multiplier (dimensionless) = permh/permhi

where:

permh is the horizontal permeability measured at the pressure " press".

permhi is the input (initial) horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

Vertical permeability multiplier (dimensionless) = permv/permvi

where:

permv is the vertical permeability measured at the pressure " press".

permvi is the input (initial) vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default value for 'press' or 'por_mult' .

Default value of ‘ver_perm_mult’ is ‘hor_perm_mult’.

Default value of ‘hor_perm_mult’ is one.

If a value of hor_per_mult is less than zero, it is replaced by a value of one.

If a value of ver_per_mult is less than zero, it is replaced by a value of hor_perm_mult on the same row.

CONDITIONS:

When the keyword *CROCKTABC appears, some conditions apply:

1. This keyword must be located in Reservoir Description section.
2. The keyword is optional. It should be entered and located under the keyword *CROCKTABU.
3. The first two columns in the table such as pressure and porosity multiplier always exist. The permeability multipliers (third and fourth columns) are optional.
4. Pressure in the table should decrease monotonically down the column.
5. Porosity multipliers should remain same or decrease with the decrease in pressure.
6. First pressure in the table must be the same as that of the last pressure in the *CROCKTABU table for the same rock type.
7. There must be at least two *CROCKTABC tables for one rock type corresponding to at least two *CROCKTABU tables.
8. The minimum pressure in the table must be greater than zero.
9. This keyword can not be combined with any other keywords related to the rock compaction model.
10. The *INT table input option does not apply to this table.

EXPLANATION:

Under the keyword *CROCKTABC, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods applied to two recompaction paths given by two *CROCKTABC tables for one rock type. On the unloading path, when pressure in the rock is below the minimum pressure in the *CROCKTABU table, the rock behavior would follow an arbitrary recompaction path that lies between the two given *CRACKTABC tables. If the pressure keeps on decreasing, the rock follows the path until it reaches the minimum pressure in the table.

The use of the *CROCKTABU card, *CROCKTABC card and the *CROCKTABR card defines a complete “Unloading Recompaction Reloading” path. When *CROCKTABC and *CROCKTABR are used with *CROCKTABU tables, each *CROCKTABU table (unloading curve) must be followed by a *CROCKTABC table (recompaction curve) and a *CROCKTABR table (reloading curve).

Dilation Rock Table in Reloading Zone (Optional) *CROCKTABR

PURPOSE:

This keyword is used for the rock dilation model only. It must be entered and located after existing keyword *CROCKTABC. The keyword indicates that rock behavior is in a reloading state when pressure in the rock is increasing in the recompaction zone.

FORMAT:

```
*CROCKTABR
  press  por_mult  hor_perm_mult  ver_perm_mult
```

DEFINITIONS:

press

Minimum pressure in this table is equal to minimum pressure in the *CROCKTABC table (kPa | psi). Maximum pressure in the table is pressure of the intersection point between this reloading path and the dilation path.

por_mult

Porosity multiplier (dimensionless) = V/Vi

where:

V is the pore volume measured at the pressure " press"

Vi is the initial pore volume. It is assumed that the porosity array (*POR) is defined at the initial pressure.

hor_perm_mult

Horizontal permeability multiplier (dimensionless) = permh/permhi

where:

permh is the horizontal permeability measured at the pressure " press".

permhi is the input (initial) horizontal permeability. Permhi is *PERMI when flow in the I direction is considered, and is *PERMJ when flow in the J direction is considered.

ver_perm_mult

Vertical permeability multiplier (dimensionless) = permv/permvi

where:

permv is the vertical permeability measured at the pressure " press".

permvi is the input (initial) vertical permeability. Permvi is the array defined by *PERMK.

DEFAULTS:

Optional keyword. No default value for 'press' or 'por_mult' .

Default value of 'ver_perm_mult' is 'hor_perm_mult'.

Default value of 'hor_perm_mult' is one.

If a value of hor_per_mult is less than zero, it is replaced by a value of one.

If a value of ver_per_mult is less than zero, it is replaced by a value of hor_perm_mult on the same row.

CONDITIONS:

This keyword must be entered and located under the keyword *CROCKTABC. This keyword exists only when *CROCKTABC exists.

When the keyword *CROCKTABR appears, some conditions apply:

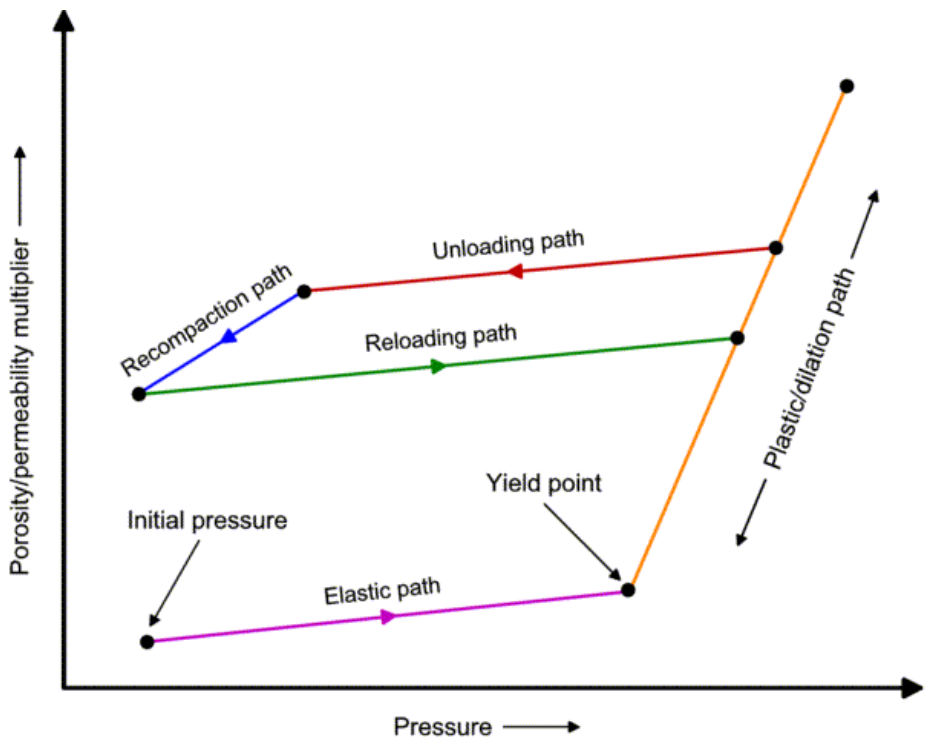
1. This keyword must be located in Reservoir Description section.
2. The first two columns in the table such as pressure and porosity multiplier always exist. The permeability multipliers (third and fourth columns) are optional.
3. Pressure in the table should increase monotonically down the column.
4. Porosity multipliers should remain same or decrease with the decrease in pressure.
5. First pressure in the table must be the same as that of the last pressure in the *CROCKTABC table for the same rock type.
6. The pressure in the last row of the table must lie on the dilation path and must be lower than the first pressure on the *CROCKTABU in its "Unloading Recompaction Reloading" path.
7. There must be at least two *CROCKTABR tables for one rock type.
8. The minimum pressure in the table must be greater than zero.
9. This keyword can not be combined with any other keywords related to the rock compaction model.
10. The *INT table input option does not apply to this table.

EXPLANATION:

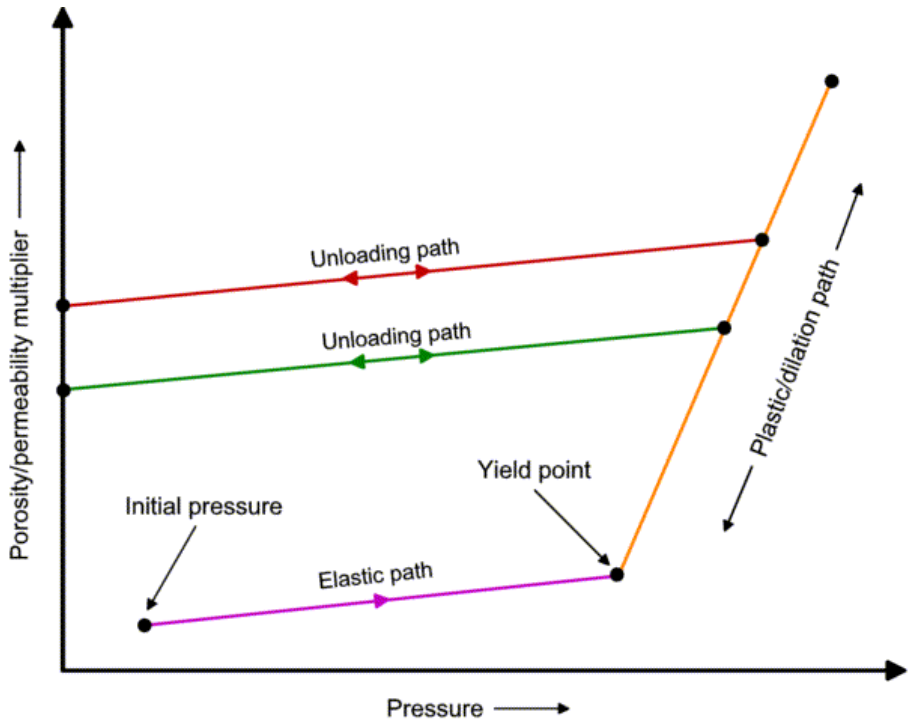
Under the keyword *CROCKTABR, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods applied to two unloading paths given by two *CROCKTABR tables for one rock type. On the recompaction path, when pressure in the rock increases, the rock behavior would follow an arbitrary reloading path that lies between the two given *CROCKTABR tables. If the pressure keeps on increasing, the rock follows the path until it reaches the dilation path. The rock behavior will jump over the dilation plasticity path if pressure in the rock is larger than pressure at the intersection between the reloading path and the dilation path. When pressure in the rock decreases, the rock will follow the reloading path until it meets the recompaction path. If pressure in the rock is less than pressure at the intersection point between the recompaction path and the reloading path, the rock will follow the recompaction path.

GENERAL RESTRICTIONS

1. For one rock type, it is noted that if a closed loop contains keywords *CROCKTABU, *CROCKTABC and *CROCKTABR, other loops in that rock type must also include those keywords (see Model 1). If an open loop contains one keyword *CROCKTABU, other loops in that rock type also contain that keyword only (see Model 2). A mix of model 1 and model 2 in one rock type is not allowed.
2. In one reservoir, either the compaction model is used or the dilation model is used but both models can not be applied simultaneously.
3. It is strongly recommended that the compaction model is used when initial pressure in a reservoir is high and decreasing due to production. Whereas the dilation model is used when initial pressure in the reservoir is low and increasing due to injection. An improper selection of the models for a project may affect results.



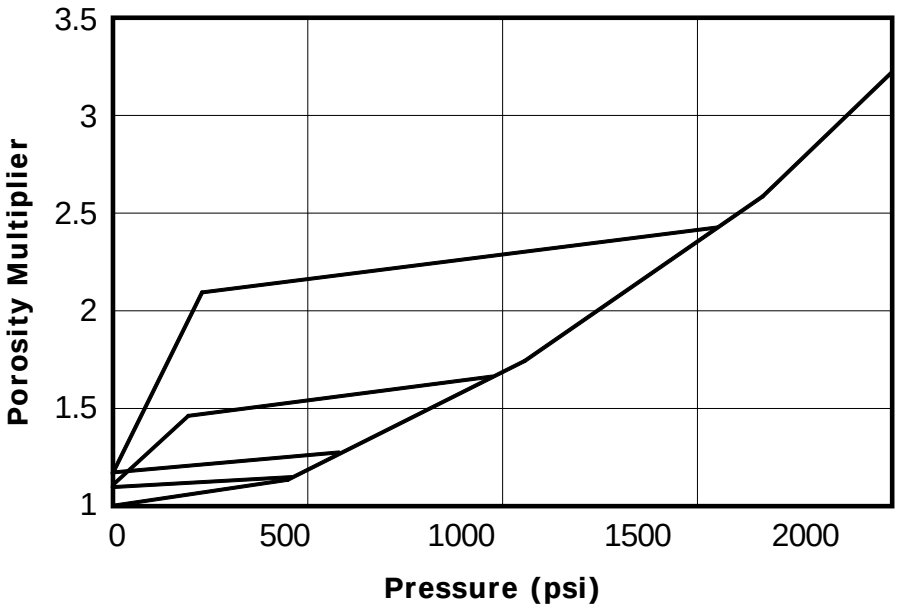
MODEL 1: A Closed Loop Dilation Model



MODEL 2 : An Open Loop Dilation Model

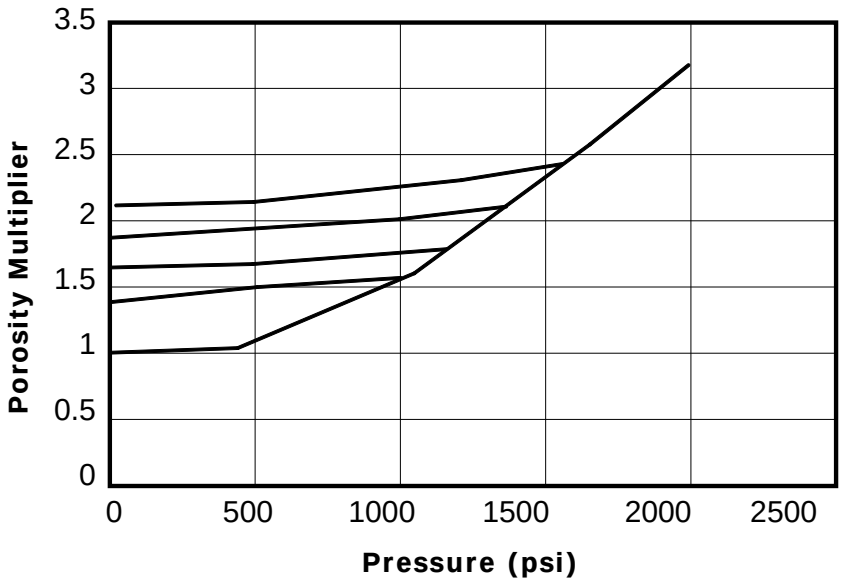
```
Example1:  A sample data of closed loop dilation model is given as
below:
*CROCKTYPE 1
*CROCKTABE
** pressure   por.mult     hor.trans.mult  ver.trans.mult
     14.7     1                   1               1
    100       1.008566484         1               1
    200       1.018702746         1               1
    300       1.028940879         1               1
    400       1.039281906         1               1
*CROCKTABD
    400       1.039281906         1               1
    500       1.114638347         1               1
    600       1.195458746         1               1
    800       1.375104874         1               1
   1000       1.581747107         1               1
   1200       1.819442253         1               1
   1400       2.092856753         1               1
   1600       2.407358287         1               1
   2000       3.185247519         1               1
```

```
*CROCKTABU
** pressure   por.mult     hor.trans.mult  ver.trans.mult
   1000       1.581747107          1               1
   900        1.56600846           1               1
   800        1.550426416          1               1
   700        1.534999415          1               1
   600        1.519725916          1               1
   500        1.50460439           1               1
   400        1.489633326          1               1
   300        1.474811227          1               1
   200        1.46013661           1               1
*CROCKTABC
   200        1.46013661           1               1
   100        1.239838343          1               1
    90        1.219725828          1               1
    50        1.142485901          1               1
     0        1.0527776            1               1
*CROCKTABR
     0        1.0527776            1               1
   100        1.063358191          1               1
   200        1.074045118          1               1
   300        1.084839451          1               1
   400        1.095742269          1               1
   488.2      1.105446145          1               1
*CROCKTABU
  1600        2.407358287          1               1
  1400        2.3596894            1               1
  1200        2.312964419          1               1
  1000        2.267164655          1               1
   800        2.222271787          1               1
   600        2.178267857          1               1
   400        2.135135264          1               1
   200        2.092856753          1               1
*CROCKTABC
   200        2.092856753          1               1
   100        1.541460769          1               1
    90        1.495036423          1               1
    50        1.322905928          1               1
     0        1.135338718          1               1
*CROCKTABR
     0        1.135338718          1               1
   100        1.146749062          1               1
   200        1.158274082          1               1
   300        1.16991493           1               1
   400        1.18167277           1               1
   500        1.193548779          1               1
   614        1.207233087          1               1
```




```
Example2:  A sample data of open loop dilation model is given as below:
*CROCKTYPE 1
*CROCKTABE
** pressure   por.mult     hor.trans.mult  ver.trans.mult
     14.7     1
    100       1.00856648
    150       1.01362194
    200       1.01870275
    250       1.02380901
    300       1.02894088
    350       1.03409847
              1.03928191
*CROCKTABD
    400       1.03928191
    500       1.11463835
    600       1.19545875
    800       1.37510487
   1000       1.58174711
   1200       1.81944225
   1400       2.09285675
   1600       2.40735829
   2000       3.18524752
```

```
*CROCKTABU
** pressure   por.mult     hor.trans.mult  ver.trans.mult
   1000       1.58174711
    800       1.55042642
    700       1.53499942
    600       1.51972592
    500       1.50460439
    400       1.48963333
    300       1.47481123
    200       1.46013661
    100       1.44560801
     50       1.43839801
      1       1.4313671
      0.5     1.43129553
*CROCKTABU
   1200       1.81944225
   1000       1.78341488
    800       1.7481009
    600       1.71348619
    500       1.69643671
    400       1.67955689
    300       1.66284502
    200       1.64629943
    100       1.62991848
     50       1.62178923
      1       1.6138619
      0.5     1.61378121
*CROCKTABU
   1400       2.09285675
   1200       2.05141541
   1000       2.01079467
    800       1.97097826
    600       1.93195028
    400       1.8936951
    300       1.87485252
    200       1.85619743
    100       1.83772795
     50       1.82856225
      1       1.81962421
      0.5     1.81953323
*CROCKTABU
   1600       2.40735829
   1400       2.3596894
   1200       2.31296442
   1000       2.26716465
    800       2.22227179
    600       2.17826786
    300       2.11389031
    200       2.09285675
    100       2.07203248
     50       2.06169818
      1       2.05162056
      0.5     2.05151799
```


3.5
3
2.5
2
1.5
1
0.5
0
Porosity Multiplier
0
500
1000
1500
2000
2500
Pressure (psi)

Treat CCPOR / CROCKTAB / CROCKTABH as Depletion Parameters / Tables (Optional) *DEPLETION

PURPOSE:

The *DEPLETION option makes the input reference porosity (*POR) the actual input porosity and makes the porosity a function of pressure depletion rather than pressure.

The *DEPLETION keyword in the data set indicates that all *CROCKTYPES using *CCPOR (*CPROR), *CROCKTAB and *CROCKTABH use the Depletion Option.

FORMAT:

*DEPLETION

DEFINITIONS:

*DEPLETION
Occurrence of this keyword turns the Depletion option on for all *CROCKTYPES using *CCPOR (*CPROR), *CROCKTAB and *CROCKTABH.

DEFAULTS:

Optional keyword. No default.

CONDITIONS:

This keyword must be located in the Reservoir Description keyword group, just after the first *CROCKTYPE keyword.

Dilation Tables (*CROCKTABE/D/U/C/R) do not make use of this option.

Any grid blocks not explicitly defined in a *CTYPE region will also use the Depletion option (using *CPOR and *PRPOR), if (1) *DEPLETION is read in and if (2) at least one block is assigned to a valid *CROCKTYPE which can use the Depletion option.

The Depletion option will work when a *CROCKTAB table uses the *CIRREVERS option.

EXPLANATION:

The Depletion option is in reality two separate functions.

In the Case of models where *CCPOR and *CPRPOR are used, the value of *CPRPOR for each block is overwritten by the initial reservoir pressure in each block (Prs0(iblk)). This has the effect of changing the porosity reference pressure of each block to be the initial block pressure.

Hence *POR input becomes the actual measured porosity of the block, not the porosity at a reference pressure *CPRORP (actually every block has its own CPRPOR which has been set equal to Prs0(iblk)).

In the Case of Models where *CROCKTAB and *CROCKTABH tables are used (*CROCKTAB and the *CIRREVERS option may also be used), the use of the Depletion option operates in a completely different manner. When the Depletion option is used with

*CROCKTAB, the compaction tables are interpolated with respect to P^* = Pshift(itype) + Prs(iblk) - Prs0(iblk).

Where Prs(iblk) is the current grid block pressure, Prs0(iblk) is the initial grid block pressure, “itype” is the *CROCKTYPE of grid block “iblk” and Pshift(itype) is the pressure in the *CROCKTAB table of *CROCKTYPE itype where the Porosity Multiplier (por_mult) is equal to or closest to 1.00.

GEM will interpolate within the *CROCKTAB table to find the best value of Pshift.

As P^* rather than Prs(iblk) is used in the interpolation procedure to find the value of por_mult, this procedure initially calculates every grid blocks por_mult to be the value at Pshift(itype). This causes the input reference porosity (*POR) to be the measured porosity (as por_mult = 1.00).

The value of por_mult for each grid block calculated as depletion occurs, using this procedure, is calculated in the tables using a pressure offset from Pshift(itype) by Prs(iblk) – Prs0(iblk).

Therefore, it is how much pressure in a block has depleted from the initial pressure which determines por_mult and hence compressibility, rather than the actual pressure itself.

When the Depletion option is used, GEM will printout the value of Pshift(itype) it determines for each ‘itype’ containing a *CROCKTAB table as well as diagnostics on how it arrived at the value chosen. To make the choice (by interpolation) of Pshift as straightforward as possible, the user should use values of por_mult which increase with pressure and which start at a por_mult value below 1.0 and end at values either equal to or greater than 1.0. Por_mult should only increase with pressure.

Compaction/Dilation Rock Region (Optional) *CTYPE

PURPOSE:

*CTYPE assigns compressibility/compaction/dilation rock types to reservoir grid blocks.

ARRAY:

*CTYPE

DEFAULTS:

Optional keyword. The default is *CTYPE *CON 1 (all blocks use compressibility/ compaction/dilation rock type 1).

CONDITIONS:

This keyword must be located in the Reservoir Description section (or in the Recurrent Data section with limited functionality). The *DYNAGRID keywords cannot be used with *CTYPE.

Number of *CTYPE regions must be less than or equal to number of *CROCKTYPE.

CONDITIONS FOR USE IN RECURRENT DATA:

In the Recurrent Data section, *CTYPE can be used to switch from one *CROCKTAB table to another *CROCKTAB tables (any number of tables and any number of times). In order to change *CTYPE in recurrent data, (1) all *CTYPEs must contain *CROCKTAB tables, (2) all blocks must be assigned to *CROCKTAB tables and (3) all porosity multipliers in each of the tables which are switched between must be identical. Using this option, permeability multipliers can be added at times other than the start of the run.

The *DYNAGRID keywords cannot be used with *CTYPE in recurrent data.

If grid refinement is used in a model with *CTYPE in recurrent data, automatic inheritance of *CTYPE does not occur in recurrent data (only when *CTYPE is in the initial reservoir description section does this occur). Thus it is always necessary to explicitly refer to the children blocks when altering *CTYPE.

When *CROCKTAB tables are changed in recurrent data, permeability modification will occur immediately based on the pressure of the previous timestep.

The new tables being switched TO, may contain either the *CIRREVRS keyword or *CROCKTABH tables to model hysteresis.

It is not recommended (but possible) to switch FROM a table with hysteresis or which uses *CIRREVERS. All past hysteresis information will be lost. The switch will place the block on the drainage curve of the new table regardless of whether the block was previously on the drainage or imbibition curve when using the old table.

EXPLANATION:

This keyword assigns rock properties defined by *CROCKTYPE keywords to specific grid block ranges. *CTYPE indicates the input of an array of grid values. All array reading subkeywords are valid. See the section on Array Reading Options for more details.

Examples:

The following are examples of a reservoir consisting of 50X50X4 grid blocks.

Example 1:

```
*CROCKTYPE    1
*CROCKTAB: table of rock compaction data for rock type 1
**pressure    porosity          perm.                  perm.
             multiplier    multiplier (hoziz.)    multiplier
(vert.)
   1000         0.96             0.98                  1.2
   2000         0.99             0.99                  1.3
   3000         1.0              1.0                   1.4
   4000         1.01             1.0                   1.5
*CROCKTYPE    2
*CROCKTAB
*CIRREVERS
**pressure     porosity multiplier
   1000               0.96
   2000               0.97
   3000               0.98
   4000               0.99
*CROCKTYPE    3
*CROCKTAB
**pressure     porosity multiplier     perm. multiplier
   1000               0.96                   0.98
   2000               0.99                   0.99
   3000               1.0                    1.0
   4000               1.01                   1.0
*CROCKTABH
   1000               0.96                   0.98
   2000               0.97                   0.985
   3000               0.98                   0.99
   4000               0.99                   0.995
*CROCKTABH
   2000               0.99                   0.99
   3000               0.995                  0.995
   4000               0.998                  0.998
*CROCKTABH
   3000               1.0                    1.0
   4000               1.01                   1.0
*CROCKTYPE    4
    *CCPOR        3.0e-6
    *CPRPOR      14.7
*CTYPE   *KVAR  1 2 3 4   ** assign different rock type
                          ** to each k layer
```

In the above example, columns 3 and 4 do not appear in the data table of rock type 2. In this case the permeability multipliers in vertical and horizontal directions are default to be one.

Moreover, column 4 does not appear in rock type 3, in such a case, values of permeability multipliers are assumed to be the same in both vertical and horizontal directions.

Example 2: (using the same model 50X50X4):

```
*CPOR    3.0e-6    ** This keyword was discussed in the manual.
*PRPOR  14.7       ** This keyword was discussed in the manual.
*CROCKTYPE    1
   *CCPOR      *MATRIX      2.5e-6
   *CCPOR      *FRACTURE    2.0e-5
   *CPRPOR     *MATRIX     14.7
   *CPRPOR     *FRACTURE   14.7
*CTYPE   *MATRIX    *IJK   1:50   1:50     3:4    1
*CTYPE   *FRACTURE  *IJK   1:50   1:50     3:4    1
```

In example 2 above, k layers 1 and 2 were not explicitly defined to belong in *CTYPE region 1. Thus they will use the global values of *CPOR and *PRPOR in their porosity calculation for both matrix and fracture. Layers 3 and 4 will use the separate matrix and fracture values of *CCPOR and *CPRPOR defined in *CTYPE 1.

The *CPOR and *PRPOR values are used for layer 1 and 2 because *CTYPE was found, and layers 1 and 2 were omitted from any *CTYPE region.

If *CTYPE would have been defaulted in example 2 (i.e. *CTYPE keyword omitted), all blocks, including those in layer 1 and 2, would have been assigned to *CTYPE 1. In this case *CPOR and *PRPOR would only be used in the analytical aquifer calculation (if required).

Example 3: (using two rock models)

This example shows how two rock models can be used in one simulation.

Data for rock dilation model:

```
*CROCKTYPE 1
*CROCKTABE
** pressure   por.mult     hor.trans.mult  ver.trans.mult
     14.7     1            1               1
    100       1.0085665    1.0160131       1
    150       1.0136219    1.0255614       1
    200       1.0187027    1.035232        1
    250       1.023809     1.0450268       1
    300       1.0289409    1.0549481       1
    350       1.0340985    1.064998        1
    400       1.0392819    1.0751787       1
*CROCKTABD
    400       1.0392819    1.0751787       1
    500       1.1146383    1.2327992       1
    600       1.1954587    1.424045        1
    800       1.3751049    1.9523725       1
   1000       1.5817471    2.8069041       1
   1200       1.8194423    4.3221774       1
   1300       1.9513667    5.56285         1
   1400       2.0928568    7.4048444       1
   1500       2.244606    10.3099231       1
*CROCKTABU
    600       1.1954587    1.424045        1
    550       1.1894964    1.409072        1
    500       1.1835637    1.3943166       1
    450       1.1776607    1.3797745       1
    400       1.1717871    1.3654418       1
    300       1.1601276    1.3373887       1
    250       1.1543414    1.3236606       1
    200       1.1485841    1.3101264       1
```

```
*CROCKTABC
    200       1.1485841    1.3101264       1
    150.0003  1.1133907    1.2300356       1
    100.0005  1.0792756    1.1565273       1
     50.0008  1.0462059    1.088905        1
      0.001   1.0141494    1.0265618       1
*CROCKTABR
      0.001   1.0141494    1.0265618       1
     85.1742  1.0228241    1.0431316       1
    170.3474  1.031573     1.0600671       1
    255.5205  1.0403968    1.077379        1
    340.6937  1.049296     1.0950784       1
    425.8669  1.0582714    1.1131769       1
*CROCKTABU
   1000       1.5817471    2.8069041       1
    800       1.5504264    2.6555066       1
    700       1.5349994    2.5841702       1
    600       1.5197259    2.5155461       1
    500       1.5046044    2.4494987       1
    400       1.4896333    2.3859011       1
    300       1.4748112    2.3246338       1
    200       1.4601366    2.2655846       1
*CROCKTABC
    200       1.4601366    2.2655846       1
    150.0003  1.3454868    1.853819        1
    100.0005  1.2398394    1.5401945       1
     50.0008  1.1424873    1.2959291       1
      0.001   1.0527793    1.1020726       1
*CROCKTABR
      0.001   1.0527793    1.1020726       1
     97.6353  1.0631084    1.1230351       1
    195.2696  1.0735388    1.1445465       1
    292.9039  1.0840716    1.1666257       1
   390.5383   1.0947077    1.1892928       1
   488.1726   1.1054481    1.2125686       1
*CROCKTABU
  1200        1.8194423    4.3221774       1
  1000        1.7834149    4.0419251       1
   800        1.7481009    3.7873389       1
   600        1.7134862    3.5553807       1
   500        1.6964367    3.4470591       1
   400        1.6795569    3.3434488       1
   300        1.662845     3.244281        1
   200        1.6462994    3.1493054       1
*CROCKTABC
   200        1.6462994    3.1493054       1
   150.0003   1.4804376    2.3476953       1
   100.0005   1.3312861    1.8083243       1
    50.0008   1.1971613    1.4283473       1
     0.001    1.0765494    1.1508205       1
*CROCKTABR
     0.001     .0765494    1.1508205       1
   105.0777   1.0879211    1.174786        1
   210.1545   1.0994129    1.1994412       1
   315.2312   1.111026     1.2248124       1
   420.308    1.1227619    1.2509275       1
   525.3847   1.1346217    1.2778154       1
```

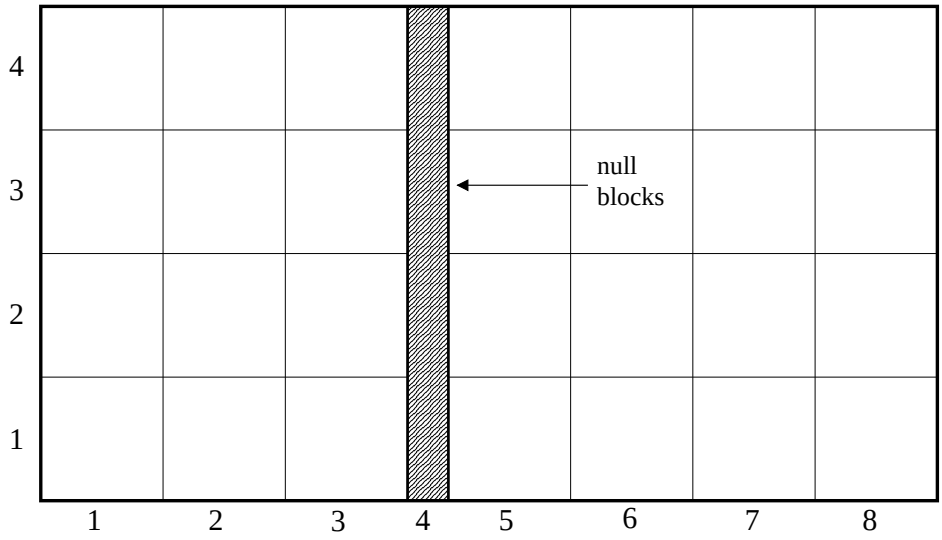
```
*CROCKTABU
  1400        2.0928568    7.4048444       1
  1300        2.0720325    7.091086        1
  1200        2.0514154    6.7965223       1
  1000        2.0107947    6.258994        1
   800        1.9709783    5.7816897       1
   600        1.9319503    5.3559411       1
   300        1.8748525    4.7986518       1
   200        1.8561974    4.6316616       1
*CROCKTABC
   200        1.8561974    4.6316616       1
   150.0003   1.6300455    3.0589838       1
   100.0005   1.431447     2.1545655       1
    50.0008   1.257045     1.587556        1
     0.001    1.1038914    1.2091707       1
*CROCKTABR
     0.001    1.1038914    1.2091707       1
   113.438    1.1164849    1.2368996       1
   226.8749   1.1292221    1.265511        1
   340.3119   1.1421046    1.2950425       1
   453.7489   1.155134     1.3255336       1
   567.1858   1.1683121    1.3570259       1
```

Data for rock compaction model:

```
*CROCKTYPE    2               ** rock type number 2
 *CROCKTAB                    ** data table for main path
** press    por.mult   perm.mult (hor)
    14.7    1.0        1.0    ** press : pressure
   100.0    1.000256   1.001  ** por.mult : porosity multiplier
   200.0    1.000556   1.002  ** perm.mult (hor): hor. perm. multiplier.
   500.0    1.001456   1.005  ** Vertical permeability multiplier is the
  1000.     1.002956   1.01   ** same as horizontal perm. multiplier
when
  2000.     1.005956   1.02   ** column 4 does not appear.
  5000.     1.014956   1.05
  7000.     1.020956   1.07
 10000.     1.029956   1.1
 20000.     1.059956   1.2
*CROCKTABH
  1000.0   1.002956   1.01
  2000.0   1.004956   1.02
  3000.0   1.006956   1.03
  4000.0   1.008956   1.04
  5000.0   1.010956   1.05
  6000.0   1.012956   1.06
  7000.0   1.014956   1.07
  8000.0   1.016956   1.08
 *CROCKTABH
  2000.0   1.005956   1.02
  3000.0   1.007956   1.03
  4000.0   1.009956   1.04
  5000.0   1.011956   1.05
  6000.0   1.013956   1.06
  7000.0   1.015956   1.07
  8000.0   1.017956   1.08
```

```
*CROCKTABH
  5000.0   1.014956   1.05
  6000.0   1.016956   1.06
  7000.0   1.018956   1.07
  8000.0   1.020956   1.08
```

Two reservoirs connected by null blocks are shown in the figure below:



Assume there is only one layer K=1, the above two rock models are assigned in the field as follows:

```
*CTYPE  *IJK
        1:3    1:4    1         1
        5:8    1:4    1         2
-or-
*CTYPE  *IJK
        1:4    1:4    1         1
        4:8    1:4    1         2
```

The above application is valid since each reservoir will used one rock model only.

However, invalid cases can be seen as follows:

```
*CTYPE  *IJK
        1:3    1:4    1         1
        3:8    1:4    1         2
-or-
*CTYPE  *IJK
        1:8    1:2    1         1
        1:8    3:4    1         2
```

As a general rule, if two adjacent active blocks have different rock models to each other, the simulator will stop running.

Palmer and Mansoori Parameters (Optional) *POISSR, *YOUNGM, *STRINF, *PRESLN, *EXPPM

PURPOSE:

These keywords are used to enter the Palmer & Mansoori model parameters. The Palmer and Mansoori model accounts for change in fracture porosity and absolute permeability due to change in net overburden stress and matrix shrinkage/swelling. Matrix shrinkage or swelling occurs as a result of desorption or adsorption of gaseous species.

FORMAT:

*POISSR	ν
*YOUNGM	E
*STRINF	ε_L
*PRESLN	p_L
*EXPPM	α

DEFINITIONS:

*POISSR

Keyword specifying a value for the Poisson ratio required for the P&M model and its multi component variants. Dimension less value.

*YOUNGM

Keyword used to specify a value for the Young's modulus used in calculating the pore compressibility for the P&M model and its multi component variants. For SI units enter a value in kPa and for field units enter a value in psia.

*STRINF

Original Palmer and Mansoori Model:

Keyword used to specify a value for the strain at infinite pressure required in the Palmer and Mansoori equation. The strain is the fractional change in volume divided by the initial volume and therefore has no units.

Multi component extensions of the Original Palmer and Mansoori Model:

In this case the *STRINF keyword should be followed by a value for the strain at infinite pressure for each adsorbing component. The values will be assigned to specific components according to the component sequence specified with the *COMPNAMES keyword. The strain is the fractional change in volume divided by the initial volume and therefore has no units. As an example consider the following:

```
*NC 3
*COMPNAMES CH4, N2, CO2
*STRINF 0.0045 0.0035 0.0065
```

Implies component CH4 is assigned a value of 0.0045, component N2 a value of 0.0035 and component CO2 a value of 0.0065.

*PRESLN

Original Palmer and Mansoori Model:

Keyword specifying a value for the Langmuir pressure required in the Palmer and Mansoori equation. For SI units enter a value in kPa and in field units enter a value in psia.

Multi component extensions of the Original Palmer and Mansoori Model:

In this case the *PRESLN keyword should be followed by a value for the Langmuir pressure for each adsorbing component. For SI units enter a value in kPa and in field units enter a value in psia.

The values will be assigned to specific components according to the component sequence specified with the *COMPNAMES keyword.

*EXPPM

Keyword specifying a value for the exponent used in calculating the change in fracture permeability as a function of the change in fracture porosity.

DEFAULTS:

Optional keywords. No default values.

CONDITIONS:

These keywords must be located in the Reservoir Description section. The values entered will be associated with a given compaction/dilation rock type. This number is specified with the *CROCKTYPE keyword which must precede the above keywords. If these keywords do not appear, the simulator will NOT apply the P&M theory for the specific grid blocks associated with that compaction/dilation rock type.

EXPLANATION:

Flow in the fracture system in coal is described by Darcy flow. The absolute permeability appearing in Darcy's Law is not constant but varies in situ with the change in the net overburden stress (overburden pressure – pore pressure) and with effects associated with desorption/adsorption of gas in the matrix. As the gas is desorbed matrix volume shrinks which in turn allows for fracture apertures to open. As the pressure is reduced therefore there are two competing effects, pore closure due compressibility effects and pore enlargement due to matrix shrinkage. The Palmer and Mansoori model accounts for both effects and allows for changes in porosity and absolute permeability to be calculated as a function of changes in fracture pressure and matrix shrinkage/swelling. For derivation and comprehensive discussion of the Palmer and Mansoori model please refer to SPE 39105 and SPE 36737. Equation 1 below reproduces the Palmer and Mansoori relationship with our modification of the first term to avoid negative porosity due to pressure change.

$$\frac{\phi}{\phi_i} = \exp[c_f\,(p - p_i)] + \frac{\varepsilon_L}{\phi_i}\left(1 - \frac{K}{M}\right)\left(\frac{p_i}{p_i + p_L} - \frac{p}{p + p_L}\right) \qquad (1)$$

where:

ϕ_i	:	initial natural fracture porosity
ϕ	:	fracture porosity at pressure p
c_f	:	Fracture pore volume compressibility (1/kPa|1/psia)
p_i	:	initial pressure (kPa|psia)
p	:	pressure (kPa|psia)
ε_L	:	strain at infinite pressure
K	:	bulk modulus (kPa|psia)
M	:	axial modulus (kPa|psia)
p_L	:	Langmuir pressure (kPa|psia)
exp		Exponential function with base e (~2.71828)

The pressures appearing in the pore compressibility term are the fracture pressures and in the matrix shrinkage/swelling term are the corresponding matrix pressures for a given grid block.

The ratio of bulk to axial modulus is related to the Poisson's ratio as indicated by equation 2:

$$\frac{K}{M} = \frac{1}{3}\left(\frac{1+\nu}{1-\nu}\right) \qquad (2)$$

Where ν is the Poisson's ratio (dimensionless)

The pore volume compressibility values can be entered directly using the *CCPOR keyword or calculated from entered parameters as follows, equation 3(a) and 3(b).

$$M = E\frac{(1-\nu)}{(1+\nu)(1-2\nu)} \qquad (3a)$$

Where E is the Young's modulus (kPa | psia) and

$$c_f = \frac{1}{\phi_i M} \qquad (3b)$$

The P&M theory relates the absolute permeability ratio to the porosity ratio in the following manner, equation 4:

$$\frac{k}{k_i} = \left(\frac{\phi}{\phi_i}\right)^{\alpha} \qquad (4)$$

where α is typically equal to 3.

For cases where the adsorbed gas is a mixture, say of methane and carbon dioxide, then it is not clear what the appropriate values of ε_L and p_L should be for use in equation 1. Although both methane and carbon dioxide volumetric strain versus pressure relationships is Langmuir, the values of the parameters ε_L and p_L differ significantly. One possibility is to extend the Palmer and Mansoori model for mixtures using component based values of ε_L and p_L. In this case the Palmer and Mansoori equation can be modified by introducing the generalized multi component Langmuir concept as follows, equation 5:

$$\frac{\phi}{\phi_{ref}} = \exp[c_f(p - p_{ref})] + \frac{1}{\phi_{ref}}\left(1 - \frac{K}{M}\right)\left(\sum_{j=1}^{j=n} \frac{\varepsilon_{Lj} y_{ref,j} p_{ref}/p_{Lj}}{1 + p_{ref}\sum_{k=1}^{k=n} y_{ref,k}/p_{Lk}} - \sum_{j=1}^{j=n} \frac{\varepsilon_{Lj} y_j p/p_{Lj}}{1 + p\sum_{k=1}^{k=n} y_k/p_{Lk}}\right) \quad (5)$$

where:

ϕ_{ref}	:	Reference state natural fracture porosity
ϕ	:	fracture porosity at pressure p
c_f	:	pore volume compressibility (1/kPa | 1/psia)
p_{ref}	:	Reference state pressure (kPa | psia)
p	:	pressure (kPa | psia)
ε_{Li}, ε_{Lk}	:	strain at infinite pressure, component j /component k
K	:	bulk modulus (kPa | psia)
M	:	axial modulus (kPa | psia)
p_{Lj}, p_{Lk}	:	Langmuir pressure (kPa | psia), component j /component k
n	:	Number of adsorbing components
$y_{ref,j}$, $y_{ref,k}$	:	Composition at reference conditions, component j /component k
y_j, y_k	:	Composition of gas mixture, component j /component k
exp		Exponential function with base e (~2.71828)

NOTE: The pressures appearing in the pressure strain (pore compressibility) term are the fracture pressures and in the adsorption strain (matrix shrinkage/swelling) term are the corresponding matrix pressures for a given grid block.

The above equation relates the fracture porosity (ϕ) at pressure "p" and composition "y", state 2, to reference conditions, state 1. State 1 is stipulated by values of fracture pressure, pref, composition, yref and the corresponding porosity, ϕ_{ref}. The specific choice of reference conditions and the manner in which the composition is calculated leads to two formulations.

CMG MODELS

For the CMG models p_{ref} is the initial (reservoir) fracture/matrix pressure and ϕ_{ref} is the corresponding fracture porosity.

*MODEL 1: Use keyword *PALMAN_CMG_COAL in the component properties section of the data file.*

In this case the compositions y_{ref} , y appearing in equation 5 are the free gas matrix block compositions.

*MODEL 1: Use keyword *PALMAN_CMG2_COAL in the component properties section of the data file.*

In this case the compositions y_{ref} , y appearing in equation 5 are the adsorbed gas compositions. The adsorbed gas composition is calculated based on the relative amounts of each species adsorbed. These are obtained by applying the Langmuir multi component adsorption isotherm model.

The following keywords illustrate an example of how reference conditions are specified for these models.

```
** RESERVOIR DESCRIPTION SECTION OF THE DATA FILE
*POR *FRACTURE *CON 0.0012    **porosity at initial res.
                                pressure
** Absolute permeability in md

*PERMI *FRACTURE *CON 3.65    ** permeability at initial res.
                                 pressure
*PERMJ *FRACTURE *CON 3.65
*PERMK *FRACTURE *CON 0.25

*CROCKTYPE 1
** ROCK COMPRESSIBILITY

*CPRPOR *FRACTURE 1146.0 **initial res. pressure 1146 psia

*POISSR   0.21
*YOUNGM   521000
*STRINF   0.0088 0.0078 0.0098
*PRESLN   625.0 725.0 325.0
*EXPPM    3.0

*CTYPE *FRACTURE *CON 1
```

MAVOR/ARC_MODEL

In this case reference condition is at atmospheric pressure and the corresponding fracture porosity and fracture permeability entered should correspond to values adjusted for atmospheric pressure. The composition appearing in equation 5 is the adsorbed gas composition. The gas composition is calculated based on the relative amounts of each species adsorbed. These are obtained by applying the Langmuir multi component adsorption isotherm model. This requires the pressure and the corresponding free gas composition. The free gas composition at atmospheric conditions is assumed to be the same as the initial reservoir gas composition. Since adsorption at atmospheric pressure is much smaller than at reservoir conditions, the particular composition at atmospheric conditions is not crucial to the accuracy of the calculations. To select this model specify keyword *PALMAN_ARC_COAL in the Component Properties section.

The following keywords illustrate an example of how reference conditions are specified for this model.

```
** RESERVOIR DESCRIPTION SECTION OF THE DATA FILE

*POR *FRACTURE *CON 0.00214 **porosity at atmospheric pressure

** Absolute permeability in md

*PERMI *FRACTURE *CON 20.75    **permeability at atmospheric
                                  pressure
*PERMJ *FRACTURE *CON 20.75
*PERMK *FRACTURE *CON 1.42
*CROCKTYPE 1
** ROCK COMPRESSIBILITY

*CPRPOR *FRACTURE 14.7 **psia, reference pressure atmospheric
```

```
*POISSR   0.21
*YOUNGM   521000
*STRINF   0.0088 0.0078 0.0098
*PRESLN   625.0 725.0 325.0
*EXPPM    3.0

*CTYPE *FRACTURE *CON 1
```

Fault Transmissibilities (Optional) *TRANSF

PURPOSE:

Adjusts transmissibilities on a fault basis.

FORMAT:

*TRANSF	'Fault_Name'	fault_trans_mult
<pair or single>	<pair or single>	…

DEFINITIONS:

*TRANSF
: Keyword introducing the fault name, multiplier and fault description.

'Fault_Name'
: A name for this fault (which must be contained within single quotes).

fault_trans_mult
: Transmissibility multiplier for the connections across this fault. Multipliers of this type apply cumulatively to previously applied multipliers.

<pair or single>
: Identifiers for the connections that are to make up this fault. The identifier can either be of the "pair" or "single" variety. A "pair" identifier looks like:

  i1 j1 k1 [*IDIR or *JDIR or *KDIR] i2 j2 k2

  which refers to an existing connection between cells, while a "single" identifier looks like:

  i1 j1 k1 [*IDIR- or *IDIR+ or *JDIR- or *JDIR+ or *KDIR- or *KDIR+]

  which refers to all connections on a certain cell face. Multiple mixed "pair" or "single" identifiers can follow a *TRANSF line.

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group. "Pair"-type identifiers should refer to pairs of cells that actually are connected in the grid direction specified by the given direction identifier (*IDIR, *JDIR or *KDIR). Refined grid cells cannot be referenced in the description of a "pair" or "single" identifier.

EXPLANATION:

This keyword allows the imposition of a single multiplier to a group of connections. The group of connections can be thought of as corresponding to a single fault, and the multiplier as a way of adjusting the sealing properties of this fault. If the descriptor is of "pair" type, the multiplier will be applied to an (existing) connection. If the descriptor is of "single" type, the

multiplier will be applied to all connections that the cell has to other cells on a specified face. The face for the latter is identified using *IDIR-/+, *JDIR-/+ or *KDIR-/+ descriptors. The "-" identifier refers to the face crossed by moving from the cell to its neighbour along the grid direction corresponding to decreasing the appropriate I, J or K index, and the "+" identifier to increasing the appropriate I, J or K index. The multiplier can be 0 if desired, which will eliminate connections.

For non-isothermal simulations, this multiplier applies only to fluid flow, not to heat flow.

Cap Rock Leakage Using Wells (Optional)

***WEL_OPN_THR, *WEL_OPN_MLT, *WEL_OPN_MMX, *WEL_OPN_MMN**

PURPOSE:

These keywords signal the input of parameters that will be used for modelling cap rock fluid leakage using wells. These parameter values are associated with a particular compaction/dilation rock type assigned by a preceding *CROCKTYPE keyword, and this rock type is associated with a particular cap rock region through the assignment of the rock type number to the desired cells via the *CTYPE keyword. To enable this feature, production wells also have to be defined in the chosen cap rock cells using standard well definitions. Additionally, these wells must be declared as cap rock leakage wells using keyword *CRL_WELLS in the Well and Recurrent data section.

The parameters discussed here can be used to enable a pressure-dependent well productivity multiplier that allows fluid leakage (via well production) once a sufficiently large pressure drop (between the cap rock cell and the well's BHP) has existed.

FORMAT:

*WEL_OPN_THR	wel_opn_thr
*WEL_OPN_MLT	wel_opn_mlt
*WEL_OPN_MMX	wel_opn_mmx
*WEL_OPN_MMN	wel_opn_mmn

DEFINITIONS:

wel_opn_thr
: Minimal pressure difference between the cap rock block and the well's BHP that will cause the cap rock seal to break down and become leaky (kPa|psi).

wel_opn_mlt
: Multiplier used to multiply the difference between the current pressure drop between the block and the well, and the value for wel_opn_thr. This product will become a transmissibility multiplier which will be assigned to the well productivity, once it is range limited by the following values (dimensionless).

wel_opn_mmx
: Maximum allowed value for the multiplier calculated above (dimensionless).

wel_opn_mmn
: Minimum allowed value for the multiplier calculated above (dimensionless). This minimum is not applied to a well productivity until the pressure drop exceeds wel_opn_thr, and after that, it is always applied, even if the pressure drop no longer exceeds the threshold.

DEFAULTS:

Optional keywords. No default values.

CONDITIONS:

These keywords must be located in the Reservoir Description section. The values entered must be associated with a compaction/dilation rock type, meaning there must be a preceding *CROCKTYPE keyword, and a rock type number. Also, production wells have to be defined in the appropriate cap rock cells using standard well definitions and declared as cap rock leakage wells using *CRL_WELLS keyword in the Well and Recurrent data section. The usual compaction/dilation option keywords like *CCPOR, *CPRPOR etc. cannot be used in conjunction with the cap rock leakage through wells option. Use of *CROCK keyword in the Rock-Fluid data section, to define rock compressibility associated with rock-type regions (see *RTYPE), is however, permitted.

EXPLANATION:

Reservoirs are initially bounded by competent cap rock that is sealing, meaning that there is no communication between cells on one side and additional permeable media on the other. As pressures change in the reservoir, the pressures on one side of the seal may differ more and more from pressures on the other side; that is, a pressure drop begins to appear across the cap rock. When this pressure drop becomes sufficiently large, the seal provided by the cap rock may be breached, and flow across the breach may occur.

At the moment of seal breakage, the cap rock may still have little conductivity across it. However, since the pressure drop across the cap rock may be driven higher by external influences (even though fluid can now move across it), further seal degeneration could occur, resulting in larger and larger outflow. Eventually there will be some limit to this growth, and there may also be a decline in outflow as the pressure drop becomes smaller as the seal heals somewhat. However, it would never be expected to heal completely.

These keywords allow modelling of fluid loss, primarily gas, to the reservoir surroundings when cap rock has been breached. This feature works in conjunction with *CROCKTYPE and *CTYPE, as well as with standard production wells operating on BHP constraints and declared as cap rock leakage wells through *CRL_WELLS keyword in Well and Recurrent data section. The *CTYPE keyword needs to identify cap rock cells associated with a compaction/dilation rock type for which the above keywords have been set (using *CROCKTYPE), which have also been perforated as standard production wells that have been set to operate on BHP constraints. There is usually one well per cap rock cell. The BHP value is intended to correspond to the pressure on the “other” side of the cap rock, outside of the reservoir. The parameters discussed here can then be used to enable a pressure-dependent well productivity multiplier calculation that allows fluid leakage (via well production) once a sufficiently large pressure drop between the cap rock cell and the well’s BHP has existed.

The new well productivity multiplier is built in the usual way (see the Well and Recurrent Data section). However, a time-varying transmissibility multiplier, WellMult, is also incorporated. WellMult is set to zero initially, and is activated the first time ($p_{block} - p_{BHP}$) exceeds wel_opn_thr. After this point, the following formula is applied:

$$\text{WellMult} = \max(\text{wel_opn_mlt} \times \max((p_{block} - p_{BHP}) - \text{wel_opn_thr}, \text{zero}), \text{wel_opn_mmn})$$

WellMult will then be set to wel_opn_mmx if the above calculation exceeds the latter value. Thus, WellMult depends continuously on the pressure drop across the cap rock, and never drops below wel_opn_mmn, even if ($p_{block} - p_{BHP}$) later become less than wel_opn_thr.

Example:

```
*GRID *VARI 40 40 25
 :
 ** Cap Rock Leakage Using Wells
 *CROCKTYPE 1
   *WEL_OPN_THR 700.
   *WEL_OPN_MLT 10.
   *WEL_OPN_MMX 5.
   *WEL_OPN_MMN 0.1
 *CROCKTYPE 2
   *WEL_OPN_THR 650.
   *WEL_OPN_MLT 10.
   *WEL_OPN_MMX 5.
   *WEL_OPN_MMN 0.1
 *CTYPE *IJK
        1:10  1:10 1:25 1
       11:40 11:40 1:25 2
 :
*RUN
 *DATE 2000 01 01
 *DTWELL 0.01
 *WELL  1 'CO2-Injector'
   *INJECTOR 'CO2-Injector'
   *INCOMP SOLVENT  1 0
   *OPERATE MAX STG  1.4E+06 CONT
   *OPERATE *MAX *BHP 1.E+05
   *GEOMETRY I 0.2 0.37 1. 0.
   *PERF GEO   'CO2-Injector'
        20 20 5 1. OPEN FLOW-FROM 'SURFACE'
 *WELL 2 'CAP1'
   *PRODUCER 2
   *OPERATE *MIN *BHP 8500.
   *PERF *GEO 2
    15 15 1  1.
*WELL 3 'PROD'
   *PRODUCER 3
   *OPERATE *MAX *STG 1.E+05
   *OPERATE *MIN *BHP 1500.
   *PERF *GEO 3
    10 10 1  1.
 *WELL 4 'CAP2'
   *PRODUCER 4
   *OPERATE *MIN *BHP 8500.
   *PERF *GEO 4
    5 5 1  1.
 *CRL_WELLS ‘CAP*’ ** Wells 2 and 4 have leakage option
```

Thermal Rock Properties (Optional) *CP-ROCK, *THCONR0

PURPOSE:

These keywords are used in the thermal option to define the thermal rock properties.

FORMAT:

*CP-ROCK	cprock
*THCONR0	thconr

DEFINITIONS:

*CP-ROCK

c_R, heat capacity of rock [J/kg·K | Btu/lb·^{o}F].

*THCONR0

τ_c, thermal conductivity of rock and fluid [J/m·s·K | Btu/ ft·hr·^{o}F].

DEFAULTS:

Optional keywords. If these keywords do not appear, and *THERMAL. *ON appears in the Fluid Model section, default values for rock thermal properties will be used [c_R = 885.77 J/kg·K | 0.21 Btu/lb·^{o}F, τ_c = 1.73 J/m·s·K | 1.0 Btu/ ft·hr·^{o}F].

CONDITIONS:

These keywords must be located in the Reservoir Description section. The values entered will be associated with a given compaction/dilation rock type. This number is specified with the *CROCKTYPE keyword which must precede the above keywords.

Heat Loss (Optional) *HEAT-LOSS, *HLPROP

PURPOSE:

These keywords are used in the thermal option to define heat loss to the surrounding rock.

FORMAT:

```
*HEAT-LOSS  (*BOUNDARY)
            (*BOTTOM)
            (*BOTTOM-TOP)
            (*BOTTOM-TOP-BOUNDARY)
            (*REGION)                 i1:i2 j1:j2 k1:k2   (*IDIR)
                                                          (*JDIR)
                                                          (*KDIR)
*HLPROP
rock_den  rock_cp  rock_conductivity
```

DEFINITIONS:

*HEAT-LOSS
: Specification of heat loss to the surrounding (format is similar to aquifer specifications).

*HLPROP
: Specification of properties of surrounding rock

rock_den
: mass density of surrounding rock [kg/m3 or lb/ft3].

rock_cp
: heat capacity of surrounding rock [J/kg·K | Btu/lb·^{o}F].

rock_conductivity
: thermal conductivity of surrounding rock [J/m·s·K | Btu/ ft·hr·^{o}F].

*BOUNDARY
: Heat loss is to take place from all fundamental grid edge blocks (either I equals 1 or ni, or J equals 1 or nj, or both) in the appropriate I or J direction (ni = last block in x direction, nj = last block in y direction).

*BOTTOM
: Heat loss is to take place from the bottom of the reservoir; that is, the bottom of all fundamental grid blocks in the deepest layer.

*BOTTOM-TOP
: Heat loss is to take place from the bottom and top of the reservoir; that is, the bottom of all fundamental grid blocks in the deepest layer and the top of all fundamental grid blocks in the shallowest layer.

*BOTTOM-TOP-BOUNDARY

Heat loss is to take place from the blocks as described for *BOTTOM-TOP and *BOUNDARY.

*REGION

Heat loss is to take place from the region indicated by the following range (i1:i2 j1:j2 k1:k2) and direction keywords (*IDIR, *JDIR or *KDIR) keywords.

i1:i2

Indicates the I beginning and ending indices of the fundamental grid region where the aquifer is to be connected.

j1:j2

Indicates the J beginning and ending indices of the fundamental grid region where the aquifer is to be connected.

k1:k2

Indicates the K beginning and ending indices of the fundamental grid region where the aquifer is to be connected.

*IDIR

The direction of inflow between the aquifer and the reservoir boundary is to be in the I direction.

*JDIR

The direction of inflow between the aquifer and the reservoir boundary is to be in the J direction.

*KDIR

The direction of inflow between the aquifer and the reservoir boundary is to be in the K directions.

DEFAULTS:

Optional keywords. If *HEAT-LOSS is absent, there is no heat loss to the surrounding rock. The defaults given below will be used if *HEAT-LOSS is present but no *HLPROP keyword is specified. If *HLPROP is specified with zeros entered for any of rock_den, rock_cp or rock_conductivity, the following defaults will be used in place of the zero values:

rock_den = 2650.0 kg/m^3 | 165.43 lb/ft^3

rock_cp = 885.77 J/kg·K | 0.21156 Btu/lb·^oF

rock_conductivity = 1.7307 J/m·s·K | 1.00 Btu/ ft·hr·^oF

CONDITIONS:

If *HEAT-LOSS is absent, the keyword *HLPROP cannot appear in the data set. If *HEAT-LOSS is present, one of the sub-keywords identifying the region for heat transfer must be used.

EXPLANATION:

Reference: "A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators", Vinsome, P.K.W. & Westerveld, J.D., JCPT, July-September 1980, Volume 19, No. 3.

Keyword *OUTPRN *GRID *OBHLOSS causes the heat loss rate for each grid block to be dumped to the ".out" file. After that, a summary of heat loss rate and accumulation is written, split between overburden and underburden. Here, "overburden" is the total heat loss for all the blocks with heat loss in the direction opposite gravity; "underburden" is the total heat loss for all the blocks with heat loss in the gravity direction. When heat loss is specified only in those two layers (e.g. using *BOTTOM-TOP) and those layers are distinct, then the overburden and underburden heat loss will sum to the total shown in the material balance summary.

The following example illustrates heat loss to the overburden and underburden, with rock density specified, and rock heat capacity and thermal conductivity defaulted:

```
*HEAT-LOSS  *BOTTOM-TOP
*HLPROP  158.5  0.0  0.0
```

Other Reservoir Properties

Summary of Other Reservoir Properties

This section contains data describing other reservoir properties. These data relate to rock expansion coefficient used in GEM runs coupled with geomechanics.

Critical Keyword Ordering

The critical keyword ordering is:

*END-GRID
Other keywords

It is recommended to follow the order in which keywords appear in this manual, when appropriate.

Indicate End of Grid Definition (Conditional) *END-GRID

PURPOSE:

*END-GRID flags the beginning of the data that defines the other reservoir properties.

FORMAT:

*END-GRID

CONDITIONS:

This keyword must occur after all the grid definition keywords in chapter Reservoir Description and before any subsequent keywords in this chapter, i.e., Other Reservoir Properties. If no keywords like *CTPOR, ..., etc. of this chapter are present in the data file, use of *END-GRID is not necessary.

EXPLANATION:

This keyword signals the Grid Module to stop reading and processing data, and pass the control back to GEM.

Thermal Expansion Coefficient (Optional) *CTPOR, *CPTPOR, *TRPOR

PURPOSE:

*CTPOR signals the input of rock thermal expansion.

*CPTPOR signals the input of pressure-temperature cross-term coefficient.

*TRPOR signals the input of pressure-dependent rock compressibility.

FORMAT:

*CTPOR	*ctpor*
*CPTPOR	*cptpor*
*TRPOR	*trpor*

DEFINITIONS:

ctpor

Effective thermal expansion coefficient of the formation (1/C | 1/F). The lower limit is 0, and the suggested upper limit is 0.01 1/C (0.0056 1/F).

cptpor

Pressure-temperature cross-term coefficient of the formation effective porosity (1/kPa-C | 1/psi-F).

trpor

Reference temperature used in calculation of thermal expansion by geomechanics module (C|F).

DEFAULTS:

If *CTPOR is absent, the formation thermal expansion coefficient is zero.

If *CPTPOR is absent, cptpor = 0 is assumed.

If *TRPOR is absent, the default value is (25C | 60.33F).

CONDITIONS:

This keyword must be in the Other Reservoir Properties keyword group.

Component Properties

Notes on Component Properties

TYPES OF COMPONENTS

Generally, two types of components may be used in GEM:

1. Program Components: Components whose properties are taken from GEM's internal data base of component properties. GEM keeps data for upwards of 60 components commonly encountered in reservoir simulation.
2. User Components: Components whose properties are not taken from GEM's database and whose properties are input by the user. This type of component is normally used when the properties of a component in the fluid do not match those of the program components.

The number of Program components and User components are entered by the keyword *NC. All program components should be ordered before any user components. The properties of a user component are entered via the following keywords:

*SG, *TB, *MW, *AC, *PCRIT, *VCRIT, *TCRIT, *ZCRIT, *PCHOR and *HCFLAG.

All hydrocarbon phase properties are calculated using an equation of state (EOS). The basic properties needed for EOS calculations are the critical properties of each component and the interaction coefficients (Dij) of each component pair.

The critical properties (pc and Tc) are generally not known for User components. They may be estimated internally in GEM by using built-in correlations based on other properties such as Sg, Tb and MW. GEM provides several correlations selected by the *CRIT keyword.

INTERACTION COEFFICIENTS

The Dij's involving the user components must be specified by the user. For hydrocarbon-hydrocarbon component pairs, Dij is calculated from a semi-empirical correlation using critical volumes and an adjustable parameter defined by *PVC3. The form of the semi-empirical correlation is described in the notes for the keyword *PVC3 following.

The Dij's for all other hydrocarbon-non-hydrocarbon component pairs involving user components must be entered by the user using the keyword *BIN.

MULTIPLE EOS REGIONS

The user may specify a number of different EOS parameter sets (see the *EOSSET keyword), and then assign these sets to different portions of the reservoir using the *EOSTYPE keyword. This feature is similar to the stipulation of multiple initialization regions or rock type regions.

A typical application for EOS parameter sets is the modelling of compartmentalized reservoirs with wells completed in more than one region; that is commingled production.

GEM previously supported the use of different EOS sets for reservoir and surface fluid modelling to alleviate the problem of matching fluid behavior over a wide range of conditions. This can also be accomplished with multiple EOS sets, using two sets. The first set would be used for flash calculations at reservoir conditions and the second set for flash calculations at separator and standard conditions. The new multiple EOS set option offers much more scope for separator calculations. The user may now choose between any of the EOS sets available for use in any stage of a (EOS) separator calculation.

The alternate EOS parameter set used previously was defined using the following keywords:

*BINS, *PVC3S, *PCRITS, *OMEGAS, *OMEGBS, *TCRITS and *VSHIFTS

These keywords have been retained to maintain backward compatibility. However, incoming data will now be scanned to ensure that either the earlier sets of keywords for entering one or two EOS sets or the new system for multiple EOS sets is used - mixed formats are not supported. Future use of *BINS, *PVC3S, ... is not recommended.

See the *SEPARATOR keyword in the INITIAL CONDITIONS section and the *WELSEP keyword in the WELL AND RECURRENT DATA section for further discussions of the use of EOS sets defined using the EOSSET keyword.

PHASE IDENTITY

When the EOS calculation determines that the hydrocarbon fluid exists as a single phase, the simulator alone cannot determine whether a phase should be called 'oil' or 'gas' without tedious critical point calculations. Nevertheless, phase identity is important in reservoir simulation, as the oil and gas phases use different relative permeabilities and capillary pressure data. In GEM, the identity of a single phase appearing during the simulation is determined by any one of the following five approaches:

1. An over-riding specification, where the user uses the *PHASEID *OIL or *PHASEID *GAS keyword pairs;
2. Specification of a reference density, where the phase identity is determined by comparison of the phase density with a reference density entered via *REFDEN;
3. Phase classified by comparing the phase molar volume to the mixture pseudo-critical volume, specified using *PHASEID *CRIT;
4. Phase classified by comparing the block temperature to the mixture pseudo-critical temperature, specified using *PHASEID *TCMIX;
5. Phase classified by whether its mass density is closer to a reference gas density or a reference oil density. If necessary, these reference densities are taken from the unstable phases found during stability testing. This method is invoked with *PHASEID *DEN.

The above keywords apply only after the start of the simulation; methods to control single phase identity during initialization of the reservoir are described in the INITIAL CONDITIONS section following.

FLASH METHOD AND DERIVATIVE METHOD

GEM's default formulation uses the analytical calculation of Jacobian entries, and EOS flashes are performed at the Newtonian iteration level. This makes GEM efficient and able to obtain results essentially identical to those that were obtained from earlier techniques. The application of this formulation can be controlled by the keywords *FLASH-METHOD and *DERIVATIVEMETHOD, although the defaults are recommended.

Note that when the keywords *SOLUBILITY or *HENRY-CORR-XXX are used for solubility modelling, *FLASH-METHOD-OGW (equivalent to *OGW-FLASH *NO_H2OVAP) is turned on automatically. This enables a special three phase flash (Oil/Gas/Water) for this situation. If the keywords *HENRYC-*REFPH-*VINFINITY appear without the *SOLUBILITY keyword, *FLASH-METHOD-OGW is not turned on automatically.

Fluid Model (Required) *MODEL

PURPOSE:

*MODEL signals the input of the fluid component model to use for the simulation.

FORMAT:

```
*MODEL      (*PR)
            (*SRK)
```

DEFINITIONS:

*PR

Use the Peng-Robinson equation of state to model the fluid properties of the oil and gas phases.

*SRK

Use the Soave-Redlich-Kwong equation of state to model the fluid properties of the oil and gas phases.

DEFAULTS:

Required keyword. No default.

CONDITIONS:

This keyword must be at the start of the COMPONENT PROPERTY keyword group, following immediately after the RESERVOIR DESCRIPTION keyword group.

Number of Components (Required) *NC

PURPOSE:

*NC specifies the number of primary hydrocarbon components in the fluid.

Note that even though GEM always models an aqueous phase which consists primarily of a water component, this water component is not to be counted here unless *H2O_INCLUDED is specified (see *H2O_INCLUDED).

FORMAT:

*NC nc (nuser)

DEFINITIONS:

nc
: The number of primary hydrocarbon components, excluding the water component (unless *H2O_INCLUDED is specified). The value for nc should include both the program and user components.

nuser
: The number of user components.

DEFAULTS:

Required keyword. No default for nc. If nuser is absent, nuser=0, is assumed.

CONDITIONS:

This keyword must be the second keyword of the COMPONENT PROPERTY keyword group, following immediately after the *MODEL keyword.

EXPLANATION:

There are two types of components:

1. Program Components: Components whose properties are supplied by GEM's component data base. See keyword *COMPNAME for a list of program components.
2. User Components: Components whose properties must be entered by the user.

For example, to simulate a 6 component fluid, using the Peng-Robinson equation of state, enter the following:

```
*MODEL PR
*NC 6
```

as the first two keywords in the component property section. In this case, all 6 components are program components.

Component Name (Required) *COMPNAME

PURPOSE:

*COMPNAME identifies the component of the reservoir fluid.

FORMAT:

*COMPNAME cname

DEFINITIONS:

cname

A total of nc alphanumeric character strings to identify the names of the nc components. One character string is required for each component. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes.

DEFAULTS:

Required keyword. No default.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

There are two types of components:

Program Components:

Components whose properties are supplied by GEM's internal data base.

User Components:

Components whose properties must be input by the user.

When ordering the components, all the program components must appear first followed by all the user components. Intermixing program and user components is not allowed.

The program components are identified by their component names. If the entered cname matches any of the following 60 (reserved) component names, the component will be treated as a program component; otherwise, it will be treated as a user component. Be sure to enter the cname's with the proper spelling.

CH4 (or C1)	FC6	FC26
C2H6 (or C2)	FC7	FC27
C3H8 (or C3)	FC8	FC28
IC4	FC9	FC29
NC4	FC10	FC30
IC5	FC11	FC31

NC5	FC12	FC32
NC6	FC13	FC33
NC7	FC14	FC34
NC8	FC15	FC35
NC9	FC16	FC36
NC10	FC17	FC37
NC16	FC18	FC38
N2	FC19	FC39
CO2	FC20	FC40
H2S	FC21	FC41
H2O	FC22	FC42
TOLUENE	FC23	FC43
BENZENE	FC24	FC44
CYCLO-C6	FC25	FC45

Petroleum fractions FC6 to FC45 are obtained from the data of Katz and Firoozabadi (1978) using Lee-Kesler critical properties correlations (1975). The rest of the component properties are obtained from Reid, Sherwood and Prausnitz (1977).

For example, suppose the reservoir fluid consists of 8 primary components: C1, C2, C3, NC4, NC5, FRACTION_1, FRACTION_2 and FRACTION_3. For this case, enter *COMPNAME as:

```
*NC       8 3
*COMPNAME 'CH4' 'C2H6' 'C3H8' 'NC4' 'NC5'
          'FRAC1' 'FRAC2' 'FRAC3'
```

In this case the first 5 components are program components and the last 3 are user components. Properties need only be entered for the latter 3 components.

In the above example, it is possible to enter the properties of methane directly, as follows:

```
*NC       8 4
*COMPNAME 'C2H6' 'C3H8' 'NC4' 'NC5' 'METHANE'
          'FRAC1' 'FRAC2' 'FRAC3'
```

Since the component name 'METHANE' does not match any of the program component names, it will be treated together with 'FRAC1', 'FRAC2' and 'FRAC3' as user components and thus must appear AFTER the last program component 'NC5'.

Equation of State Set (Optional) *EOSSET

PURPOSE:

*EOSSET signals the input of a new set of EOS component properties.

FORMAT:

*EOSSET set_number

DEFINITIONS:

set_number

The set number to be used in referencing these EOS parameters (critical component properties, interaction coefficients, ...) entered following the *EOSSET keyword.

DEFAULTS:

Optional keyword. See the conditions and explanation sections below for enumeration of valid formats for input of component properties.

CONDITIONS:

This keyword must appear in the COMPONENT PROPERTY keyword group, following the entry of the *MODEL, *NC, *COMPNAME and *HCFLAG keywords which apply to all EOS sets. The viscosity keywords are optional - greater detail on controlling the selection of the viscosity model and parameters on a set by set basis is provided in the EXPLANATION section below.

The EOS sets must be entered sequentially, that is *EOSSET 1 must be followed immediately by all the component property input keywords for Set 1 (such as *PCRIT, *TCRIT, ...), which must precede *EOSSET 2 inputs which likewise must come before *EOSSET 3 inputs ...

Old versions of GEM supported up to two EOS sets though special keywords. The usual keywords *PCRIT, *TCRIT, *VSHIFT, ... had versions *PCRITS, *TCRITS, *VSHIFTS, ... which were used for separator calculations. This system is no longer recommended.

Thus, none of the component property input keywords such as *PCRIT, *TCRIT, ... should be encountered prior to the *EOSSET 1 keyword. Also, the old keywords used for specifying an additional (surface) EOS set under previous versions of GEM (*PCRITS, ...) should NOT appear anywhere in the data.

The following is valid:

```
*EOSSET 1
*PCRIT
*TCRIT
*BIN
...
*EOSSET 2
*PCRIT
*TCRIT
*BIN
...
```

```
*EOSSET 3
*PCRIT
*TCRIT
*BIN
...
```

EXPLANATION:

The following component property input keywords may appear under the *EOSSET keyword:

*PVC3, *MW, *AC, *PCRIT, *VCRIT, *TCRIT, *OMEGA, *OMEGB, *VSHIFT, *PCHOR, *BIN, *VISVC, *VISCOSITY, *VISCOR, *VISCOEFF, *MIXVC, *MIXINGRULE, *VSHIF1, *TREFVS.

<u>Asphaltene (solid) model parameters</u>

*SOLIDMODEL, *SOLIDTHERMAL,*SOLID_TREF

*ADSCST, *WSLMAX, *RFMAX

*SOLID-CONV-RATE

*SOLID_ALPHA, *SOLID_BETA, *SOLID_CRITVEL, *SOLID_GAMMA, *SOLID_SIGMA

*RF_EXPONENT

If a given keyword is missing and it is not required, then defaults will be applied. The selection of the default values differs between the primary (first) set and other sets. For the first set either internally stored values will be applied, or the values will be calculated from internal correlations. Later sets inherit their values from the first set.

For example, if *PVC3 is missing from the first set, then the value 1.0 is assigned. If *PCRIT is missing, the critical pressure will be obtained from a correlations based on the other data specified for the first set (i.e., critical temperatures, acentric factors, etc). Then for sets other than the first set, the defaults are the corresponding values for the first set.

Prior to version 2011.12, the *VISCOR, VISCOEFF, and *MIXVC keywords were required to be placed before any *EOSSET keywords. Now, each EOS region is allowed to have completely independent viscosity-related input. This means that, if the *EOSSET keyword appears in the data set, no viscosity input keywords may appear before the *EOSSET keyword.

The same viscosity model (specified with *VISCOR) may be used for all EOS sets, with differing values of correlation coefficients and component viscosity data, or different viscosity model types may be specified for each EOS set.

For example, it is now possible to select *VISCOR *HZYT for EOS Set 1 and *VISCOR *LE for Set 2. The critical molar volume for the viscosity calculation, which is a parameter in the *HZYT and *LE models, may be stipulated differently for each set as before.

The system for defaulting viscosity parameters is now the same as for other EOS data: the first EOS set uses internally stored defaults or calculated values for any data that is not specified, and all later sets inherit their values from the first set, provided the viscosity model specified with *VISCOR (or defaulted) is the same as the first set. If a viscosity model specified for a later set is different than that for the first set, the internal defaults for that model will be applied in the later set.

The exception to the defaulting rule is that VCVIS will always default to VCRIT for the current EOS set.

Example 1: Consider the following viscosity model specification for multiple EOS sets as used in template gmsmo046.

```
*EOSSET 1
** Modified Pedersen viscosity model
** (For 21 C)
*VISCOR *MODPEDERSEN
*VISCOEFF   1.0000000E-04  2.4695000E+00  8.8536000E-03  2.2164000E+00
            3.7152000E-01
*EOSSET 2
** Mixing rule viscosity model
** (For 21 C)
*VISCOR *MIX
*VISCOSITY   0.0112  0.076  1.072  7.019  103168.0  103168.0
*MIXINGRULE  0.21
*EOSSET 3
** Original Pedersen viscosity model
** (For 21 C, tuned for solvent addition)
*VISCOR *PEDERSEN
*VISCOEFF   2.9100000E-01  1.0000000E+00  7.6521528E-05  6.1180000E+00
            5.4625435E-01
*EOSSET 4
** JST viscosity model - HZYT mixing rule for low P viscosity
** (For 41 C)
*VISCOR *HZYT
*MIXVC      1.0000000E+00
*VISVC      9.9000000E-02  2.0300000E-01  1.1295196E+00  1.6489186E+00
            4.0361675E+00  4.0361675E+00
*VISCOEFF   1.0230000E-01  2.3364000E-02  5.8533000E-02 -4.0758000E-02
            9.3324000E-03
*EOSSET 5
** JST viscosity model - LE model for low P viscosity
** (For 41 C)
*VISCOR *LE
*MIXVC      1.0000000E+00
*VISVC      9.9000000E-02  2.0300000E-01  1.1295196E+00  1.6489186E+00
            4.0361675E+00  4.0361675E+00
*VISCOEFF   1.0230000E-01  2.3364000E-02  5.8533000E-02 -4.0758000E-02
            9.3324000E-03
*EOSTYPE *KVAR 1 2 3 4 5
```

Multiple EOS set convention related to solid (asphaltene) parameters specification

Keywords related to modeling of solid (asphaltene) precipitation, flocculation, deposition and adsorption may be specified for each equation of state set (EOSSET). For set number 1, if the user does not explicitly specify a parameter value with an appropriate keyword, then the internal default value for that parameter will be assigned to set number 1. For all other sets if a value is not specified for that set explicitly via a keyword, then the value assigned to set number 1 is inherited. However, it is possible to switch OFF inheritance for all sets by specifying the keyword *INHERIT-SOLID-OFF. If inheritance is turned off, then for set number 2 and subsequent sets, the internal default values for all parameters will apply unless the user has explicitly specified a value for that parameter via a keyword.

Example 1: Consider the following asphaltene model specification for multiple EOS sets as used in template gmspr028dat.

```
*MODEL   *PR
*NC    7    7
*COMPNAME
'N2+C1 ' 'C2-C5   ' 'C6-C9   ' 'C10-C15 ' 'C21-C26 '
'C32-C34 ' 'ASPHALT '
*TRES  92.0
*EOSSET 1
*PCRIT ...
*SOLIDMODEL
-11.060 0.9 60000.0 0.9
*RFMAX 10.0
*WSLMAX 1.0E-04
*ADSCST 0.012
*EOSSET 2
*PCRIT ...
*EOSTYPE *KVAR 1 1 1 1 2 2 2 2
```

In this case set number 2 inherits parameter values specified with *SOLIDMODEL,*RFMAX, *WSLMAX and *ADSCST for EOSSET number 1 since these keywords are missing in set number 2.

In example two below the presence of INHERIT-SOLID-OFF keyword prevents inheritance of parameter values from set number 1 for set number 2. Therefore for layers 5-8 assigned to set number 2 with the *EOSTYPE keyword, will pick up the internal default parameter values. The defaults are chosen so that there is no precipitation possible, hence for grid blocks corresponding to layers 5-8 there will be no in situ precipitation, however precipitated asphaltene from neighboring blocks can still appear in these grid blocks by flow or entrainment.

Example 2: Inheritance does not apply to set number 2. Grid blocks assigned to set number 2 do not allow precipitation.

```
*MODEL   *PR
*NC    7    7
*COMPNAME
'N2+C1 ' 'C2-C5   ' 'C6-C9   ' 'C10-C15 ' 'C21-C26 '
'C32-C34 ' 'ASPHALT '
*TRES  92.0
*INHERIT-SOLID-OFF
*EOSSET 1
*PCRIT ...
*SOLIDMODEL
-11.060 0.9 60000.0 0.9
*RFMAX 10.0
*WSLMAX 1.0E-04
*ADSCST 0.012
*EOSSET 2
*PCRIT ...
*EOSTYPE *KVAR 1 1 1 1 2 2 2 2
```

Templates gmspr028 through gmspr031 provide examples of using multi-EOS asphaltene (solid) model feature.

Equation of State Model Type (Optional) *EOSTYPE

PURPOSE:

*EOSTYPE indicates the start of input of equation of state set assignments. An equation of state set number is assigned to each grid block (see *EOSSET above).

ARRAY:

*EOSTYPE set_number_array

DEFAULTS:

Optional keyword. The default is *EOSTYPE *CON 1, which uses the first EOS set encountered for all grid blocks.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group. It should appear after all the EOS set data has been entered. In particular, all EOS sets must have been fully defined.

EXPLANATION:

*EOSTYPE indicates the input of an array of grid values. All array reading option qualifiers are valid. See the section on ARRAY READING OPTIONS for more details.

Example: Assuming that a GEM simulation using a ni = nj = 20 and nk = 2 grid is being done, and layer 1 is a different, and separated reservoir, from layer 2, then the following might be used:

*EOSTYPE *KVAR 1 2

This data assigns EOS set 1 to layer 1 and EOS set 2 to layer 2. A third EOS set might be used for the initialization and well separators. In this case, *EOSSET 1, *EOSSET 2 and *EOSSET 3 data would have appeared previously in the data.

User Component Properties (Conditional) *SG, *TB, *MW, *AC, *PCRIT, *VCRIT, *TCRIT, *ZCRIT, *PCHOR, *HCFLAG

PURPOSE:

These keywords are used to specify the properties of the user components.

*SG identifies specific gravity (dimensionless)

*TB identifies the average normal boiling point (deg C | deg F)

*MW identifies the molecular weight (gm/gmol)

*AC identifies the acentric factor (dimensionless)

*PCRIT identifies the critical pressure (atm)

*VCRIT identifies the critical volume (m3/k-mol)

*TCRIT identifies the critical temperature (deg K)

*ZCRIT identifies the critical compressibility factor (dimensionless)

*PCHOR identifies the parachor (dimensionless)

*HCFLAG identifies whether the user component is a hydrocarbon component (see below)

FORMAT:

*SG	sg
*TB	tb
*MW	amolw
*AC	ac
*PCRIT	pc
*VCRIT	vc
*TCRIT	tc
*ZCRIT	zc
*PCHOR	parachor
*HCFLAG	hcflag

DEFINITIONS:

The designations sg, tb, amolw, ac, pc, vc, tc, zc, parachor and hcflag are lists that consist of nuser numbers, where nuser is the total number of user components.

See *NC and *COMPNAME for an explanation of user components.

Note that hcflag values only affect how binary interaction coefficients are chosen and are set as follows:

1: the user component is a hydrocarbon component

0: the user component is a non-hydrocarbon component.

Further details are given below.

DEFAULTS:

Conditional keywords. See explanation below for defaults.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

These keywords are used to enter the properties of the user components. Enter zero for those that are not known. If any of these properties is unknown for ALL user components, do not enter the keyword. Not all properties are required.

A vc value is required for viscosity calculations if the Jossi-Stiel-Thodos correlation is used. A zc value can be entered in place of vc, as vc can be calculated directly from zc. If both *ZCRIT and *VCRIT are missing, zc will be calculated from the acentric factor.

Equation of state fluid properties calculations need tc, pc, ac and amolw values for all components. If any of these properties are not known, as indicated by either zero values or missing keywords, they will be calculated internally from empirical correlations using the other available information. Use the *CRIT keyword to select the desired empirical correlation.

The sg and tb values are used only to calculate the required critical properties and amolw. If all the required tc, pc and amolw values are known, then the *SG and *TB keywords are not required. If either sg or tb are required and are not known, they will be calculated internally using empirical correlations relating sg, tb and amolw. Thus, the minimum required data for the sg, tb, pc, tc, ac and amolw group is any one of sg, tb or amolw.

In general, the more information that is supplied about the fluid components, the more accurate will be the predicted phase properties.

The parachor is used to calculate interfacial tension. If it is not known, parachor values will be calculated from the molecular weights of the components.

The hcflag values indicate whether a user component should be classified as a hydrocarbon component. The classification of a component as a hydrocarbon or non-hydrocarbon simply directs the simulator to obtain certain interaction coefficients either from user input values or from a formula. If the user wants GEM to use special values for the interaction coefficients which may have been fitted to available data, then the components between which these interaction coefficients are to apply MUST have been flagged as non-hydrocarbons. Otherwise interaction coefficient values computed from the correlation will overwrite them.

Binary interaction coefficients between pairs of hydrocarbon components are calculated using a built-in correlation (see *PVC3); otherwise, (that is, for hydrocarbon-non-hydrocarbon pairs and non-hydrocarbon-non-hydrocarbon pairs), the interaction coefficient values will be taken from data entered using the *BIN keyword.

For example, consider an 8 component fluid with 3 user components. Assume no interfacial tension calculation is required, so that the *PARACH keyword need not be entered. The following are all valid (and complete) specifications of the user components:

```
** Molecular weight is not known for the first 2
** user components, so they will be calculated
** from the available sg.
** No boiling temp. information is available, so
** they will be calculated from the sg and tb.
** All pc, tc and ac values are then calculated
** from sg, tb and mw.
*SG   0.701   0.741    0.771
*MW   0.0     0.0    211.5      ** gm/gm-mole
```

-or-

```
** pc, tc and ac calculated from sg, tb and mw.
*SG  0.701  0.741  0.771
*TB  113.1  184.7  262.7    ** degree F
*MW  115.1  156.6  211.5    ** gm/gm-mole
```

-or-

```
** Specify pc, tc and ac for the first 2 user
** comp.
** Calculate pc, tc and ac from sg, tb and mw for
** the last user comp.
*SG       0.0    0.0  0.771
*TB       0.0    0.0  262.7    ** degree F
*MW       0.0    0.0  211.5    ** gm/gm-mole
*AC     0.394  0.532    0.0
*PCRIT  24.50  19.80    0.0    ** atm
*TCRIT  569.0  630.0    0.0    ** deg K
```

Equation of State Omega Parameters (Optional)

***OMEGA, *OMEGB**

PURPOSE:

These keywords are used to specify the equation of state omega-A and omega-B parameters.

*OMEGA identifies directly the equation of state omega-A parameter (dimensionless)

*OMEGB identifies directly the equation of state omega-B parameter (dimensionless)

FORMAT:

```
*OMEGA  omega
*OMEGB  omegb
```

DEFINITIONS:

omega

A set of nc numbers representing omega-A values for all nc components of the fluid.

omegb

A set of nc numbers representing omega-B values for all nc components of the fluid.

DEFAULTS:

Optional keywords. Defaults for the Peng-Robinson Equation of state (EOS) are:

omega = 4.57235530d-01
omegb = 7.77960740d-02

Default for the SRK EOS are:

omega = 4.2748020d-01
omegb = 8.6640350d-02

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

The omega and omegb's are used in the EOS for reservoir fluid properties calculations.

User Component Interaction Coefficients (Conditional) *BIN

PURPOSE:

This keyword specifies the interaction coefficient parameters for the user components.

FORMAT:

*BIN delij

DEFINITIONS:

delij: A total of nuser * (2 * nc - nuser - 1) / 2 real numbers representing interaction coefficients. Note that nc and nuser are the total number of components and total number of user components respectively (see the *NC keyword).

DEFAULTS:

Conditional keywords. Default is zero.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords. It is required only when there are user components in the fluid system (nonzero nuser).

EXPLANATION:

The delij are the binary interaction coefficients between the user components and all other components.

Entered delij values concerning interactions between user components and hydrocarbons may be overwritten if the user component is specified as a hydrocarbon. See the explanations in the keyword *PVC3 for more information.

For example, consider an 8-component fluid with 3 user components, numbered component 6 through 8. Enter the *BIN keyword as follows:

```
*BIN D61 D62 D63 D64 D65
     D71 D72 D73 D74 D75 D76
     D81 D82 D83 D84 D85 D86 D87
```

In the above, Dij is a number representing the interaction coefficients between component i and component j.

Consider a 4-component fluid made up entirely of user components. *BIN data may be entered as follows:

```
*BIN D21
     D31 D32
     D41 D42 D43
```

Interaction coefficients between a component and itself are always assumed to be zero, i.e. Dii = 0.0. Dij's are assumed to be symmetric, i.e. Dij = Dji.

Parameters for Hydrocarbon Interaction Coefficients (Optional) *PVC3

PURPOSE:

*PVC3 specifies the parameter used to calculate the interaction coefficients between hydrocarbon-hydrocarbon component pairs (components both having 1's set using *HCFLAG)..

FORMAT:

*PVC3 pvc3

DEFINITIONS:

pvc3

A real number used to generate binary interaction coefficients, delij, between hydrocarbon components. The following is used:

$$\text{delij} = 1 - \left| \frac{2 * \sqrt{\text{vc3}(i) * \text{vc3}(j)}}{\text{vc3}(i) + \text{vc3}(j)} \right|^{\text{PVC3}}$$

where:

vc3(i) = (critical volume of component i) ** (1/3)

Note that setting PVC3 = 0 signifies zero interaction coefficients between hydrocarbons.

A typical value for PVC3 is 1.2.

DEFAULTS:

Optional keyword. Default is zero.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

The above equation for delij will be applied to all hydrocarbon-hydrocarbon component pairs including both program and user components; that is, between components that have had their hcflag's set to 1's using *HCFLAG.

Thus, delij values for user components entered using *BIN may be overwritten. To stop such overwriting, flag the user components to be to be non-hydrocarbons by setting their hcflag's in *HCFLAG to be 0.

Critical Property Correlations for User Components (Conditional) *CRIT

PURPOSE:

*CRIT identifies the correlation to be used to calculate the critical properties of user components that have not been entered.

FORMAT:

*CRIT (*LEEKESLER)
(*RIAZI)
(*TWU)

DEFINITIONS:

*LEEKESLER
The Lee-Kesler correlation (Lee and Kesler, 1975) will be used.

*RIAZI
The Riazi-Daubert correlation (Riazi and Daubert, 1980) will be used.

*TWU
The Twu correlation (Twu, 1984) will be used.

DEFAULTS:

Conditional keyword. Default is *LEEKESLER

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

This keyword applies only when user components are present and some of their critical properties have not been specified. In this case, they are calculated from specific gravity, normal boiling point and molecular weight data using the selected correlation. See *PCRIT and *TCRIT for additional information.

Example:

```
*CRIT  *TWU
```

Volume Shift Parameters (Optional) *VSHIFT, *VSHIF1, *TREFVS

PURPOSE:

*VSHIFT specifies the volume shifts to be applied to the equation of state molar volumes. *VSHIF1 and *TREFVS indicate that a linear temperature-dependent volume shift model will be applied.

FORMAT:

```
*VSHIFT   shv
*VSHIF1   shv1
*TREFVS   trefvs
```

DEFINITIONS:

shv

A series of nc real numbers corresponding to the dimensionless volume corrections for the nc components in the fluid. If the temperature-dependent volume shift model is used, shv correspond to corrections at the reference temperature (trefvs) of each component.

shv1

A series of nc real numbers corresponding to the gradients of the linear temperature-dependent volume corrections for the nc components in the fluid. (deg C^{-1} | deg F^{-1})

trefvs

A series of nc real numbers corresponding to the reference temperature in temperature-dependent volume corrections calculation for the nc components in the fluid. (deg C | deg F)

DEFAULTS:

Optional keywords. The default is zero for volume shifts shv, and also zero for gradients of volume shift shv1. If *VSHIF1 is entered but *TREFVS is not, the default reference temperature is 60 F | 15.56 C.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

The volume translation method of Peneloux et. al. (1982) is available in GEM to improve the density predictions. In this method the computed phase compressibility (Z factor) and molar volume, v are modified as follows:

$$Z = Zo - P(v\sim) / (RT)$$
$$v = vo - v\sim$$

with

$$v\sim = \sum\left[x(i) * Ti(i)\right]$$
$$T(i) = \left[shv(i) + shv1(i) * (tres - trefvs(i))\right] * bc(i)$$
$$bc(i) = omegb(i) * R * Tc(i) / Pc(i)$$

where:

Σ	= summation over all nc components;
Zo	= compressibility factor calculated from the EOS;
vo	= molar volume calculated from the EOS, m3/kmol;
nc	= the total number of components;
x(i)	= the mole fraction of component i in the fluid; and
tres	= the reservoir temperature
omegb(i)	= the EOS omega-B parameter.

For the PR EOS, omega_B = 0.07780;

For the SRK EOS, omega_B = 0.08664.

Typically shv < 0 for components lighter than C6, and shv > 0 for components heavier than C6, for the PR EOS.

Example: Consider an 8-components fluid, with volume shift applied only to methane and the 3 heaviest fractions. Enter:

```
*NC        8 3
*COMPNAME 'CH4'  'C2H6'  'C3H8'  'NC4'  'NC5'
          'FRAC1'  'FRAC2'  'FRAC3'
*VSHIFT   -0.0001 0.0 0.0 0.0 0.0
             0.001 0.001 0.001
```

Example: Consider a 12-components fluid, with temperature dependent volume shift applied and reference temperature of 100F for each component. Enter:

```
*UNIT     *FIELD

*NC       12 12
*COMPNAME 'CO2'  'N2-C1'  'C2'  'C3'  'IC4-NC4'
          'IC5-NC5'  'FC6'  'CH3-FC8'  'ETH-FC1'  'FC1-FC2'
          'FC2-C29'  'C30+'
*VSHIFT   0.0 -0.13093 0.00706 -0.03954 –0.05833
          -0.04588 -0.05945 -0.00619 0.08993 0.15586
          0.17142 0.33946
*VSHIF1   0.000441 -0.000016 0.000518 0.000463 0.000139
          0.000012 -0.000056 -0.000072 -0.000075 -0.000099
          -0.000119 -0.000037
*TREFVS   100.0 100.0 100.0 100.0 100.0
          100.0 100.0 100.0 100.0 100.0
          100.0 100.0
```

Heating Values (Optional) *HEATING_VALUES

PURPOSE:

*HEATING_ VALUES specifies the heating values (heat of combustion) for each component. The number of values and order should follow according to the component list entered with the *COMPNAME keyword. These numbers are used by GEM to calculate the separator gas stream heating value by simple mole fraction averaging.

FORMAT:

*HEATING_VALUES hval

DEFINITIONS:

Hval

A series of nc real numbers corresponding to the heating value (heat of combustion) for the nc components in the fluid. For field units enter values in Btu/gmol and for SI units in kcal/gmol.

DEFAULTS:

Optional keyword. Default is zero.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

The heating value of a stream or component is the heat of combustion assuming the reaction goes to completion i.e. the reaction takes place with excess oxygen and the final products are carbon dioxide and water.

GEM will calculate and report the heating value of all the well streams in the output file using the known composition of the stream by mole fraction averaging the entered component heating values supplied by the user.

Note : Selection of *OUTPRN *WELL *BRIEF in the I/O section will however suppress the heating value printing.

The heating value of the separator gas stream is also available for plotting by the RESULTS program by individual wells and by groups for production wells. This is done automatically if the *HEATING_VALUES keyword is present in the component properties section.

$$hval \sim = \Sigma \left[x(i) * Hval(i) \right]$$

where:

Σ	= summation over all nc components;
val	= heating value of the separator gas stream for a given well, iw.
Nc	= the total number of components;
x(i)	= the mole fraction of component i in the separator gas; and
Hval(i)	= the heating value entered for each component.

Approximate heating values for the Library components such as methane, ethane etc have been added to the list of properties in WinProp. Following the recommended procedure for creating the fluid description (EOS) model for GEM, WinProp will automatically add the *HEATING_VALUES keyword and associated values to the .gem output file. The table below shows the library components and the heating values stored in WinProp. For the carbon fraction FC7-FC45 the heating value equals 1002.57 kcal/gmol + 157.44 kcal/gmol increment for every carbon number greater than 6. Same values are used for NC6 and FC6 and NC7 and FC7 etc. For pseudo components values of zero will be assigned as at present there is no method implemented for estimating these values.

Component Name	Heating Value (Btu/gmol)
H2S	0.0
N2	0.0
CO2	0.0
CH4 (or C1)	844.29
C2H6 (or C2)	1478.46
C3H8 (or C3)	2105.16
IC4	2711.54
NC4	2711.54
IC5	3353.66
NC5	3353.66
FC6	3975.91
FC7	4600.28
FC8	5224.64
FC9	5849.00
FC10	6473.36
FC11	7097.73
FC12	7722.09
FC13	8346.45
FC14	8970.82
FC15	9595.18
FC16	10219.54
FC17	10843.91
FC18	11468.22
FC19	12092.63
FC20	12717.00
FC21	13341.36
FC22	13965.72
FC23	14590.08
FC24	15214.45
FC25	15838.81
FC26	16463.17
FC27	17087.54
FC28	17711.90

FC29	18336.26
FC30	18960.63
FC31	19584.99
FC32	20209.35
FC33	20833.71
FC34	21458.08
FC35	22082.44
FC36	22706.80
FC37	23331.17
FC38	23955.53
FC39	24579.89
FC40	25204.26
FC41	25828.62
FC42	26452.98
FC43	27077.35
FC44	27701.71
FC45	28326.07
NC6	3975.91
NC7	4600.28
NC8	5224.64
NC9	5849.00
NC10	6473.36
NC16	10219.54
TOLUENE	3705.97
BENZENE	3097.15
CYCLO-C6	3715.32
H2O	0.0

Example: Consider an 8-components fluid, with the last three components being pseudo components. In Field units, the heating values as written out by WinProp would result in the following lines appearing in the .gem output file:

```
*NC        8 3
*COMPNAME 'CH4'  'C2H6'  'C3H8'  'NC4'  'NC5'
          'FRAC1'  'FRAC2'  'FRAC3'
          844.29 1478.46 2105.16 2711.54 3353.66 0.0 0.0 0.0
```

Equation of State Parameters at Surface Conditions (Optional)

***BINS, *PVC3S, *PCRITS, *OMEGAS, *OMEGBS, *TCRITS, *VSHIFTS**

PURPOSE:

These keywords specify equation of state (EOS) parameters for fluid properties calculations at separator and surface conditions.

Use of these keywords is no longer recommended. Use EOSSET instead.

*BINS specifies the interaction coefficient parameters for the user components in the second set.

*PVC3S specifies the parameter used to calculate the interaction coefficients between hydrocarbon-hydrocarbon component pairs in the second set.

*PCRITS specifies critical pressure [atm] in the second set.

*OMEGAS identifies the EOS omega-A parameters in the second set.

*OMEGBS identifies the EOS omega-B parameters in the second set.

*TCRITS specifies the critical temperature [deg K] in the second set.

*VSHIFTS specifies the volume shifts in the second set.

FORMAT:

*BINS	delijs
*PVC3S	pvc3s
*PCRITS	pcs
*OMEGAS	omegas
*OMEGBS	omegbs
*TCRITS	tcs
*VSHIFTS	shvs

DEFINITIONS:

delijs

A total of nuser * (2 * nc - nuser - 1) / 2 real numbers representing the interaction coefficients. Note that nc and nuser are the total number of components and total number of user components respectively.

pcs

Nc numbers corresponding to the critical pressure of the nc components of the fluid [atm].

pvc3s

A number used to define binary interaction coefficients between hydrocarbon components.

omegas

Nc numbers representing the omega-A values of the nc components of the fluid.

omegbs

Nc numbers representing the omega-B values of the nc components of the fluid.

tcs

Nc numbers corresponding to the critical temperatures of the nc components of the fluid [deg K].

shvs

Nc numbers corresponding to the dimensionless volume corrections of the nc components in the fluid.

DEFAULTS:

Optional keyword. Defaults are the corresponding values used for calculations at reservoir conditions:

Default for delijs is delij specified in *BIN;

Default for pcs is pc specified in *PCRIT;

Default for pvc3s is pvc3 specified in *PVC3;

Default for omegas is omega specified in *OMEGA;

Default for omegbs is omegb specified in *OMEGB;

Default for tcs is tc specified in *TCRIT;

Default for shvs is shv specified in *SHV.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

Older versions of GEM allowed the specification of two sets of EOS parameters through the use of these keywords. The *EOSSET keyword is now recommended. For compatibility reasons, the older method will be described here.

The usual EOS set (entered without the *EOSSET keyword) is used for reservoir calculations:

*BIN, *PVC3, *PCRIT, *OMEGA, *OMEGB, *TCRIT and *VSHIFT.

Another EOS set for separator and surface condition calculations can be entered using:

*BINS, *PVC3S, *PCRITS, *OMEGAS, *OMEGBS, *TCRITS and *VSHIFTS.

Refer to the descriptions of the individual keywords for a description of their actions.

NOTE:

CMG's phase behavior package WINPROP also supports two sets of EOS parameter input. Please refer to the WINPROP Users' Manual for more information.

The concept of specifying a different EOS model for reservoir and surface conditions has been generalized to give the user the ability to input any number of EOS sets and the ability to assign particular EOS sets to selected regions of the reservoir, and to separator stages. See the description of the *EOSSET keyword above and the *SEPARATOR keyword in the INITIAL CONDITIONS section and the *WELSEP keyword in the WELL AND RECURRENT DATA section for further information.

Reservoir Temperature (Required) *TRES

PURPOSE:

*TRES specifies the reservoir temperature

FORMAT:

```
*TRES     tres
-or
*TRES     depth1     tres1   depth2   tres2
```

DEFINITIONS:

tres
: A positive real number representing the reservoir temperature (deg C | deg F)

depth1
: A real number representing a depth at which a temperature is known; this depth is used in specifying a temperature gradient within the reservoir (m | ft).

tres1
: A positive real number representing the temperature at depth depth1 (deg C | deg F).

depth2
: A real number representing a second depth at which a temperature is known; this depth is used in specifying a temperature gradient within the reservoir (m | ft).

tres2
: A positive real number representing the temperature at depth depth2 (deg C | deg F).

DEFAULTS:

Required keyword. No default.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords. *TRES must be followed by either a single real number or four real numbers. If only a single number follows *TRES, then the temperature at all positions in the reservoir is set equal to this value. If four numbers follow *TRES, then the temperature at a reservoir grid block depends upon the depth of the grid block centre as follows:

T(block) = tres1+[(tres2-tres1)/(depth2-depth1)]* (depth(block) - depth1)

making temperature a function of depth (only).

When temperature varies in the reservoir, GEM takes account of the variation only in EOS flashes and in the computation of hydrocarbon properties. Water properties are assumed to be only pressure-dependent in the reservoir and, when a requirement for temperature dependence arises, this dependence is computed using the temperature at the mid-depth of the reservoir.

When the mixing rule with user-supplied amu parameters (see the entry under *VISCOSITY in this manual) is used to compute the hydrocarbon viscosities, no temperature dependence is included in the hydrocarbon viscosities.

When the Herning-Zipperer-Yoon-Thodos viscosity correlation is used (see *VISCOR following), the temperature dependence is computed at the beginning of the simulation (using the temperature at mid-depth in the reservoir) and it is also updated to take account of variations from grid block to grid block at subsequent times if a non-zero temperature gradient has been entered for the reservoir, or if the thermal option is invoked.

Similarly, if the Lee-Eakin viscosity correlation (invoked with *VISCOR *LE) is used for the hydrocarbon viscosities, the temperature variations from grid block to grid block are included in the viscosity computation..

EXPLANATION:

Examples:

```
*TRES  39.0
Reservoir temperature is 39 degrees everywhere.
*TRES  1300.  160.  1500.  170.
Temperature increases by 5 degrees for 100 units increase
in depth.
```

Identity of Single Phase Fluid (Optional) *PHASEID, *REFDEN

PURPOSE:

*PHASEID specifies one of several methods for identifying single phase reservoir fluids as oils or gases when they are found in grid blocks during a simulation.

*REFDEN specifies a reference mass density for the determination of the identity (oil or gas) of single phase reservoir fluids found during a simulation.

FORMAT:

```
*PHASEID    (*OIL)
            (*GAS)
            (*CRIT)
            (*DEN)
            (*TCMIX)
-or-
*REFDEN     dvref
```

DEFINITIONS:

*OIL

Subkeyword indicating that all single phase blocks in the reservoir will be identified as oil phase blocks during simulation.

*GAS

Subkeyword indicating that all single phase blocks in the reservoir will be identified as gas phase blocks during simulation.

*CRIT

Subkeyword indicating that single phase identities will be determined according to fluid properties using the following criteria (Gosset et.al.,1986):

a) At supercritical conditions, named gas;
b) At subcritical conditions, named oil when its molar volume < critical molar volume. To avoid expensive critical point calculations, the EOS critical properties for the mixture computed from mixing rules are assumed to be the true critical values.

*DEN

A single phase fluid is classified according to whether its mass density is nearer to that of a reference gas phase or that of a reference liquid phase. These reference densities are computed automatically, using unstable phases from stability tests where necessary, and need not be entered by the user.

*TCMIX

Subkeyword indicating that single phase identities will be determined by comparing the block temperature to the pseudo-critical temperature of the mixture calculated using the Li critical temperature mixing rule (Li, 1971):

a) If block temperature is greater than or equal to the pseudo-critical temperature, the phase is identified as a gas;

b) If block temperature is less than the pseudo-critical temperature, the phase is identified as an oil.

dvref

A reference mass density (kg/m3 | lb/ft3) to be used when *REFDEN is specified.

DEFAULTS:

Optional keyword. Default for *PHASEID is *DEN. No default for *REFDEN. *REFDEN overrides *PHASEID if both are entered.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

Do not enter *PHASEID together with *REFDEN.

EXPLANATION:

GEM can use one of five methods to specify the identity of single phase fluids in appearing in grid blocks during a simulation. Such identification is important for determining the correct rock-fluid properties to be used for flow calculations. The five methods are:

1. Direct specifications via *PHASEID *OIL or *PHASEID *GAS.
2. A determination made using the method of Gosset et al. in which approximate critical properties of the mixture are used to determine the phase identity.
3. A determination made by comparing the block temperature to the pseudo-critical temperature of the phase. The Li mixing rule for pseudo-critical temperature is given by:

   $$T_{c,mix} = \frac{\sum_i z_i v_{c,i} T_{c,i}}{\sum_i z_i v_{c,i}}$$

   where z_i is the global mole fraction of component i, $T_{c,i}$ and $v_{c,i}$ are the critical temperature and volume of component i for the EOS set in use for a given block.
4. A mass density-based method in which the internally-computed mass densities of a reference gas phase and a reference oil phase are used for phase identification.
5. A determination made by comparing the single phase fluid's mass density with a reference mass density entered via the *REFDEN keyword. The fluid is assumed to be gas when its mass density < dvref-delta, and assumed to be an oil when its mass density > dvref+delta, where delta = 3.12 lb/ft3 (50.0 kg/m3). When dvref-delta < mass density < dvref+delta, the phase is identified as a gas if the mole fraction of gas phase hydrocarbons in the block exceeded 0.5 at the end of the preceding timestep (or at the end of initialization, for the first timestep). During initialization the mole fraction of hydrocarbons in the gas phase is taken as zero in the oil and water zones (unless the reservoir has been specified as a gas-water

reservoir by setting the water-oil and gas-oil contact depths equal) and in the gas cap is determined from a flash of the given hydrocarbon composition.

When both *PHASEID and *REFDEN are present, *REFDEN is used.

Note that for two phase systems, the phase with the lower mass density is always assumed to be gas and the other oil. GEM is currently capable of representing both an oil and gas phase, and a solid phase under some conditions (see *SOLIDMODEL). Other co-existing hydrocarbon phases cannot be separately represented in GEM.

Example:

```
*PHASEID *CRIT
```

This would invoke application of the Gosset et al. criterion.

Average Saturation Pressure (Optional) *PSAT

PURPOSE:

*PSAT specifies a two-phase saturation pressure that can be used to save flash calculation work in some situations.

FORMAT:

*PSAT psat

DEFINITIONS:

psat

Saturation pressure of the original reservoir fluid (kPa | psia).

DEFAULTS:

Optional keyword. Default is 1.0e+20 (kPa | psia).

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

Specification of *PSAT is done to reduce simulator overhead in certain situations.

The fluid in a grid block is assumed to be in a single phase state whenever the block pressure is greater than 'abs(psat)'. This can reduces the number of flash calculations required during the simulation. However, to avoid inconsistencies, psat MUST be greater than or equal to the saturation pressure calculated by the EOS used in the simulator. This can be verified using CMG's Phase Property Package, WINPROP.

If psat is entered as a positive value, the flash calculation is skipped provided that the grid block compositions remain identical to the original reservoir fluid compositions. To skip flash calculations whenever the block pressure>abs(psat), irrespective of the block's hydrocarbon composition, enter psat as the negative of the desired value.

Thus, if PSAT = -1.0, the entire reservoir will always remain a single phase. This is useful for simulation of first contact miscible processes.

Even though use of this keyword can reduce simulation run times, great care must be taken in its use to avoid missing essential physical phenomena in the simulation.

Viscosity Correlation Specification (Optional) *VISCOR

PURPOSE:

Keyword to specify how the hydrocarbon phase viscosities are to be computed.

FORMAT:

```
*VISCOR     (*MIX)
            (*HZYT)
            (*LE)
            (*PEDERSEN)
            (*MODPEDERSEN)
```

DEFINITIONS:

*MIX

Specifies that a mixing rule will be used directly to compute hydrocarbon phase viscosities at reservoir conditions from component viscosities at reservoir conditions entered by the user. For the explicit form of the viscosity mixing rule, see the explanation under the *VISCOSITY keyword.

*HZYT

Indicates that a mixing rule due to Herning and Zipperer (Gas und Wasserfach, v. 79, 49 (1936)):

$$\mu(\text{mix}) = \frac{\sum_{i=1}^{nc} \mu(\text{low},i)x(i)\sqrt{M(i)}}{\sum_{i=1}^{nc} x(i)\sqrt{M(i)}}$$

is used to compute a low pressure viscosity for the hydrocarbon mixture, with component low pressure viscosities computed using a formula developed by Yoon and Thodos (AIChE J, v.16, 300-304 (1970)):

$$\mu(\text{low},i) = (4.610 * \text{Tr} ** 0.618 - 2.040 * \exp(-0.449 * \text{Tr}) + 1.94 * \exp(-4.058 * \text{Tr}) + 0.1 * 10 ** (-4) / \mu p,$$

where Tr = Tabs/Tc is the reduced temperature of the component and μp is the viscosity parameter μp = (Tc/(M**3*Pc**4))**(1/6). Here Tc is in K, Pc in atm, M in g/gmol and μ(low,i) is given in centipoise by this formula.

The mixture viscosity at reservoir conditions is then computed from the low pressure viscosity using a formula due to Jossi, Stiel, and Thodos (AIChE J., Vol. 8, #59 (1962)):

$$((\mu(\text{mix}) - \mu(\text{low},\text{mix})) * \mu p(\text{mix}) + 10**(-4))**(1/4) = 0.10230 + 0.023364 * \text{rr} + 0.058533 * \text{rr} ** 2 - 0.040758 * \text{rr} ** 3 + 0.0093324 * \text{rr} ** 4$$

where rr is the reduced density of the mixture: rr = (critical molar volume)/(molar volume). This expression gives μ(mix) in centipoise.

*LE

Indicates that, as for the *HZYT option, the Jossi, Stiel, and Thodos formula is used to compute the mixture viscosity at reservoir conditions from the mixture viscosity at low pressure. However, under the *LE option the mixture low pressure viscosity is computed as a function of absolute temperature and average molecular weight using a correlation due to Lee and Eakin (SPEJ, Sept. 1964, 247-249):

$$\mu(\text{low}, \text{mix}) = \left(10 ** (-4) * (17.94 + 0.0321\text{M}) * \text{T} ** (3/2)\right) / \left(1.8\text{T} + 75.4 + 13.9 * \text{M}\right)$$

where T (K) is the absolute temperature and M is the average molecular weight of the mixture (g/gmol).

*PEDERSEN

Indicates that the corresponding states correlation given in Pedersen et al. (Chem. Eng. Sci., Vol. 39, No. 6, pp. 1011-1016, 1984) is used to compute the viscosity of the mixture, according to the following formula:

$$\frac{\mu_{mix}(P,T)}{\mu_o(P_o,T_o)} = \left(\frac{T_{c,mix}}{T_{c,o}}\right)^{-1/6} \left(\frac{P_{c,mix}}{P_{c,o}}\right)^{2/3} \left(\frac{MW_{mix}}{MW_o}\right)^{1/2} \left(\frac{\alpha_{mix}}{\alpha_o}\right)$$

where

μ	=	Viscosity
T_c	=	Critical temperature
P_c	=	Critical pressure
MW	=	Molecular weight
α	=	Rotational coupling coefficient

The subscript “mix” refers to the mixture property, and the subscript “o” refers to the reference substance property. The reference substance for the Pedersen model is methane.

The mixture critical temperature and pressure are calculated using mixing rules that are a function of the component critical temperatures and pressures, and mole fractions. The molecular weight of the mixture is determined from:

$$MW_{mix} = \text{coef}(1) \times \left(MW_w^{\text{coef}(2)} - MW_n^{\text{coef}(2)}\right) + MW_n$$

where MW_w is the weight fraction averaged molecular weight, and MW_n is the mole fraction averaged molecular weight.

The rotational coupling coefficient is calculated as follows:

$$\alpha = 1 + \mathrm{coef}(3) \times \rho_r^{\mathrm{coef}(4)} \mathrm{MW}^{\mathrm{coef}(5)}$$

where ρ_r is the reduced density of the reference substance.

*MODPEDERSEN

Indicates that a modified corresponding states correlation given in Pedersen and Fredenslund (Chem. Eng. Sci., Vol. 42, No. 1, pp. 182-186, 1987) is used to compute the viscosity of the mixture. This form of the correlation differs from the one described above under *PEDERSEN in the correlation used to compute the reference fluid viscosity, and in the default values for the coefficients coef(1) through coef(5).

DEFAULTS:

If *VISCOR does not appear in the data set, then *VISCOR *HZYT is assumed. If *VISCOSITY is encountered in the data set (see below), then the default is changed to *MIX. For GEM versions 2011.10 and earlier, if a non-zero temperature gradient was entered for the reservoir under the *TRES keyword (see above), then the default correlation was changed from *HZYT to *LE. For GEM version 2011.12 and later, the temperature dependence of the hydrocarbon mixture viscosity is computed throughout the reservoir with both *HZYT and *LE options, as well as the Pedersen models.

CONDITIONS:

If it appears, this keyword must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

In versions prior to 93.10, the *HZYT and *MIX options were the only ones available in GEM for hydrocarbon viscosity calculations; *HZYT was the default and *MIX could be invoked by entering component viscosities under the *VISCOSITY keyword. Data sets constructed for versions 93.03 and earlier still work in later versions exactly as they did with the earlier versions. The *VISCOR keyword was introduced with version 93.10 to allow the use of the Lee-Eakin and Yoon-Thodos formulas as alternative ways to compute the low pressure mixture viscosities. Beginning with version 2000.10, the *VISCOR keyword is also used to select the Pedersen or Modified Pedersen corresponding states viscosity correlations.

Values for the five coefficients in the Jossi-Stiel-Thodos formula which differ from the values given above may be entered under the *VISCOEFF keyword; see below. Also, with the *MIXVC keyword, the user may enter an exponent differing from the default value of one for the mixing-rule expression used to compute the mixture critical volume. The mixture critical volume is used in turn to compute the reduced density which is used to evaluate the Jossi-Stiel-Thodos formula.

Finally, special component critical volumes for use in the mixing-rule expression used to compute the mixture critical volume for use in the Jossi-Stiel-Thodos formula can be entered under the *VISVC keyword; see below.

For the *PEDERSEN or *MODPEDERSEN correlations, values of the correlation coefficients coef(1) through coef(5) may be entered as described under the *VISCOEFF keyword below.

Because evaluating the *HZYT expression at different temperatures in the reservoir entails evaluating exponential functions several times for each component and for each grid block in the reservoir, the *LE correlation may give faster run-times for models with varying temperature compared to the *HZYT model. Note that *LE is no longer the default correlation for non-zero temperature gradient or non-isothermal models, as was the case prior to GEM 2011.12.

Example:

```
*VISCOR *LE
```

Parameters for Computing Viscosity of Hydrocarbon Fluids (Optional) *VISCOSITY, *MIXINGRULE, *VISVC, *MIXVC, *VISCOEFF

PURPOSE:

Keywords to introduce the input of parameters used to compute the hydrocarbon phase viscosities.

*VISCOSITY specifies the component viscosities to be used in a direct mixing-rule evaluation of hydrocarbon viscosities.

*MIXINGRULE specifies the exponent to be used in the direct mixing-rule evaluation of hydrocarbon phase viscosities.

*VISVC specifies the component pseudo-critical volumes used to compute the mixture critical volume for use in the Jossi, Stiel, and Thodos formula.

*MIXVC specifies the exponent used to compute the mixture critical volume for use in the Jossi, Stiel, and Thodos formula.

*VISCOEFF specifies the coefficients for the Jossi, Stiel and Thodos correlation, or for the Pedersen or Modified Pedersen correlations.

FORMAT:

```
*VISCOSITY      amus
*MIXINGRULE     emixmu
-or-
*VISVC          vcmu
*MIXVC          emixvc
*VISCOEFF       coef(1) coef(2) coef(3) coef(4) coef(5)
```

DEFINITIONS:

amus
: Nc real numbers for the viscosities of the nc hydrocarbon components (cp | cp).

emixmu
: Mixing rule exponent for direct calculation of hydrocarbon phase viscosities. Dimensionless.

vcmu
: Parameters (special values for critical volumes) used for computing the mixture critical volume (m3/kmol) in the Jossi, Stiel and Thodos correlation.

emixvc
: Exponent in the critical volume mixing rule used to evaluate the Jossi, Stiel and Thodos correlation.

coef
: A set of 5 coefficients for the Jossi, Stiel and Thodos correlation (Reid et al, 1977), or for the Pedersen or Modified Pedersen correlation as described under *VISCOR.

DEFAULTS:

Within one EOS set, if *MIX is specified, then *VISCOSITY is required, but *VISVC, *MIXVC, and *VISCOEFF produce an error if encountered. If *VISCOSITY is encountered, then *VISCOR defaults to *MIX. If *VISCOSITY appears, then the nc values for amu must be supplied (there are no default values).

Default for emixmu is one.

Default for vcmu is the critical volume of the component.

Default for emixvc is one.

Defaults for the coef values for the Jossi-Stiel-Thodos correlation are:

0.1023, 0.023364, 0.058533, -0.040758, 0.0093324

Defaults for the coef values for the Pedersen correlation are:

0.291, 1.0, 7.747E-05, 4.265, 0.8579

Defaults for the coef values for the Modified Pedersen correlation are:

1.304E-04, 2.303, 7.378E-03, 1.847, 0.5173

CONDITIONS:

These keywords must be in the COMPONENT PROPERTIES keyword group, after the *MODEL and *NC keywords. Enter either the *VISCOSITY - *MIXINGRULE keyword set, or the *VISVC - *MIXVC - *VISCOEFF keyword set within one EOS set.

DO NOT enter elements of both in the same EOS set.

Note that when the Pedersen correlations are being used, *VISVC and *MIXVC are not required; if they are present in the data set, they will be ignored.

EXPLANATION:

There are five methods to calculate the oil and gas phase viscosities in GEM. If all viscosity keywords are missing, the Jossi, Stiel and Thodos correlation with the default coefficient values and the Herning-Zipperer and Yoon-Thodos formulas is used.

1. From a viscosity mixing rule using the entered component viscosities. In this case the viscosities depend only on the composition of the fluids but not on pressure and temperature.

   vis ** emixmu = sum (x(i) * amus(i) ** emixmu)

   where

vis	=	oil or gas phase viscosity (cp | cp);
sum	=	summation over all nc components; and
x(i)	=	the mole fraction of component i.

2. From the Jossi, Stiel and Thodos correlation (AIChE J., v. 8, p. 59 (1962)) with the mixture low-pressure viscosity evaluated using the Herning-Zipperer (Gas und Wasserfach , v. 79, p. 69 (1936)) and Yoon-Thodos (AIChE J., v. 16, pp. 300-304 (1970)) formulas (the *HZYT option under *VISCOR described above). Viscosities depend on the fluid composition and on grid block pressure, but not on temperature.

```
[ (vis-viso) * xi + 1.0e-04 ] ** 0.25 = coef(1) + coef(2) * dr +
    coef(3) * dr ** 2 + coef(4) * dr ** 3 + coef(5) * dr ** 4
```

where

vis	=	phase viscosity, cp
viso	=	ow-pressure fluid viscosity, cp, calculated internally from the Herning- Zipperer and Yoon-Thodos formulas
xi	=	tc ** (1/6) * mw ** (1/2) * pc ** (2/3)
tc	=	fluid pseudo-critical temperature, deg K
mw	=	fluid molecular weight, gm/gmol;
pc	=	fluid pseudo-critical pressure, atm
dr	=	fluid reduce-molar density (= den * vc);
den	=	fluid molar density, kmol/m3;

and

```
vc ** emixvc = sum ( x(i) * vcmu(i) ** emixvc)
```

with

sum	=	summation over all nc components; and
x(i)	=	the mole fraction of component i.

3. From the Jossi, Stiel, and Thodos formula with viso computed directly from the Lee-Eakin formula (SPEJ, Sept. 1964, 247-249) as a function of the mixture average molecular weight and the absolute temperature. This option is set by entering *VISCOR *LE; see above. Under this option the hydrocarbon phase viscosities are functions of fluid composition, pressure, and temperature.

4. From the Pedersen corresponding states correlation. This option is set by entering *VISCOR *PEDERSEN; see above. Under this option the hydrocarbon phase viscosities are functions of fluid composition, pressure, and temperature.

   The viscosity of a mixture calculated using the Pedersen model depends strongly on the critical pressures, critical temperatures and molecular weights of the components, as well as the coef values.

5. From the Modified Pedersen corresponding states correlation. This option is set by entering *VISCOR *MODPEDERSEN; see above. Under this option the hydrocarbon phase viscosities are functions of fluid composition, pressure, and temperature.

   Note that the Pedersen correlations are considerably more computationally intensive than the Jossi, Stiel and Thodos correlation, and may cause noticeable increases in run times. The magnitude of the increase will vary depending on the problem being solved.

   The default values of the coefficients for the Jossi, Stiel, and Thodos correlation were derived to reproduce GAS phase viscosities. Predicted oil phase viscosities may be less accurately predicted. If laboratory viscosity data are available, the coef's and the vcmu's of the components can be obtained by fitting the Jossi, Stiel and Thodos correlation to these experimental values using the regression option of WINPROP. The altered values of the coef's, vcmu's, and emixvc are used both for the *HZYT and the *LE options.

The *PEDERSEN and *MODPEDERSEN correlations were developed to provide good liquid viscosity predictions for light and medium gravity oils. The modified correlation was fit to more viscosity data in the range from 1 to 10 cP, and is expected to perform better than the original model in this range. If experimental viscosity data are available, the Pedersen correlations may be tuned to match the data by adjusting the coef values using the regression option in WINPROP.

Example:

The following are all valid viscosity specifications for a 7-components fluid:

```
*VISVC      0.099 0.134 0.225 0.305 0.463 0.820 1.216
*MIXVC      1.1
*VISCOEFF   0.1010 0.01869 0.04683 -0.04240 0.01120
-or-
*VISVC      0.099 0.134 0.225 0.305 0.463 0.820 1.216
*VISCOEFF   0.1010 0.01869 0.04683 -0.04240 0.01120
-or-
*VISCOSITY  0.0056 0.0073 0.0098 0.012 0.025
            0.032 0.054
```

Aqueous Phase Properties (Optional) *RHOW, *DENW, *CW, *REFPW, *VISW, *RHOWS, *DENWS, *SALINITY, *AQUEOUS-VISCOSITY, *AQUEOUS-DENSITY, *COMPNAME-SAL

PURPOSE:

These keywords are used to specify the properties of the aqueous phase, and the models used for property calculations.

FORMAT:

*RHOW	ρ_{h2o_ref}
*DENW	denw
*CW	cw
*REFPW	pref
*VISW	visw
*RHOWS	ρ_{h2o_std}
*DENWS	denws
*AQUEOUS-DENSITY	(*LINEAR | *ROWE-CHOU)
*AQUEOUS-VISCOSITY	*KESTIN | {*POLYMER (*NONLIN1 | *NONLIN2 | *LINEAR)}
*SALINITY	(*PPMWT | *PPMVOL | *MOLAL | *MOLAR | *WTFRAC | *MOLFRAC) salinity
*COMPNAME-SAL	canme_sal

DEFINITIONS:

RHOW

Specifies the molar density of water component at reservoir conditions.

ρ_{h2o_ref}

A real number for the molar density of the water component (gmol/m3 | gmol/ft3) at the reference pressure P_{ref} and reservoir temperature.

DENW

Specifies the mass density of the water component at reservoir conditions.

denw

A real number for the mass density of the water component (kg/m3 | lb/ft3) at the reference pressure refpw and reservoir temperature.

CW

Specifies the isothermal compressibility for the water component.

cw

Water component isothermal compressibility (1/kPa | 1/psia).

REFPW

Specifies the reference pressure for the water component density.

P_{ref}

Reference pressure (kPa | psia) used in the water component density equation.

VISW

Specifies a constant water viscosity value. To make water viscosity a function of pressure, temperature and salinity, use AQUEOUS-VISCOSITY.

visw

Constant aqueous phase viscosity (cP | cP).

RHOWS

Specifies the molar density of water component at surface conditions.

ρ_{h2o_std}

A real number for the molar density of the water component (gmol/m3 | gmol/ft3) at surface conditions of pressure and temperature.

DENWS

Specifies the mass density of water component at surface conditions.

denws

A real number for the mass density of the water component (kg/m3 | lb/ft3) at surface conditions of pressure and temperature.

AQUEOUS-DENSITY

Specifies the method to be used for calculating density of the aqueous phase at reservoir conditions. Aqueous density of the water component at the surface must be entered with RHOWS or DENWS, or defaulted as described below. The LINEAR density correlation is used for aqueous phase density calculation at surface or separator conditions.

LINEAR

Sub keyword of AQUEOUS-DENSITY indicating that the aqueous phase density at reservoir conditions will be calculated based on a correlation whereby the partial molar density of the water component is calculated as a linear function of pressure, P, i.e.

$$\rho_{h2o} = \rho_{h2o_ref} + (1 + C_w(P - P_{ref}))$$

Where ρ_{h2o_ref} is the density of pure water at reservoir temperature and reference pressure P_{ref} Values of ρ_{h2o_ref} can be entered by the user via keyword *RHOW, C_W via keyword *CW and P_{ref} via keyword *REFPW.

The density of the aqueous phase is also a function of the composition of the aqueous phase. The aqueous phase may contain soluble gaseous components such as CO2 and ionic components such as NaCl. The presence of these components influences the calculation of the phase density as outlined below in the EXPLANATION section. LINEAR is the default method for density calculations at reservoir conditions.

Please note that at surface (separator) conditions the LINEAR method is used to calculate aqueous phase density. Partial molar density of the water component is calculated as:

$$\rho_{h2o} = \rho_{h2o_std} + (1 + C_w (P - P_{std}))$$

Where ρ_{h2o_std} is the density of pure water at standard conditions, 15 deg C and 1 atm or 14.7 psia. Value of ρ_{h2o_std} can be entered by the user via keyword *RHOWS.

ROWE-CHOU

Subkeyword of AQUEOUS-DENSITY indicating that aqueous molar density will be calculated from the Rowe-Chou correlation.

AQUEOUS-VISCOSITY

Specifies the viscosity model for the aqueous phase. It must be followed by the keyword *KESTIN. If *AQUEOUS-VISCOSITY is not used, the aqueous phase viscosity is assumed to be constant and can be specified with the keyword *VISW.

KESTIN

Sub-keyword of the *AQUEOUS-VISCOSITY keyword. This indicates that the correlation of Kestin et al. will be used to calculate aqueous phase viscosity as a function of pressure, temperature and salinity. (Kestin et al., J. Phys. Chem. Ref. Data, Vol. 10, No. 1, p.71, 1981).

POLYMER

Sub-keyword of the *AQUEOUS-VISCOSITY keyword. This indicates that the viscosity of the aqueous phase will be computed using a mixing rule for the components present in the aqueous phase including the water (H_2O) component. Pure component polymer viscosity is computed using the *AVISC and *BVISC keywords. Contributions from non-polymeric components are currently ignored. Three models are available (*NONLIN1, *NONLIN2 and *LINEAR) to define the mixing rule as indicated below. See Appendix H for more details.

NONLIN1

Sub-keyword of the *POLYMER keyword. The non-linear mixing model is defined by a function which depends on the polymer mole fraction in the phase and user-defined weighting factors. A look-up table of weighting factors against mole fraction is generated using the *VSMIXENDP and *VSMIXFUNC keywords.

NONLIN2

Sub-keyword of the *POLYMER keyword. The non-linear mixing rule is similar to that for *NONLIN1. However, the weighting factors depend on the current polymer concentration and a reference concentration input by the user through *PREFCONC keyword.

LINEAR

Sub-keyword of the *POLYMER keyword. The mixing rule in this model is linear with respect to weighting factors which are computed using the current polymer concentration and a reference concentration input by the user through *PREFCONC keyword.

SALINITY

Specifies the salinity of the aqueous phase in terms of equivalent NaCl concentration for use with the AQUEOUS-VISCOSITY model.

PPMWT

Subkeyword of SALINITY indicating units of salinity will be parts per million by weight (mg NaCl / kg brine).

PPMVOL

Subkeyword of SALINITY indicating units of salinity will be parts per million by volume (mg NaCl / L brine).

MOLAL

Subkeyword of SALINITY indicating units of salinity will be molality (gmol NaCl / kg H2O).

MOLAR

Subkeyword of SALINITY indicating units of salinity will be molarity (gmol NaCl / L brine).

WTFRAC

Subkeyword of SALINITY indicating units of salinity will be weight fraction (g NaCl / g brine).

MOLFRAC

Subkeyword of SALINITY indicating units of salinity will be mole fraction (gmol NaCl / gmol brine).

salinity

A real number for the equivalent NaCl concentration of the aqueous phase. If no units subkeyword has been specified, the default unit for salinity is molality (gmol NaCl / kg H2O).

COMPNAME-SAL

An optional keyword which indicates input of salinity component names for calculation of effective or equivalent salinity. Any one or more than one of the aqueous components could be designated as salinity components in form of quoted character string(s) following this keyword.

cname_sal

A set of alphanumeric character strings to identify the names of the salinity components. One character string is required for each salinity component. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes and separate two character strings by at least one blank space. A minimum of one and a maximum of nc_aqu such strings are expected, where nc_aqu is the number of aqueous components.

DEFAULTS:

Optional keyword.

P_{std} = 1 atm or 14.7 psi

Default for ρ_{h2o_ref} and ρ_{h2o_std} is 55 556 mol/m3 or 1 573 gmol/ft3

Default for denw and denws is 1000.8 kg/m3 or 62.478 lb/ft3

Default AQUEOUS-DENSITY method is LINEAR.

Default for C_W is 4.35x10-7 kPa-1 or 3x10-6 psia-1

Default for P_{ref} is 101.325 kPa or 14.696 psia

Default for visw will be calculated from Bingham's correlation (Perry et al, 1977) as follows:

$$\frac{100}{\text{amus}} = 2.4182\left[\text{Tmid} + \text{sqrt}\left(8078.4 + \text{T} * \text{T}\right)\right] - 120$$

Where:

Tmid = (Mid-depth reservoir temperature in deg C) – 8.435

Default is to use constant aqueous viscosity, not AQUEOUS-VISCOSITY KESTIN or POLYMER.

Default for AQUEOUS-VISCOSITY POLYMER is NONLIN1.

Default for salinity is zero.

Default unit for salinity is *MOLAL (molality in gmol NaCl / kg H_2O).

If *COMPNAME-SAL and associated input is not present, salinity calculation will be done based on the NaCl, Na^+, and Cl^- ions. If present, *COMPNAME-SAL must follow at least one or at most nc_aqu character strings, that are defined as aqueous components via keyword *COMPNAME-AQUEOUS.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. Only one of *DENW or *RHOW is allowed, and only one of *DENWS or *RHOWS is allowed.

Since salinity components are a subset of aqueous components, it is necessary that *NC-AQUEOUS, *COMPNAME-AQUEOUS and *MW-AQUEOUS appear in the data set. *COMPNAME-SAL must not occur before the *NC-AQUEOUS keyword.

EXPLANATION:

At reservoir conditions, if the AQUEOUS-DENSITY method is LINEAR, the partial molar density of water component at any pressures other than P_{ref} is computed from the compressibility as:

$$\rho_{h2o} = \rho_{h2o_ref} + (1 + C_w(P - P_{ref}))$$

where P_{ref} is the oil phase pressure.

Input for *DENW is converted to ρ_{h2o_ref} first and the above formula is then used. The molecular weight of pure water, 18.015 g/gmol, is used for this conversion.

At surface or separator conditions the aqueous density calculation method is always LINEAR, the partial molar density of the water component at any standard temperature and at any other pressures other than P_{std} is computed from compressibility as:

$$\rho_{h2o} = \rho_{h2o_std} + (1 + C_w(P - P_{std}))$$

Where Pstd is the standard pressure of 1 atm or 14.7 psi.

Input for *DENWS is converted to ρ_{h2o_std} first and the above formula is then used. The molecular weight of pure water, 18.015 g/gmol, is used for this conversion.

If solubility in the aqueous phase is NOT turned on and there are NO aqueous components specified (via keyword NC-AQUEOUS) then the aqueous phase corresponds to pure component water and the density of the aqueous phase is equal to the partial molar density of component water calculated based on the equation(s) shown above.

Calculation of aqueous phase density with dissolved gas or ions in solution

If aqueous phase solubility is turned on then the aqueous phase can contain dissolved (soluble) species such as CO_2 which will alter the aqueous phase molar density. If aqueous ionic components such as NaCl are specified then the aqueous phase molar density will also be affected.

Calculation of the aqueous phase density with the presence of dissolved or ionic species is done in a similar manner at both reservoir and surface conditions. The composition of the aqueous phase at reservoir conditions may differ between reservoir conditions and surface conditions, for example gaseous components initially dissolved in aqueous phase will come out of solution upon production to surface. Ionic species such as NaCl remain in the aqueous phase as water phase is produced to surface. The equations below are for density calculation at reservoir conditions. The equations apply equally to surface aqueous phase density with the composition vector y_i referring to aqueous phase composition at surface conditions. The same values of partial molar volumes of gaseous soluble components such as CO2 are used for calculation at reservoir and surface conditions.

Partial molar volume of water component:

$$\rho_{h2o} = \rho_{h2o_ref} + (1 + C_w(P - P_{ref}))$$

$$V_{h2o} = y_{h2o}/\rho_{h2o}$$

Molar volume of aqueous phase:

$$V_{aqu} = V_{h2o} + \sum_{1}^{nsolaq} y_i V_i$$

Molecular weight of aqueous phase:

$$MW_{aqu} = y_{h2o} MW_{h2o} + \sum_{1}^{nsolaq+nc_aqu} y_i MW_i$$

Mass density of aqueous phase:

$$\text{den_aqu} = \frac{MW_{aqu}}{v_{aqu}}$$

Where

ρi is the partial molar density of component i

yi is mole fraction of component i in the aqueous phase

vi is the partial molar volume of component i

MWi is molecular weight of component i

nsolaq is the number of soluble gaseous species such as CO2

nc_aqu is the number of ionic species such as NaCl

and

den_aqu is the calculated density of the aqueous phase at reservoir conditions.

Note that the sum in the calculation of molar volume of aqueous phase spans from 1 to nsolaq, the number of soluble components in the aqueous phase, therefore there is no volume change on mixing of ionic components such as NaCl.

For surface water injection and production rates, conversion between volume and mole units is done using the molar volume rhows. Input for DENWS will be converted to rhows.

Example input (SI units) for a 100,000 ppm brine with reservoir P = 20 MPa and T = 77 C, and using the Kestin correlation for aqueous viscosity:

```
*DENW   1067.6
*CW     3.696E-07
*REFPW  20000.0
*VISW   0.4574
*DENWS  1045.7
*SALINITY  *PPMWT  100000
*AQUEOUS-VISCOSITY  *KESTIN
```

Example input for salinity using units of molality:

```
*SALINITY  1.9012
**   OR
*SALINITY  *MOLAL  1.9012
```

Viscosity in the presence of polymer component(s)

Polymers are used for mobility control primarily through an increase in the viscosity of the aqueous phase. The change in viscosity is a function of polymer concentration, temperature, pressure, salinity and shear effects. Refer to the following keywords for polymer viscosity related data: *AVISC-AQUEOUS, *BVISC-AQUEOUS, *VISCTABLE-AQUEOUS, *SHEARTHIN, *SHEARTHICK, *SHEARTAB and *VSSALTCMP. These keywords specify the data for the aqueous components only. To specify the viscosity of the component H_2O, use *AVISC-H2O and *BVISC-H2O. Finally, use *VSMIXENDP and *VSMIXFUNC or *PREFCONC keywords to specify the mixing rule for computation of the viscosity of the aqueous phase. Refer to Appendix-H for more details.

Calculation of Aqueous Phase Salinity

Salinity of aqueous phase is computed when *SALINITY-CALC *ON (see GHG Option: Aqueous Phase Model) is specified in the data set. In that event *COMPNAME-SAL may be used to specify salinity components out of the aqueous components in the data set. The salinity in molality (mol/kg H_2O) is then calculated as,

$$\tilde{y} = \sum_{1}^{nc_sal} y_i$$

$$\text{Salinity} = \frac{1000\,\tilde{y}}{y_{H_2O} MW_{H_2O}}$$

where,

nc_sal = number of salinity components, i.e., aqueous components that contribute to salinity

y_i = mole fraction of salinity component i in the aqueous phase.

MW_{H2O}= Molecular weight of component H_2O.

If *COMPNAME-SAL and associated data is absent but *SALINITY-CALC *ON is present, the salinity is calculated based on the composition of NaCl, Na^+, and Cl^- ions in the aqueous phase.

The salinity of each grid-block could be output to Simulation Results file, output file or for a particular grid block with special history using *SALIN sub keyword of *OUTSRF, *OUTPRN, or *OUTSRF *SPECIAL ‘block_id’ respectively.

Aqueous Phase Viscosity with Polymer Components (Conditional) *AVISC-AQUEOUS, *BVISC-AQUEOUS *VISCTABLE-AQUEOUS

PURPOSE:

These keywords define the viscosity of components in the aqueous phase.

FORMAT:

```
*AVISC-AQUEOUS          avisc1 … avisc_nc_aqu
*BVISC- AQUEOUS         bvsic1 … bvisc_nc_aqu
or
*VISCTABLE-AQUEOUS
   viscosity_T_table
or
*VISCTABLE-AQUEOUS
   {*ATPRES pres
     viscosity_T_table}
```

DEFINITIONS:

$avisc_i$, $bvisc_i$

Coefficients for the correlation for temperature dependence of component viscosity in the aqueous phase. The unit of $avisc_i$ is cp (viscosity). The unit of $bvisc_i$ is temperature difference which is the same as the unit for temperature. Number of aqueous components is denoted by nc_aqu (as defined by *NC-AQUEOUS) and does not include the water component.

The correlation for the viscosity of component i, μ_{Aqi}, is

$$\mu_{Aqi} = avisc_i \,.\, e^{bvisc_i / T}$$

where T is absolute temperature. $avisc_i$ and $bvisc_i$ must be non-negative. A value of zero for $bvisc_i$ specifies a temperature-independent value of $avisc_i$ cp.

viscosity_T_table

Table of viscosity versus temperature, with each row in the form

$$T \; visc_1 \ldots visc_{nc_aqu}$$

T

Row temperature (C | F)

$visc_i$

Viscosity(cp) for each aqueous component at the row temperature T. A total of *nc_aqu* values expected.

*ATPRES *pres*

Subkeyword *ATPRES indicates that presence of a viscosity_T_table for pressure pres (kPa | psi). This allows specification of viscosity as a function of temperature and pressure. Multiple instances of *ATPRES are allowed. The range of temperature in the tables following *ATPRES can be different.

If the first viscosity_T_table table contains the *ATPRES subkeyword, all subsequent viscosity_T_table tables must have the *ATPRES subkeyword. There can be only one viscosity_T_table table if the *ATPRES subkeyword is not used at all.

DEFAULTS:

If *AVISC-AQUEOUS is specified but *BVISC- AQUEOUS is absent, $bvisc_i$ is set equal to zero.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components. *AVISC- AQUEOUS or *VISCTABLE-AQUEOUS is required. *BVISC- AQUEOUS is optional but is used only when AVISC-AQUEOUS is specified.

EXPLANATION:

The correlation for computing component viscosity using the correlation in $avisc_i$ and $bvisc_i$, stipulates that the logarithm of the component viscosity varies linearly with inverse absolute temperature:

$$\ln(\mu_i) = \ln(avisc_i) + bvisc_i/T$$

The data entered via *AVISC- AQUEOUS and *BVISC- AQUEOUS are for pure components. A “pure component” polymer can be viewed as the binary polymer-water solution with the maximum polymer concentration anticipated, usually the concentration of the injected solution. It is the concentration of the polymer solution for which the mixing function, $f_{i,p}$ (see below), is unity.

In the table option without the *ATPRES keyword the viscosity is computed as follows:

Interpolate viscosity value between two consecutive temperatures in the table that bracket the given temperature. The interpolation is carried out with respect to inverse temperature as per the expression given above for $\ln(\mu_i)$.

If values are entered via *ATPRES keyword the viscosity is computed as follows:

1. Compute viscosity values as given above for the table option without the *ATPRES keyword for two consecutive tables whose pres values bracket the given pressure.
2. Perform a linear interpolation with respect to pressure between the two values of viscosity computed in Step 1 above.

Phase Mixing Rules – Non-linear

The aqueous phase viscosity is computed from the “pure component” viscosities using the following expression:

$$\ln(\mu_{aq}) = f_{H_2O}.\ln(\mu_{H_2O}) + \sum_{i=1}^{nc_plm} f_{i,p}.\ln(\mu_i) + \sum_{i=nc_plm+1}^{nc_aqu} f_{i,np}.\ln(\mu_i)$$

In the above expression, f_{H_2O} is the weighting factor for water component (H_2O) in the aqueous phase. μ_{H_2O} is the viscosity of H_2O. μ_i is the viscosity of pure component i. For a polymer, the viscosity μ_i includes the effects of temperature, salinity and shear. $f_{i,p}$ and $f_{i,np}$ are the weighting factors for the component i depending on whether it is a polymer or a non-polymer. Number of polymer components is denoted by nc_plm while nc_aqu is the total number of aqueous components, excluding H_2O. *The third term on the right hand side of the above expression is ignored in the current version of the code*.

Refer to the *AQUEOUS-VISCOSITY keyword and Appendix-H for details of the viscosity models.

Phase Mixing Rule - Linear

The aqueous phase viscosity is computed from "pure component" viscosities using the following expression:

$$\mu_{aqu} = f_{H_2O} \cdot \mu_{H_2O} + \sum_{i=1}^{nc_plm} f_{i,p} \cdot \mu_i + \sum_{i=nc_plm+1}^{nc_aqu} f_{i,np} \cdot \mu_i$$

In the above expression, the variables are the same as defined for the non-linear mixing rules.

Viscosity calculation – sequence of steps

1. Compute the pure component viscosity for H_2O (*AVISC-H2O/*BVISC-H2O/*VISCTABLE-H2O)
2. Compute "pure component" viscosities of aqueous components(*AVISC-AQUEOUS/*BVISC- AQUEOUS /*VISCTABLE- AQUEOUS)
3. Modify "pure component" polymer viscosity due to effect of salinity (*VSSALTCMP)
4. Modify "pure component" polymer viscosity due to shear effects (*SHEARTHIN/*SHEARTHICK/*SHEARTAB)

Compute aqueous phase viscosity using values of "pure component" water (H2O) and aqueous components and the specified mixing rule. The viscosities for the polymer components are those that have been modified in steps 3 and 4.

Nonlinear Mixing Rule for Aqueous Phase Viscosity in Presence of Polymer Components *VSMIXENDP, *VSMIXFUNC

PURPOSE:

Specify nonlinear mixing rule to compute aqueous phase viscosity with the *NONLIN1 sub-keyword of *AQUEOUS-VISCOSITY *POLYMER keyword.

FORMAT:

*VSMIXENDP	comp_name	x_{low} x_{high}
*VSMIXFUNC	comp_name	$f_1 \ldots f_{11}$

DEFINITIONS:

comp_name
: Quoted name of the polymer component for which the data is entered.

x_{low}
: Polymer mole fraction corresponding to the first entry in the table generated from *VSMIXENDP and *VSMIXFUNC. It is the lower end point mole fraction and corresponds to the first entry (f_1) of *VSMIXFUNC. The allowed range is 0 to x_{high}.

x_{high}
: Polymer mole fraction corresponding to the last entry in the table generated from *VSMIXENDP and *VSMIXFUNC. It is the higher end point mole fraction and corresponds to the last entry (f_{11}) of *VSMIXFUNC. The allowed range is from x_{low} to 1.

$f_1 \ldots f_{11}$
: Eleven weighting factors that correspond to the eleven polymer mole fractions generated from x_{low} and x_{high} endpoints specified via *VSMIXENDP.

DEFAULTS:

If *VSMIXENDP is absent, $x_{low} = 0$ and $x_{high} = 1.0$ are assumed.

If *VSMIXFUNC is absent, eleven equally spaced entries from 0 to 1 (both inclusive) are generated.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components.

EXPLANATION:

The aqueous phase viscosity is computed using weighting factors and pure component viscosity values using the following relation:

$$\ln \mu_{aq} = f_i \ln \mu_i$$

where μ_{aq} is the aqueous phase viscosity, f_i and μ_i are the weighting factor and pure component viscosity of aqueous component i including polymers and the water component H_2O. Data for polymers entered via *VSMIXENDP and *VSMIXFUNC are used to compute the weighting factors for polymers. Refer to Appendix H for more details.

Effect of Salinity on Polymer Viscosity (Conditional)

***VSSALTCMP**

PURPOSE:

This keyword defines the effect of salinity on the viscosity of polymer in the aqueous phase.

FORMAT:

*VSSALTCMP comp_name x_{min} sp

DEFINITIONS:

comp_name
: Quoted name of the *polymer* component for which the data is entered.

x_{min}
: Mole fraction of NaCl below which the polymer viscosity is unaffected by the salinity. Allowed range is 10^{-5} to 10^{-1}.

sp
: Slope on a log-log plot of polymer component viscosity versus ratio of salinity to minimum salinity mole fraction. Allowed range is 0 to 1.

DEFAULTS:

If the keyword is absent, salinity effect on the viscosity is not calculated for the polymer component.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components.

EXPLANATION:

The effect of salinity on polymer viscosity is computed using the following expression:

$$\mu_p = \mu_p^0 \cdot \left(\frac{x_{NaCl}}{x_{min}} \right)^{sp} \qquad \text{for } x_{NaCl} > x_{min}$$

$$\mu_p = \mu_p^0 \qquad \text{for } x_{NaCl} \le x_{min}$$

where

μ_p^0 is the “pure component” viscosity (refer to manual page for *AVISC-AQUEOUS and *VISCTABLE-AQUEOUS); and

μ_p is the polymer viscosity including the effect of salinity

Convert Velocities to Shear Rates (Conditional) *SHEAR_FAC

PURPOSE:

Convert fluid Darcy velocities to equivalent porous media shear rates.

FORMAT:

*SHEAR_FAC factor

DEFINITIONS:

factor

Factor γ_{fac} in shear rate equation described below.

DEFAULTS:

The default value is zero. A warning is issued if the value is not input or is set to zero.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group. It is used with *SHEARTAB keyword provided *SHEAREFFEC is set to *SHR. It is also used to convert Darcy velocity to shear rate for output.

EXPLANATION:

The equation relating porous media shear rate and fluid Darcy velocity is

$$\dot{\gamma} = \frac{\gamma_{fac}\left|u_l\right|}{\sqrt{kk_{r,l}\phi S_l}}$$

where k and ϕ are absolute permeability and porosity, u_l is the Darcy velocity, $k_{r,l}$ is the relative permeability and S_l is the saturation of phase l and $\dot{\gamma}$ is the shear rate.

Viscosity Shear Dependence Option (Optional) *SHEAREFFEC

PURPOSE:

Specify viscosity dependence option for shear dependent effects on viscosity of aqueous polymer solutions.

FORMAT:

*SHEAREFFEC (*SHV | *SHR)

DEFINITIONS:

*SHV

Viscosity shear depends on Darcy velocity.

*SHR

Viscosity shear depends on shear rate.

DEFAULTS:

If *SHEAREFFEC is absent, *SHEAREFFEC *SHV is assumed.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group.

EXPLANATION:

Shear thinning or thickening data specified via keywords *SHEARTHIN, *SHEARTHICK or *SHEARTAB must correspond to shear dependence relation defined by keyword *SHEAREFFEC.

The relationship between effective porous media shear rate and fluid Darcy velocity is shown under the Explanation of keyword *SHEAR_FAC.

Shear Effects on Polymer Viscosity (Conditional) *SHEARTHIN, *SHEARTHICK

PURPOSE:

These keywords define the effect of shear rate on the viscosity of polymer in the aqueous phase.

FORMAT:

*SHEARTHIN	'comp_name'	n_{thin}	$u_{aq,lower}$	
*SHEARTHICK	'comp_name'	n_{thick}	$u_{aq,max}$	$\mu_{aq,max}$

DEFINITIONS:

'comp_name'

Polymer component name within quotes for which shear-thinning or shear-thickening parameters are being specified. A maximum of 8 characters, excluding the quotes, is expected.

n_{thin}

Exponent in the viscosity thinning equation (dimensionless). The allowed range is 0.1 to 0.99, inclusive. Values close to 1.0 approximate Newtonian behavior.

$u_{aq,lower}$

u in this context depends on the value of SHEAREFFEC:

*SHERAEFFEC *SHV: Darcy velocity (m/day | ft/day) of the aqueous phase

*SHEAREFFEC *SHR : shear rate (1/day | 1/day)

$u_{aq,lower}$ is the velocity (shear rate) at which the viscosity of the polymer fluid is equal to the viscosity of the polymer fluid at zero shear. Viewed another way, it is the "first" velocity (shear rate) at which the viscosity of the fluid deviates from Newtonian behavior.

n_{thick}

Exponent in the viscosity thickening equation (dimensionless). The allowed range is 1.01 to 5, inclusive. Values close to 1.0 approximate Newtonian behavior.

$u_{aq,max}$

Velocity (shear rate) at which the viscosity is the maximum in the shear thickening equation.

$\mu_{aq,max}$

Maximum viscosity in the shear thickening equation (cp | cp).

DEFAULTS:

No defaults. *SHEARTHIN and *SHEARTHICK can be used together to describe the combined effect of thinning and thickening.

*SHERATHIN and *SHEARTHICK must not be used with *SHEARTAB.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components.

Shear thinning or thickening data (velocity or shear rate) specified via keywords *SHEARTHIN, *SHEARTHICK must correspond to shear dependence relation defined by keyword *SHEAREFFEC.

EXPLANATION:

For Newtonian fluids, a plot of shear stress versus shear rate is a straight line passing through the origin. The slope of this line is the viscosity of the fluid. Viscosity of a Newtonian fluid is constant and is independent of the shear. Fluids that do not obey this relationship between shear stress and shear rate are called non-Newtonian.

Polymer solutions typically exhibit non-Newtonian behavior. Three types of non-Newtonian behavior are observed: shear thinning, shear thickening and combined.

If the slope of the shear stress versus shear rate plot decreases with increasing shear rate, the solution is exhibiting shear thinning since the apparent viscosity (the slope) is reducing with increased shear rate. If, however, the slope (and therefore the viscosity) is increasing, the solution is exhibiting shear thickening. In some solutions, the viscosity may decrease and then start increasing as the shear rate is increased, thus exhibiting a combined effect.

It has also been observed that the viscosity of a polymer solution may follow Newtonian behavior at low as well as at high shear rates and in the intermediate range of shear rate, the viscosity can be approximated by a power-law type equation.

GEM provides all the three models for computing the effect of shear on polymer viscosity. Refer to Appendix-H for details.

Shear Effects on Polymer Viscosity – Tabular input (Conditional) *SHEARTAB

PURPOSE:

This keyword defines the effect of shear rate on the viscosity of polymer component in the aqueous phase in a tabular format.

FORMAT:

*SHEARTAB	‘comp_name’	velocity	viscosity
or			
*SHEARTAB	‘comp_name’	shear-rate	viscosity

DEFINITIONS:

*SHEARTAB
: A viscosity versus aqueous phase velocity (shear rate) table follows.

‘comp_name’
: Polymer component name within quotes for which shear-table is being specified. A maximum of 8 characters, excluding the quotes, is expected.

velocity
: Aqueous phase velocity (m/day | ft/day)

shear-rate
: Shear rate (1/day | 1/day)

viscosity
: Viscosity value (cp | cp)

DEFAULTS:

No defaults. *SHERATHIN and *SHEARTHICK must not be used with *SHEARTAB.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components.

Shear thinning or thickening data (velocity or shear rate) specified via keyword *SHEARTAB must correspond to shear dependence relation defined by keyword *SHEAREFFEC.

EXPLANATION:

This option is useful when the relationship between velocity (shear rate) and viscosity is specified by lab data or when a simple power-law type relationship is not available.

Implicit calculation of shear effects (Conditional)

***IMPLICIT-SHEAR**

PURPOSE:

This keyword directs the simulator to update the velocity of the aqueous phase in the inner (Newton) loop. The velocity is used for computing the effect of shear on polymer viscosity.

FORMAT:

*IMPLICIT-SHEAR

DEFINITIONS:

*IMPLICIT-SHEAR
Update the aqueous phase velocity once every Newtonian iteration.

DEFAULTS:

By default, the velocity calculation for shear calculations is performed once every time step iteration.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group. It is used only when aqueous phase contains polymer components.

EXPLANATION:

This option is useful when tightly coupled calculations are required for the shear effect of aqueous phase viscosity. However, these calculations incur additional time overhead since the computations are done at Newton rather than at timestep level.

Reference Polymer Concentration (Conditional) *PREFCONC

PURPOSE:

This keyword specifies the reference concentration for a polymer component in the aqueous phase.

FORMAT:

*PREFCONC ‘comp_name’ value

DEFINITIONS:

‘comp_name’
Name of the polymer component

value
Reference polymer concentration (kg/m3 | lb/ft3)

DEFAULTS:

No defaults.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components. Required with *AQUEOUS-VISCOSITY *POLYMER *LINEAR and *AQUEOUS-VISCOSITY *POLYMER *NONLIN2.

EXPLANATION:

The reference concentration is used to compute non-linear and linear phase mixing functions for aqueous phase viscosity in the presence of polymers. Refer to the manual page for *AVISC-AQUEOUS, *BVISC-AQUEOUS and*VISCTABLE-AQUEOUS.

Solubility Data (Optional) *HENRYC, *REFPH, *VINFINITY, *SOLUBILITY, *HENRY-CORR-CO2, *HENRY-CORR-N2, *HENRY-CORR-H2S, *HENRY-CORR-C1

PURPOSE:

These keywords define the solubility of components in the aqueous phase.

FORMAT:

*SOLUBILITY	(*HENRY) (*IDEAL)
*HENRYC	hens
*REFPH	refph
*VINFINITY	v8

The following keywords activate enhanced solubility models for specific components, making the Henry's constant a function of temperature, pressure and salinity:

*HENRY-CORR-CO2
*HENRY-CORR-N2
*HENRY-CORR-H2S
*HENRY-CORR-C1

DEFINITIONS:

*SOLUBILITY

Indicates that solubility of components in the aqueous phase may be modelled. If Program Components are being used, no further input for solubility is required. If solubility of User Components is required, Henry's constants must be specified using the keywords HENRYC/VINFINITY/REFPH or HENRY-CORR-XXX. Entering the SOLUBILITY keyword automatically activates *OGW_FLASH *NO_H2OVAP.

*HENRY

Subkeyword of *SOLUBILITY indicating that the general Henry's Law will be used in calculating the solubilities of gases in the aqueous phase.

*IDEAL

Subkeyword of *SOLUBILITY indicating that ideal behavior is assumed in calculating the solubilities of gases in the aqueous phase.

hens

A set of nc real numbers for the reference Henry's law constants (kPa | psia) of the nc components at reference pressure refph. Enter zero for components insoluble in water.

refph

A set of nc real numbers representing the reference pressures (kPa | psia) for the Henry's law constants of the nc components. Enter zero for components insoluble in water. However zero is also a valid value of refph for components whose solubility is to be modelled (indicated by a non-zero value of hens).

v8

A set of nc real numbers for the partial molar volume of the nc components in water at infinite dilution. (l/mol | l/mol). The units for v8 are in litre per mol irrespective of the specified reservoir unit. If zero is entered for a component with a non-zero hens value, a default value will be calculated.

*HENRY-CORR-CO2

This keyword activates the use of Harvey's correlation for CO_2 Henry's constant. Use of this option makes the Henry's constant a function of pressure, temperature and salinity.

*HENRY-CORR-N2

This keyword activates the use of Harvey's correlation for N_2 Henry's constant. Use of this option makes the Henry's constant a function of pressure, temperature and salinity.

*HENRY-CORR-H2S

This keyword activates the use of Harvey's correlation for H_2S Henry's constant. Use of this option makes the Henry's constant a function of pressure, temperature and salinity.

*HENRY-CORR-C1

This keyword activates the use of Harvey's correlation for CH_4 Henry's constant. Use of this option makes the Henry's constant a function of pressure, temperature and salinity.

DEFAULTS:

Optional keywords. See Explanation below for more information.

By default there is no solubility of components in the aqueous phase.

If Program Components are specified, the SOLUBILITY keyword must be entered to activate the library values for hens, refph and v8.

If User Components are specified, there are no default parameters for solubility in the aqueous phase. Solubility parameters must be entered with HENRYC/VINFINITY/REFPH and/or HENRY-CORR-XXX. When these parameters are entered, SOLUBILITY HENRY is turned on automatically. The only reason to use the SOLUBILITY keyword in this case is to specify IDEAL.

If *SOLUBILITY is entered without any subkeyword, *HENRY is assumed.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

The keywords *HENRYC and *REFPH must appear together; i.e., if either one is present, the other one must be entered as well.

HENRY-CORR-XXX can be used by itself or in conjunction with HENRYC. HENRY-CORR-XXX takes precedence over any data specified with HENRYC.

EXPLANATION:

The solubility in the aqueous phase is modelled by the Henry's Law. Two versions are available.

1. The general Henry's Law, which uses the following fugacity formula:

   $f = x * H$

   where

f	=	the fugacity of the component;
x	=	composition of the component in the aqueous phase; and
H	=	Henry's Law constant.

   The Henry's Law constants, H, at any pressures, p, are calculated as follows:

   $\ln (H) = \ln (hens) + v8 * (p-refph)/RT$

   where ln is the natural log operation. If hens for a component is entered as zero, the component will not be soluble in water. The above general Henry's Law will be used when *SOLUBILITY or *SOLUBILITY *HENRY is entered, or when HENRYC/VINFINITY/REFPH and/or HENRY-CORR-XXX are specified without the *SOLUBILITY keyword.

2. The simplified Henry's Law, used when *SOLUBILITY *IDEAL is entered, makes an ideal gas simplification leading to the following assumption:

   $z * p = x * H$

   where

z	=	global composition of the hydrocarbon phases in equilibrium with the aqueous phase. If both a gas and an oil phase are present, z is calculated by mixing both the oil and gas into one phase
p	=	pressure

   H is computed as above. This approach allows for the reappearance of a gas phase from the aqueous phase when pressure declines.

When HENRY-CORR-XXX is specified, a correlation from Harvey (1996) is used for the reference Henry's constant (hens) and the reference pressure (refph) is set to the saturation pressure of water at the specified temperature. Additional temperature-dependent correlations are used for molar volume at infinite dilution (v8). The formula given in (1.) above is then used to calculate the Henry's constant H. Finally, a salting out coefficient is applied to H if any of the salinity options (*SALINITY, *SALIN-CALC or *SALINR) have been specified.

When hydrocarbon components are soluble in the aqueous phase, the component conservation equations need to be modified. Refer to Appendix A for more information.

The solubility option is activated by either of the two sets of keywords:

1. The *SOLUBILITY keyword without the *HENRYC-*REFPH-*VINFINITY or HENRY-CORR-XXX keyword set. In this case, default solubility data will be used for Program Components.

   Default data allow solubility for the following Program Components:

CH4 C2H6 C3H8 IC4 NC4 IC5 NC5 NC6 NC7 NC8 N2 CO2 H2S

The actual values used are from Li and Nghiem (1986). The rest of the Program Components and ALL User Components are assumed to be insoluble. Refer to *COMPNAME for identifications of user components.

2. The *HENRYC - *REFPH - *VINFINITY and/or HENRY-CORR-XXX keyword set. In this case the user-entered solubility data will be used. If the simplified Henry's Law is to be used enter *SOLUBILITY *IDEAL; otherwise, the *SOLUBILITY keyword is not really required as *SOLUBILITY *HENRY will be assumed.

Consider the following 4-components system examples:

```
(1)
*NC  4
*COMPNAME  'CO2'  'CH4'  'NC4'  'NC10'
*SOLUBILITY
** As default values are used, CO2, CH4 and NC4
** are soluble in water and NC10 insoluble.
** The general Henry's Law will be used.
```

```
(2)
*NC  4
*COMPNAME  'CO2'  'CH4'  'NC4'  'NC10'
*HENRYC      54073.  901790.  0.  0.
*REFPH         14.7     14.7  0.  0.
*VINFINITY   0.0356   0.0358  0.  0.
** CO2 and CH4 are soluble.
** NC4 and NC10 are insoluble.
** The general Henry's Law will be used.
```

```
(3)
*NC  4  3
*COMPNAME  'CO2'  'METHANE'  'BUTANE'  'C9+'
*SOLUBILITY
** Default solubility data are used. Only CO2 is
** soluble. The rest are user-components and are
** insoluble.
** The general Henry's Law will be used.
```

```
(4)
*NC  4  3
*COMPNAME  'CO2'  'METHANE'  'BUTANE'  'C9+'
*SOLUBILITY *IDEAL
** Default solubility data are used. Only CO2 is
** soluble. The rest are user-components and are
** insoluble.
** The simplified Henry's Law will be used.
```

```
(5)
*NC  4  3
*COMPNAME  'CO2'  'METHANE'  'BUTANE'  'C9+'
*SOLUBILITY
*HENRY-CORR-CO2
*HENRYC      54073.  901790.  0.  0.
*REFPH         14.7     14.7  0.  0.
*VINFINITY   0.0356   0.0358  0.  0.
** CO2 and METHANE are soluble.
** BUTANE and C9+ are insoluble.
** The general Henry's Law will be used.
** The enhanced correlation for CO2 will be used.
```

Gas-Phase Diffusion in Fractured Reservoirs (Optional)

***DIFFUSION**

PURPOSE:

*DIFFUSION activates the gas diffusion option for fractured reservoirs and introduces the input of the necessary parameters for this option.

FORMAT:

*DIFFUSION tortuo (diffus(k), k = 1, nc)

DEFINITIONS:

tortuo

A positive real number giving the tortuosity of the porous medium. Tortuosity is the ratio of the true path length traveled by a particle flowing through the medium to the macroscopic distance traveled. A typical measured value for tortuosity is 1.5, but when used as a parameter in the diffusion option, the tortuosity should be regarded as an adjustable parameter available to match observed transfer rates. Larger values of tortuosity slow the rate at which gas flows due to diffusion.

diffus

A vector of nc positive real numbers, where nc is the number of primary hydrocarbon components. These numbers are the diffusion coefficients (cm2/sec) for the hydrocarbon components. Note that these coefficients MUST have the units cm2/sec no matter what unit system is used for the simulation. True molecular diffusivities may be entered, but it is preferable to regard these coefficients as adjustable parameters which are used to reproduced observed transfer rates.

DEFAULTS:

Optional keyword. Default is no inclusion of a diffusion mechanism in computing the matrix-fracture molecular fluxes. The default value for the tortuosity is one and the default value for diffus(k) is zero, k = 1, ..., nc . If the diffusion effect is to be considered, all values (tortuo, diffus(k), k = 1, nc) must be entered.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

This keyword activates the inclusion of a gas phase diffusion term in the molecular flux between a fracture block and its collocated matrix block.

The gas phase diffusion rate (mol/day) for component k from block i to collocated block j is computed as

$$\text{rate} = (\text{Area/Separation}) * (\text{diffus(k)/tortuo}) * \text{phi} * \text{Sg} * \left(C(k, \text{gas}, i) - C(\text{gas}, j)\right)$$

Here Area is the contact area between blocks i and j, Separation is the distance between blocks i and j (computed from the fracture spacings), phi is the porosity of the matrix block, Sg is the smaller of the gas saturations in blocks i and j, C(k,gas,i) is the concentration of component k in the gas phase of block i (moles per unit volume of the gas phase) and C(k,gas,j) is the same for block j.

Values of true molecular diffusivities vary with temperature, pressure (through molar density), and molecular weight. At low pressures, the diffusivity is roughly directly proportional to the square root of the Kelvin temperature, inversely proportional to pressure, and inversely proportional to the square root of the molecular weight. At higher pressures, which are typical of reservoir conditions, the dependencies are more complicated and cannot be simply characterized.

Typical values for diffusivities for a methane-propane system at 160 degrees F and pressures between 1500 and 3000 psia lie between 20 * 10 ** (-5) and 100 * 10 ** (-5) cm2/sec.

Note that many processes other than true molecular diffusion may contribute to the transport which one may want to model using this diffusion mechanism. We recommend that diffusivities be adjusted to fit the transport being modelled.

Flash Method Technique (Optional) *FLASH-METHOD, *QNSS-TO-NR, *FLASH-METHOD-OGW

PURPOSE:

*FLASH-METHOD specifies whether the thermodynamic equations should be solved at the same iteration level as the flow equations (*FLASH-METHOD *OUTER) or should be solved to convergence in every Newtonian iteration in a flash subroutine (*FLASH-METHOD *INNER). It has been found that using *FLASH-METHOD *OUTER can lead to significant savings in CPU time.

*QNSS-TO-NR specifies whether after a certain point flash convergence method will switch from Quasi-Newton-Successive-Substitution (QNSS) to Newton-Raphson (NR) to achieve an efficient convergence.

*FLASH-METHOD-OGW turns on a special three phase flash technique that can be used when aqueous phase solubility is being modeled. When the keywords *SOLUBILITY or *HENRY-CORR-XXX are used for solubility modelling, *FLASH-METHOD-OGW (equivalent to *OGW-FLASH *NO_H2OVAP) is turned on automatically. If the keywords *HENRYC-*REFPH-*VINFINITY appear without the *SOLUBILITY keyword, *FLASH-METHOD-OGW is not turned on automatically.

FORMAT:

*FLASH-METHOD	(*INNER)		
	(*OUTER)	(*KVAL)	(*NONE)
		(*LOG-KVAL)	(*SPLIT)
			(*FLASH) (epsfla)
			(*FLASHF)
*QNSS-TO-NR	(epsswitch)	(iterswitch)	
*FLASH-METHOD-OGW			

DEFINITIONS:

*INNER

Subkeyword which indicates that the thermodynamic equations are solved to convergence in every Newtonian iteration in a flash subroutine. This is the only method which was available in versions of before GEM 95.00. There are no further subkeywords for *FLASH-METHOD *INNER and *INNER should be the last character string on the line. *INNER is NOT the default.

*OUTER

Subkeyword which indicates that the thermodynamic equations should be solved in the Newtonian iterations along with the flow equations. Two sets of subkeywords can be used with *FLASH-METHOD *OUTER, as indicated in the FORMAT line above. *FLASH-METHOD *OUTER is the default.

*KVAL

Subkeyword of *OUTER. Indicates that K-values themselves, rather than their natural logarithms, should be used as thermodynamic variables. *KVAL is NOT the default.

*LOG-KVAL

Subkeyword of *OUTER. Indicates that natural logarithms or K-values, rather than the K-values themselves, should be used as thermodynamic variables in the outer iterations. *LOG-KVAL is the default.

*NONE

Subkeyword of *OUTER. Indicates that the thermodynamic variables from the Newtonian iteration are used without refinement to compute properties. *NONE is NOT the default.

*SPLIT

Subkeyword of *OUTER. Indicates that the vapor mole fraction from the Newtonian iteration is refined by solving the material balance equation to convergence with the given K-values. *SPLIT is NOT the default.

*FLASH

Subkeyword of *OUTER. Indicates that the thermodynamic variables as updated in the Newtonian iteration are refined by several QNSS (flash) iterations before properties are computed. *FLASH is the default.

epsfla

If *FLASH is specified, epsfla is the tolerance for the QNSS iterations: the sum of squared residuals is required not to exceed epsfla. The default value is epsfla = 1.0D-04 .

*FLASHF

Subkeyword of *OUTER. Indicates that after the update from a Newtonian iteration, the thermodynamic variables are determined by running a full flash to convergence, possibly including a stability test. Run in this way, the *OUTER iteration is nearly identical to the *INNER iteration. *FLASHF is NOT the default.

epsswitch

If *QNSS-TO-NR is specified, the flash convergence process can switch from QNSS to Newton-Raphson when sum of square residuals is larger than epsswitch after iterswitch QNSS iterations. The default for epsswitch is 1.0D-6. Values no larger than 1.D-04 and no smaller than 1.D-12 can be specified.

iterswitch

If *QNSS-TO-NR is specified, the flash convergence process can switch from QNSS to Newton-Raphson when sum of square residuals is larger than epsswitch after iterswitch QNSS iterations. The default for iterswitch is 50. Values no smaller than 20 and no larger than 100 can be specified.

DEFAULTS:

Optional keyword. Defaults for individual subkeywords are as indicated above. If *FLASH-METHOD is omitted altogether, or entered without any subkeywords, the default is *FLASH-METHOD *OUTER *LOG-KVAL *FLASH 1.0D-04.

*QNSS-TO-NR is an optional keyword for switching from QNSS to NR during flash convergence. The default is not to switch to NR, but try to achieve convergence with the QNSS procedure.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

The *OUTER method avoids the extra computational work of solving the thermodynamic equations to full convergence in every Newtonian iteration and therefore allows the simulation to consume less CPU time. The user can usually allow this keyword to be defaulted since internal checks prevent *OUTER from being applied where it would not work.

*QNSS-TO-NR allows switching of flash convergence method from QNSS to NR at a specified convergence tolerance and at a specified number of QNSS iterations. Normally convergence with the default QNSS procedure is adequate for majority of the fluid systems, however, for certain difficult fluids (e.g. near critical region) it may be advantageous to switch to the NR method after QNSS has obtained a fairly good estimate for initiating the NR method. This option should be used judiciously to avoid extra computational overhead. The number of flash convergence iterations for each grid block could be obtained via *ITERVLE sub-keyword of *OUTPRN *GIRD or *OUTSRF *GRID. Also see related sub keyword *NBLKFFAIL of *OUTSRF *SPECIAL.

*FLASH-METHOD-OGW turns on a special three phase flash technique that can be used when aqueous phase solubility is being modelled When the keywords *SOLUBILITY or *HENRY-CORR-XXX are used for solubility modelling, *FLASH-METHOD-OGW is turned on automatically. If the keywords *HENRYC,*REFPH,*VINFINITY appear without the *SOLUBILITY keyword, *FLASH-METHOD-OGW is not turned on automatically.. Use of this option may over-ride the setting for *DERIVATIVE-METHOD.

Example:

```
*FLASH-METHOD *INNER
```

this would cause the simulation to be run using the flash algorithm and inner iterations as all simulations were run with GEM versions 94.10 and earlier.

Method for Evaluating Entries of the Jacobian Matrix (Optional) *DERIVATIVEMETHOD

PURPOSE:

*DERIVATIVEMETHOD is functionally a numerical methods keyword but is so closely tied in its use to the *FLASH-METHOD keyword that it is included in the COMPONENT PROPERTIES section. If *FLASH-METHOD *OUTER is specified or defaulted, the user can choose to have the derivatives of the residuals of the flow equations (which are entries in the Jacobian matrix) evaluated either analytically or numerically using this keyword.

If *FLASH-METHOD *INNER has been specified, the derivatives of the flow residuals are always evaluated numerically.

FORMAT:

*DERIVATIVEMETHOD	(*MIXED)
	(*NUMERALL)
	(*NUMERFLOW)

DEFINITIONS:

*MIXED

Subkeyword which indicates that the Jacobian entries are to be computed analytically except for derivatives of well residuals and derivatives of block flow residuals with respect to well bottom-hole pressures. These latter two types of entry are evaluated numerically. This method can be applied ONLY if *FLASH-METHOD *OUTER has been specified or defaulted.

*NUMERALL

Subkeyword which indicates that the flow and well residuals are to be differentiated numerically, and that the thermodynamic residuals are differentiated numerically as well.

*NUMERFLOW

Subkeyword which indicates that the flow and well residuals are to be differentiated numerically. If *FLASH-METHOD *INNER has been specified, these two types of entry are the only ones appearing in the Jacobian; if *FLASH-METHOD *OUTER has been specified or defaulted and *NUMERFLOW is specified, the thermodynamic residuals are differentiated analytically.

DEFAULTS:

Optional keyword. If *FLASH-METHOD *OUTER is specified or defaulted, the default is *DERIVATIVEMETHOD *MIXED . If *FLASH-METHOD *INNER is specified, the default is *DERIVATIVEMETHOD *NUMERFLOW.

Certain special GEM options require *NUMERALL.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. If *FLASH-METHOD *INNER has been specified, the *MIXED subkeyword is ignored.

EXPLANATION:

Analytical differentiation avoids much variable shifting and property re-computation and thus saves CPU time. However it is not easily implemented when the thermodynamic equations are solved in the inner iteration, and is supported in GEM only for the *OUTER iteration.

The user can safely let this keyword be defaulted unless special options are being used that required *NUMERALL.

The four distinct allowed combinations of the *FLASH-METHOD and *DERIVATIVEMETHOD keywords are the following:

```
*FLASH-METHOD *INNER
*DERIVATIVEMETHOD *NUMERFLOW
```

and

```
*FLASH-METHOD *OUTER
*DERIVATIVEMETHOD *NUMERFLOW
```

and

```
*FLASH-METHOD *OUTER
*DERIVATIVEMETHOD *NUMERALL
```

and

```
*FLASH-METHOD *OUTER
*DERIVATIVEMETHOD *MIXED
```

The last is the default.

Example:

```
*DERIVATIVEMETHOD *NUMERFLOW
```

this would cause the simulation to be run using the numerical derivatives for all flow and well residuals. If *FLASH-METHOD *INNER had been specified, this line would simply re-establish the default; if *FLASH-METHOD *OUTER were specified this line would override the default and cause flow residuals and all terms involving wells to be differentiated numerically, which may increase the CPU time consumed.

Molecular Diffusion (Optional) *DIFCOR-OIL, *DIFCOR-GAS, *DIFCOR-AQU, *DIFFC-OIL, *DIFFC-GAS, *DIFFC-AQU, *RESISTIVITY, *TORPSDIFF, *TORNOPSDIFF, *TORNOPS2DIFF

PURPOSE:

*DIFCOR-OIL, *DIFCOR-GAS and *DIFCOR-AQU select a correlation type for the molecular diffusion option.

*DIFFC-OIL, *DIFFC-GAS and *DIFFC-AQU define coefficients that specify molecular diffusion rates.

*RESISTIVITY defines a resistivity factor for use with the molecular diffusion option.

*TORPSDIFF, *TORNOPSDIFF and *TORNOPS2DIFF select different options controlling the tortuosity model.

FORMAT:

*DIFCOR-OIL	*SIGMUND
*DIFCOR-GAS	*SIGMUND
*DIFCOR-OIL	*WILKE
*DIFFC-OIL	D^*_{io}
*DIFFC-GAS	D^*_{ig}
*DIFFC-AQU	D^*_{iw}
*RESISTIVITY	resistivity_factor
*TORPSDIFF	
*TORNOPSDIFF	
*TORNOPS2DIFF	

DEFINITIONS:

*SIGMUND

Specifies the Sigmund correlation for molecular diffusion. Applicable to oil and gas phases.

*WILKE

Specifies the Wilke-Chang correlation for oil phase molecular diffusion.

D^*_{io}

A vector of nc positive real numbers, where nc is the number of primary hydrocarbon components, that specify oil phase molecular diffusion constants in cm^2/s. Note that these coefficients MUST have the units cm^2/sec no matter what unit system is used for the simulation.

D^*_{ig}

A vector of nc positive real numbers, where nc is the number of primary hydrocarbon components, that specify gas phase molecular diffusion constants in cm^2/s. Note that these coefficients MUST have the units cm^2/sec no matter what unit system is used for the simulation.

D^*_{iw}

A vector of nc positive real numbers, where nc is the number of primary hydrocarbon components, that specify aqueous phase molecular diffusion constants in cm^2/s. Note that these coefficients MUST have the units cm^2/sec no matter what unit system is used for the simulation.

resistivity_factor

A value used in representing tortuosity effects.

DEFAULTS:

Optional keyword. The default is no inclusion of this molecular diffusion mechanism. The TORPSDIFF tortuosity model is the default one.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

These keywords activate modelling of molecular diffusion in the reservoir model. The parameters determining the amount of diffusion can be calculated by two different correlations for the oil phase, by a single correlation for the gas phase, and can be entered directly (always in cm2/sec) for the oil, gas and aqueous phases.

The Sigmund method (Sigmund, 1976) for calculating the molecular diffusion [cm^2/s] is as follows. The binary diffusion coefficient between Component i and j in the mixture is:

$$D_{ij} = \frac{\rho_k^0 \, D_{ij}^0}{\rho_k} \cdot \left(0.99589 + 0.096016\,\rho_{kr} - 0.22035\,\rho_{kr}^2 + 0.032874\,\rho_{kr}^3\right)$$

with

$$\rho_{kr} = \rho_k \cdot \left(\frac{\sum_{i=1}^{nc} y_{ik}\, v_{ci}^{5/3}}{\sum_{i=1}^{nc} y_{ik}\, v_{ci}^{2/3}} \right) \quad \text{and} \quad \rho_k^0 D_{ij}^0 = \frac{0.0018583\, T^{1/2}}{\sigma_{ij}^2\, \Omega_{ij}\, R} \cdot \left(\frac{1}{M_i} + \frac{1}{M_j} \right)^{1/2} .$$

The diffusion of Component i in the mixture is:

$$D_{ik} = \frac{1 - y_{ik}}{\sum_{j \neq i} y_{ik}\, D_{ij}^{-1}}$$

In the above equations, the collision diameter σ_{ij} and the collision integral Ω_{ij} of the Lennard-Jones potential are related to the component critical properties through the following equations (Reid et al, 1977):

$$\sigma_i = \left(2.3551 - 0.087\,\omega_i\right)\cdot\left(\frac{T_{ci}}{p_{ci}}\right)^{1/3} \qquad \varepsilon_i = k_B\left(0.7915 + 0.1963\,\omega_i\right)T_{ci}$$

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \qquad \varepsilon_{ij} = \sqrt{\varepsilon_i\,\varepsilon_j} \qquad T^*_{ij} = \frac{k_B}{\varepsilon_{ij}}$$

$$\begin{aligned}\Omega_{ij} = 1.06306\left(T^*_{ij}\right)^{-0.15610} &+ 0.19300\exp\left(-0.47635\,T^*_{ij}\right)\\ &+ 1.03587\exp\left(-1.52996\,T^*_{ij}\right)\\ &+ 1.76474\exp\left(-3.89411\,T^*_{ij}\right)\end{aligned}$$

where k_B is the Boltzmann's constant (= 1.3805E-16 ergs/K).

The Wilke-Chang molecular diffusion coefficient [cm^2/s] correlation is (Wilke and Chang, 1955):

$$D_{ik} = \frac{7.40E-08\left(M'_{ik}\right)^{1/2}\,T}{\mu_k\,v_{bi}^{0.6}} \quad \text{with} \quad M'_{ik} = \frac{\sum_{j\neq i} y_{jk}\,M_j}{1 - y_{ik}}$$

The viscosity μ_k[cp] is calculated from the Lorentz, Bray, and Clark correlation, and the partial molar volume of Component i at the boiling point, v_{bi}[cm^3/mol], is estimated from the Tyn and Calus method (Reid et al, 1977):

$$v_{bi} = 0.285\,v_c^{1.048}$$

where v_c is the critical volume [cm^3/mol].

Total dispersion is the mixing of fluids caused by diffusion, local velocity gradients, locally heterogeneous streamline lengths, and mechanical mixing (Lake, 1989). The dispersive flux J_i of Component i represented by a concentration y is given by:

$$J_i = -\sum_k \phi\rho_k S_k \mathbf{D}_{t,ik}\nabla y_{ik}, \quad i = 1,\ldots,n_c;\ \ k = o, g, w$$

where the total dispersion tensor components are given by expressions of the form:

$$D_{t,ik,mm} = \frac{D_{ik}}{\tau_k} + \frac{\left(\alpha_{kl} - \alpha_{kt}\right)}{\phi\,S_k\left|u_k\right|}u_{km}^2 + \alpha_{kt}\frac{\left|u_k\right|}{\phi\,S_k}$$

$$D_{t,ik,mn} = \frac{\left(\alpha_{kl} - \alpha_{kt}\right)}{\phi\,S_k\left|u_k\right|}u_{km}\,u_{kn}$$

where D_{ik} is the diffusion coefficient discussed above. F_k is the formation resistivity factor, the α_{kl} and α_{kt} are the longitudinal and traverse dispersivity parameters for the dispersion model and μ_k is the viscosity, all for phase k.

Keywords for controlling dispersion (that is, setting the longitudinal and tangential dispersivities) are presented in the Rock-Fluid section (*DISPERARRAY-...). Only a single (phase-independent) resistivity factor F can be used in the simulator, and it is set using *RESISTIVITY.

The tortuosity is the ratio of the true path length traveled by a particle flowing through the medium to the macroscopic distance it travels. If *TORPSDIFF is selected (which is the default if no *TOR…DIFF keyword appears), then the tortuosity of phase k is related to the formation resistivity factor F using the formula:

$$\tau_k = F \times \phi \times S_k$$

If *TORNOPSDIFF is selected, then $\tau_k = F$ is used. *TORNOPS2DIFF is equivalent to using *TORNOPSDIFF and also replacing the term

$$\left(\phi \rho_k S_k D_{t,ik} \nabla y_{ik}\right) \text{ by } \left(\phi S_k D_{t,ik} \nabla\left(\rho_k y_{ik}\right)\right)$$

in the expression for the flux J_k.

Note that when diffusion coefficients are being read in directly (*DIFFC-), they should be regarded as adjustable parameters that need to be tuned to give acceptable results. Indeed, laboratory values may not correspond to what is needed for large grid blocks used in reservoir simulation.

Fully implicit runs may be required (*AIMSET *CON 3; see *AIMSET) when molecular diffusion is being modelled. Use of the keywords mentioned in this section may over-ride settings for *DERIVATIVE-METHOD.

Asphaltene Precipitation Model (Optional) *SOLIDMODEL, *SOLIDTHERMAL, *SOLID_TREF

PURPOSE:

These keywords allow the user to enter parameters related to the asphaltene precipitation model. These parameters may all be generated using WinProp and automatically exported with the GEM fluid model.

FORMAT:

*SOLIDMODEL	ref_fugacity volsmd ref_pressure volsad
*SOLIDTHERMAL	hfusld del_cpsld t_triple p_triple
*SOLID_TREF	tref

DEFINITIONS:

*SOLIDMODEL
: Specifies the four parameters required by the isothermal asphaltene precipitation model implemented in GEM, as follows:

ref_fugacity
: Natural logarithm of the reference fugacity (ln(atmospheres)).

volsmd
: Solid phase molar volume (L/gmol).

ref_pressure
: Reference pressure corresponding to the reference fugacity (kPa | psia).

volsad
: Molar volume of solid phase used for calculating the pore volume occupied by deposited/adsorbed solid. This is usually equal to volsmd (L/gmol).

*SOLIDTHERMAL
: Specifies the parameters necessary to use the asphaltene precipitation model for non-isothermal predictions. SOLID_TREF must be used with this keyword.

hfusld
: Heat of fusion of the solid at the triple point (cal/gmol).

del_cpsld
: Difference between the heat capacities of the asphaltene component in the liquid state and in the solid state, $C_{p,liquid} - C_{p,solid}$ (cal/gmol K).

t_triple
: Triple point temperature of the solid component (C | F).

p_triple
: Triple point pressure of the solid component (kPa | psia).

*SOLID_TREF

Indicates input of the reference temperature for the precipitation model. This keyword is required if the non-isothermal precipitation model (SOLIDTHERMAL) is used. For isothermal predictions this keyword is optional, the reference temperature will default to the reservoir temperature.

tref

Reference temperature corresponding to the reference fugacity (C | F)

DEFAULTS:

Optional keywords. The default is to not include asphaltene precipitation in the simulation. When the*SOLIDMODEL keyword is used then all four associated parameters are required (no internal defaults). If *SOLIDTHERMAL is used, *SOLIDMODEL and *SOLID_TREF must also be specified.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. The asphaltene precipitation model implemented in GEM considers the solid phase to be a pure component and this component should be the last hydrocarbon component specified in the component list.

EXPLANATION:

The solid or asphaltene phase is represented by a single component in the thermodynamic model implemented in GEM. The oil and gas phase component fugacities are calculated by the EOS, while the solid component fugacity is obtained from the following equation:

$$\ln f_s = \ln f_s^* + \frac{v_s}{R}\left[\frac{p - p_{tp}}{T} - \frac{p^* - p_{tp}}{T^*}\right]$$
$$- \frac{\Delta H_{tp}}{R}\left[\frac{1}{T} - \frac{1}{T^*}\right] - \frac{\Delta C_p}{R}\left[\ln\left(\frac{T^*}{T}\right) - T_{tp}\left(\frac{1}{T} - \frac{1}{T^*}\right)\right]$$

where f_s is the fugacity at pressure p and temperature T, f_s^* is the reference fugacity at reference conditions of pressure p^* and temperature T^*, v_s is the solid phase molar volume of the component, ΔC_p is the solid-liquid heat capacity difference, ΔH_{tp} is the heat of fusion at the triple point, p_{tp} and T_{tp} are the triple point pressure and temperature, and R is the universal gas constant.

For isothermal predictions, this equation can be simplified to give:

$$\ln f_s = \ln f_s^* + v_s (p - p^*) / RT$$

For a detailed discussion on the steps required to generate values of these parameters please refer to the WinProp User Guide, specifically the chapter on Flash Calculations under the sub-section entitled “Asphaltene/Wax Modelling”. There is also a case study on asphaltene precipitation included in the appendix of the WinProp manual. This case study describes in detail a step-by-step procedure culminating in the generation of an EOS model for GEM that includes the parameters of the solid model for a typical oil.

Use of these options may over-ride the setting for *DERIVATIVE-METHOD.

Solid Adsorption (Optional) *ADSCST, *WSLMAX, *RFMAX

PURPOSE:

*ADSCST, *WSLMAX and *RFMAX control the modelling of the adsorption of the solid asphaltene phase, which is entrained in the oil phase, onto the reservoir rock. *SOLIDMODEL must have been used to invoke the asphaltene model.

FORMAT:

*ADSCST	adscst
*WSLMAX	wslmax
*RFMAX	rfmax

DEFINITIONS:

*ADSCST

Introduces the parameter for the Langmuir isotherm model, where adscst should be expressed in units of (g / μg or 1/ppm).

*WSLMAX

Specifies the maximum mass fraction of adsorbed solid per unit mass of rock, in dimensionless units of (kg-solid/kg-rock).

*RFMAX

Maximum resistance factor associated with the build-up of the adsorbed solid on the rock. Its value should be at least 1.0.

DEFAULTS:

Optional keywords. RFMAX defaults to 1.0.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

These keywords activate modelling of solid adsorption in the reservoir using a Langmuir isotherm model. An extended Langmuir isotherm for multicomponent adsorption (Arri et al., 1992; Hall et al., 1994) is used elsewhere in GEM, but it simplifies to the following for solid adsorption:

$$w_i = \frac{w_{i,max}\, C_i\, c_{io}}{1 + C_i\, c_{io}}$$

where

w_i	=	mass of adsorbed solid per unit mass of rock (kg-solid/kg-rock)
$w_{i,\,max}$	=	maximum mass of adsorbed solid per unit mass of rock (kg-solid/kg-rock)
c_{io}	=	concentration of solid in the oil phase (μg / g or ppm)
C_i	=	parameter for Langmuir isotherm (g / μg or 1/ppm)

Here 'i' refers to the single component associated with the solid phase and 'o' refers to its concentration in the oil phase. The references to 'rock' in this section refer to the actual rock material found in the reservoir, considered in isolation from its pore space.

Once the solid phase has adsorbed on the reservoir rock, partial plugging of the formation is expected. GEM uses a simple model for these phenomena based on the following resistance factor:

$$rfact = 1.0 + (rfmax-1.0)\ w_i\ /\ w_{i,\ max}$$

This factor is divided into each of the gas/oil/aqueous phase mobilities, thereby reducing the volumetric flow rates for all flowing phases.

Adsorption of solid asphaltene phase calculations require use of actual rock density, not including pores, which is taken from the block-wise input of rock density via keyword *ROCKDEN in the Rock-Fluid Data section.

Use of this option may over-ride the setting for *DERIVATIVE-METHOD.

Irreversible Asphaltene Model (Optional) *SOLID-CONV-RATE

PURPOSE:

This keyword allows the user to enter forward and reverse rates for the conversion of precipitated asphaltene to flocculated asphaltene. This conversion may be made reversible, fully irreversible or partially irreversible. Asphaltene "flocs" may deposit in the porous media as described in the section on the "Asphaltene Deposition Model".

FORMAT:

*SOLID-CONV-RATE for_rate back_rate

DEFINITIONS:

for_rate
: A real number for the forward reaction rate (1/day) for the conversion of precipitated asphaltene to flocculated asphaltene particles.

back_rate
: A real number for the backward reaction rate (1/day) for the conversion of flocculated asphaltene particles back to precipitated asphaltene.

DEFAULTS:

Optional keyword. The default is to not include flocculation of asphaltene precipitate in the simulation. When the *SOLID-CONV-RATE keyword is used then all three associated parameters are required (no internal defaults).

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. For the SOLID-CONV-RATE model to be active, the SOLIDMODEL keyword must be defined. When the SOLID-CONV-RATE model is used, deposition of asphaltene flocs will be calculated as described in the section on the "Asphaltene Deposition Model", not with the Langmuir adsorption model.

EXPLANATION:

The asphaltene precipitation model with parameters as input using the *SOLIDMODEL keyword exhibits complete thermodynamic reversibility; precipitated asphaltene will go back into solution if the system is returned to a thermodynamic state outside the asphaltene precipitation envelope.

Irreversibility of solid precipitates is modelled by allowing the thermodynamic asphaltene precipitate (solid s_1) to be transformed via a simple reversible chemical reaction into another solid, s_2. This can be viewed as the flocculation of smaller asphaltene particles into larger aggregates. The reaction may be written as follows:

$$s_1 \leftrightarrow s_2$$

The reaction rate for the formation of s_2 is:

$$r = k_{12}\, C_{s_1,o} - k_{21}\, C_{s_2,o}$$

where

k_{12}	=	forward rate of formation of solid s_2 from s_1 [day^{-1}]
k_{21}	=	reverse rate of formation of solid s_1 from s_2 [day^{-1}]
r	=	reaction rate [mol/(m^3 day)]
$C_{s1,o}$	=	concentration of suspended solid s_1 in oil phase [mol/m^3]
$C_{s2,o}$	=	concentration of suspended solid s_2 in oil phase [mol/m^3]

Note that s_2 can go back into solution by first becoming s_1 through the reverse reaction, and then dissolving into the oil phase through thermodynamic equilibrium. If k_{21} is zero, the reaction is irreversible and s_2 will not go back into solution. If $k_{21} << k_{12}$, the precipitation of s_2 is reversible, but may take a long time to complete. The above chemical reaction allows the modelling of irreversible precipitation or a slow redissolution of the precipitated asphaltene.

Use of this option may over-ride the setting for *DERIVATIVE-METHOD.

Solid Deposition Model (Optional) *SOLID_ALPHA, *SOLID_BETA, *SOLID_CRITVEL, *SOLID_GAMMA, *SOLID_SIGMA

PURPOSE:

These keywords control the deposition of flocculated solid particles. The conversion of precipitated to flocculated asphaltene particles is controlled with the *SOLID-CONV-RATE keyword.

FORMAT:

*SOLID_ALPHA	alpha
*SOLID_BETA	beta
*SOLID_CRITVEL	critvel
*SOLID_GAMMA	gamma
*SOLID_SIGMA	sigma

DEFINITIONS:

alpha
: Surface deposition rate constant (1/day).

beta
: Entrainment rate constant (1/m|1/ft).

critvel
: Critical interstitial velocity (m/day|ft/day).

gamma
: Instantaneous pore throat plugging deposition rate constant (1/m|1/ft).

sigma
: Snowball effect deposition constant (dimensionless).

DEFAULTS:

Optional keywords. The default is for no deposition to occur.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. For the deposition model to be active, the *SOLIDMODEL and *SOLID-CONV-RATE keywords must be defined.

EXPLANATION:

The SOLID-CONV-RATE keyword controls the generation of flocculated solid particles from the thermodynamic asphaltene precipitate model. The keywords described in this section control the deposition of the flocculated solid.

An equation relating asphaltene deposition rate to the primary physical deposition processes (adsorption, pore throat plugging and re-entrainment) was presented by Wang and Civan (2001). In the implementation of the deposition model, only the flocculated particles (solid s_2) are considered to deposit. Physically, this implies that the small asphaltene precipitate

particles are more likely to flow with the oil phase, while the larger aggregates which are formed according to the irreversible asphaltene rate equation are more likely to deposit on the reservoir rock. The discretized form of the deposition rate equation is:

$$\frac{V_{s_2^d}^{n+1} - V_{s_2^d}^{n}}{\Delta t} - \alpha\, C_{s_2^f}^{n+1}\, \phi^{n+1} + \beta\, V_{s_2^d}^{n+1} \left(v_o^n - v_{cr,o}\right) - \gamma\, u_o^n\, C_{s_2^f}^{n+1} = 0$$

where

$V_{s_2^d}$	=	volume of deposited solid s_2 per grid block volume
$C_{s_2^f}$	=	volumetric concentration of flowing solid s_2 per volume of oil
v_o	=	oil phase interstitial velocity
$v_{cr,o}$	=	critical oil phase interstitial velocity
u_o	=	oil phase Darcy velocity
α	=	surface deposition rate coefficient
b	=	entrainment rate coefficient
γ	=	pore throat plugging rate coefficient

The surface deposition rate coefficient α is a positive constant and is dependent on the rock type. The entrainment rate coefficient is set to zero when the interstitial velocity is less than the critical interstitial velocity, and may be set to a positive constant otherwise. The pore-throat plugging coefficient is set to zero if the average pore throat diameter is larger than some critical value. If it is smaller, the coefficient is calculated as:

$$\gamma = \gamma_i \left(1 + \sigma\, V_{s_2^d}\right)$$

where

γ_i	=	instantaneous pore throat plugging rate coefficient
σ	=	snowball-effect deposition constant

This formulation allows the plugging deposition rate to increase proportionally with the total amount of deposits.

Once the solid phase has adsorbed on the reservoir rock, partial plugging of the formation is expected. If the keywords RFMAX and WSLMAX are defined, the resistance factor equation described in the Solid Adsorption section will be used. The user may also choose different equations as specified by *RFCALC (*PERM-VS-POR), which may cause an amount of solid greater than WSLMAX to be deposited, therefore the resistance factor calculated may be greater than RFMAX.

Disallow Solid-Model Inheritance (Optional) *INHERIT-SOLID-OFF

PURPOSE:

This keyword is used to disallow inheritance of solid model parameters from EOS set 1 to subsequent EOS sets.

FORMAT:

*INHERIT-SOLID-OFF

DEFINITIONS:

This keyword takes no parameters or qualifiers.

DEFAULTS:

Optional keyword. The default is to have inheritance of solid-model parameters defined in EOS set 1 to subsequent EOS sets. This keyword, if present, changes the default behavior so that solid-model parameters of EOS set 1 are not inherited by other EOS sets.

CONDITIONS:

This keyword, if present, must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords and before the first occurrence of EOSSET keyword.

EXPLANATION:

This keyword is useful in a data set where asphaltene modeling is being done and multiple EOS sets are defined using keyword EOSSET. Normally asphaltene solid model parameters are specified in EOS set 1 and subsequent EOS sets inherit these parameters. If necessary some overrides are then applied with individual solid model keywords defined in EOS set 2 and higher.

However when keyword *INHERIT-SOLID-OFF is present no inheritance of solid model parameters takes place from EOS set 1 to other EOS sets. Thus, for instance, one may easily have a solid model in EOS set 1, no solid model in EOS set 2, and a different solid model in EOS set 3 than one in EOS set 1, using keyword *INHERIT-SOLID-OFF.

Coal Matrix-to-Cleat Modelling Parameters(Optional)

***LANG-DIFFUSION-COAL, *FREE-DIFFUSION-COAL, *NO-WBLOCK-COAL, *UNI-WBLOCK-COAL, *DBL-WBLOCK-COAL**

PURPOSE:

*LANG-DIFFUSION-COAL and the older *FREE-DIFFUSION-COAL select the type of coal-cleat (matrix-fracture) diffusion modelling to be used for Coal Bed Methane (CBM) and Extended CBM (ECBM) modelling.

*NO-WBLOCK-COAL, *UNI-WBLOCK-COAL and *DBL-WBLOCK-COAL control water blocking functions for coal-cleat (matrix-fracture) diffusive flows.

FORMAT:

These keywords take no parameters or qualifiers.

DEFINITIONS:

*LANG-DIFFUSION-COAL
: Specifies concentrations based on Langmuir adsorption are used to calculate concentration gradients for diffusive flow calculations.

*FREE-DIFFUSION-COAL
: Specifies concentrations based on free gas densities are used to calculate concentration gradients for diffusive flow calculations.

*NO-WBLOCK-COAL
: Specifies that coal-cleat (matrix-fracture) diffusive fluxes are not modified by the presence of water in the cleats.

*UNI-WBLOCK-COAL
: Specifies that the basic coal-cleat (matrix-fracture) diffusive fluxes are multiplied by $S_{g,cleat} = (1 - S_{w,cleat})$ when calculating the gas fluxes from cleats to the matrix, and that unmodified fluxes are used for flows out of the coal to the cleat system.

*DBL-WBLOCK-COAL
: Specifies that the basic coal-cleat (matrix-fracture) diffusive fluxes are multiplied by $S_{g,cleat} = (1 - S_{w,cleat})$ when calculating the gas flux between the coal and cleat systems, in any direction.

DEFAULTS:

*LANG-DIFFUSION-COAL and *UNI-WBLOCK-COAL are the defaults.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. They only have an effect if *COAL-DIF-*TIME or *COAL-DIF-COMP have been specified in the Rock-Fluid section (CBM/ECBM modelling enabled).

EXPLANATION:

GEM changed its CBM/ECBM (coal bed methane) default diffusion model starting at release GEM 2003.00. The new default model was available in some earlier versions by using the keyword *LANG-DIFFUSION-COAL. Earlier CBM/ECBM diffusion behavior can be recovered in versions GEM 2003.00 and later by using the keyword *FREE-DIFFUSION-COAL.

Determining the proper level of diffusion for coal bed methane modelling when working with desorption times often requires history matching; see the description of the *COAL-DIF-TIME keywords in the Rock-Fluid section. Nevertheless, GEM will match the reported values for canister desorption times better when using the new default diffusion model. Of course, if an older version of GEM was calibrated to actual observed experimental data, or field rates, and these data sets are to be used in a 2003.00 or later model, specify *the FREE-DIFFUSION-COAL keyword to obtain the old behavior. It is recommended that the new default diffusion model be used for all new projects.

The *FREE-DIFFUSION-COAL diffusion formulation models coal-to-fracture (cleat) diffusion based on the following formula for the diffusive mass flow rate of a species k in the gas phase for a grid cell:

Q(Default,k) = Vol * [Shape * Diffus(k)] * F(Sg) * (C(k,m)- C(k,f))

where:

Vol	Cell volume;
Shape	Inverse area;
Diffus(k)	Diffusion value;
F(Sg)	Function of fracture gas saturation modelling water blocking;
C(k,f)	Mass concentration of species k in the free gas phase in the cleats;
C(k,m)	Mass concentration of species k in the free gas phase in the micro-pore space within the matrix system.

Note that GEM allows the grouping [Shape * Diffus(k)] to be input as compositionally-dependent values using the *COAL-DIF-TIME keyword (more precisely, the input values specify the inverse of that grouping); see the discussion relating to the *COAL-DIF-TIME keyword. *COAL-DIF-COMP inputs the Diffus(k) values directly and Shape is derived based on the spacing between cleats entered using the *DIFRAC/DJFRAC/DKFRAC keywords.

This older *FREE-DIFFUSION-COAL model assumes that matrix-fracture diffusion is being controlled by a standard Fickian model for diffusion in free gas, with an empirical value for the diffusion coefficient accounting for the presence of coal.

The new default model (obtained using the *LANG-DIFFUSION-COAL keyword in certain older versions) often appears in the literature and can be described as follows:

Q(Lang,k) = Vol * [Shape * Diffus(k)] * F(Sg) * (Lang(k,m)- Lang(k,f))

where:

Lang(k,m)	Extended Langmuir isotherm for the coal, multiplied by coal density, evaluated at matrix composition and pressure;
Lang(k,f)	Extended Langmuir isotherm for the coal, multiplied by coal density, evaluated at fracture composition and pressure.

Note that Lang(k,·) is described using the usual *ADGCSTC and *ADGMAXC keywords from the Rock-Fluid section, and the *ROCKDEN keyword, that are entered for the *MATRIX system. No *FRACTURE adsorption is expected CBM/ECBM.

The *NO-WBLOCK-COAL, *UNI-WBLOCK-COAL and *DBL-WBLOCK-COAL keywords account for the fact that water present in the cleats could affect the flow of gas between the coal and the cleats. The default model (*UNI-WBLOCK-COAL) assumes that water present in the cleats could reduce the contact area for gas trying to flow from the cleats into the coal, while having no effect as cleat depressurization causes gas to evolve from the coal. The other two keywords offer alternate models where water is never considered (*NO-WBLOCK-COAL) and water always interferes regardless of flow direction (*DBL-WBLOCK-COAL).

Gas Adsorption for Water Filled Blocks (Optional)

***ADS_WATERZONE**

PURPOSE:

*ADS_WATERZONE determines whether the grid block gas saturation is to be used in the equation calculating the amount of a particular gaseous species adsorbed onto reservoir rock at a particular pressure and for a given gas composition. Adsorption is modeled using equations based on the multi component Langmuir model.

FORMAT:

This keyword takes no parameters or qualifiers.

DEFINITIONS:

*ADS_WATERZONE
: Specifies that the grid block gas saturation should be included in the formula used to calculate gas phase adsorption.

DEFAULTS:

Optional keyword. The default is to not include the gas saturation in the formula for calculating gas phase adsorption.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

The extended Langmuir isotherm for multi component adsorption is (Arri et al., 1992; Hall et al., 1994):

$$\omega_i = \frac{\omega_{i,max} B_i y_{ig} p}{1 + p \sum_j B_j y_{jg}} \qquad (1)$$

Where

B_i	=	parameter for Langmuir isotherm relation
ϖ_i	=	moles of adsorbed component per unit mass or rock
$\varpi_{i,max}$	=	maximum moles of adsorbed component i per unit mass of rock
p	=	Pressure
y_{ig}	=	molar fraction of adsorbed component i in the gas phase

Where the above sum runs over the adsorbed components. Normally, only the adsorption of light components (e.g. C_1, C_2, N_2, CO_2) is significant.

Both B_i and ϖ_i are parameters of the Langmuir isotherm for Component i.

The product y_{ig}*p represents the partial pressure of component i in the gas phase.

If the keyword *ADS_WATERZONE is specified, then the adsorption for component i, as calculated from the multi component Langmuir model (equation (1) above) is multiplied by the grid block gas saturation, Sg_i:

$$\omega_i = \omega_i * Sg_i \qquad (2)$$

Selection of Model Accounting for Coal Fracture Porosity and Absolute Permeability Changes (Optional) *PALMAN_ARC_COAL, *PALMAN_CMG_COAL, *PALMAN_CMG2_COAL

PURPOSE:

*PALMAN_CMG_COAL, *PALMAN_CMG2_COAL and *PALMAN_ARC_COAL select between the versions of the modified Palmer Mansoori models which account for changes in fracture (cleat) porosity and permeability with pore pressure and/or adsorption / desorption. These models are often needed for Coal Bed Methane (CBM) and Enhanced CBM (ECBM) simulation.

FORMAT:

These keywords take no parameters or qualifiers.

DEFINITIONS:

*PALMAN_CMG_COAL
: The CMG version of the modified Palmer Mansoori model applicable to multi component adsorption is to be used for calculating fracture porosity and absolute permeability changes. The free matrix gas composition is used in the adsorption strain component of the model.

*PALMAN_CMG2_COAL
: The CMG version of the modified Palmer Mansoori model applicable to multi component adsorption is to be used for calculating fracture porosity and absolute permeability changes. The adsorbed gas composition is used in the adsorption strain component of the model.

*PALMAN_ARC_COAL
: The Mavor/ARC version of the modified Palmer Mansoori model applicable to multi component adsorption is to be used for calculating fracture porosity and absolute permeability changes.

DEFAULTS:

*PALMAN_ARC_COAL is the default.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. They only have an effect if *COAL-DIF-*TIME or *COAL-DIF-COMP have been specified in the Rock-Fluid section (CBM/ECBM modeling enabled).

EXPLANATION:

For a detailed discussion of Palmer and Mansoori model and the variant proposed by Mavor applicable to multi component adsorption/desorption including ECBM please refer to manual pages under the Reservoir Description section.

Trace Component (Optional) *TRACE-COMP

PURPOSE:

This keyword defines a trace component that prevents the fluid phase (gas or oil) from completely disappearing in the water zone when gas solubility is activated

FORMAT:

*TRACE-COMP itc

DEFINITION:

itc

Index of the gaseous component to be considered as the trace component. The value must be between 1 and nc where nc is the number of components in the gas (or oil) phase.

EXPLANATION:

This keyword allows the identification of a trace component that is not soluble in the aqueous phase and that does not flow. The role of this component is to prevent the gas phase from completely disappearing. It must be followed by an integer corresponding to the component number. For example, if the fluid system has 2 components, with CO_2 being the 1st component and C_1 being the 2nd component, *TRACE-COMP 2 indicates that the 2nd component is a trace component. For CO_2 sequestration in an aquifer, the trace-component is usually a gaseous component such as C_1 or another CO_2-like component with properties identical to those of CO_2. The Henry's constant of the trace component must be zero (not soluble in the aqueous phase). The aquifer is initialized with a small gas saturation (e.g. 1.0E-04) containing a mixture of CO_2 and this trace-component such that the initial CO_2 aqueous mole fraction is equal to the measured value.

EOS Cubic Root Selection (Optional) *CUBIC-ROOT-SELECTION

PURPOSE:

This keyword provides the user the option of selecting the compressibility root of the EOS when there are three real roots.

FORMAT:

*CUBIC-ROOT-SELECTION	*GAS
	*LIQUID
	*MIN-GIBBS

DEFINITIONS:

*GAS
: The largest root is selected.

*LIQUID
: The smallest root is selected.

*MIN-GIBBS
: The root that corresponds to the lowest Gibbs energy for the phase is selected.

DEFAULT:

*MIN-GIBBS

EXPLANATION:

This option may be required when the system approaches the critical point of one of the dominant components, e.g. CO_2 in the case of CO_2 sequestration in an aquifer. The selection of the root corresponding to the lowest Gibbs energy may give rise to an alternation of the root selection between the smallest root and the largest root and consequently may cause convergence problems.

H2O Vaporization (Optional) *H2O_INCLUDED, *OGW_FLASH, *SWR-H2OVAP, *SATWCUTOFF

PURPOSE:

These keywords are used to model H_2O vaporization.

FORMAT:

*H2O_INCLUDED	
*OGW_FLASH	*NO_H2OVAP
	*ON
*SWR-H2OVAP	swr_h2ovap
*SATWCUTOFF	satwcutoff

DEFINITIONS:

*H2O_INCLUDED

This keyword must be placed after the *NC keyword to indicate that the fluid component set includes H_2O as the last component. The properties of H_2O must be specified in the data set like other components such as CO_2, C_1, etc.

*OGW_FLASH (*NO_H2OVAP) (*ON)

*NO_H2OVAP: H_2O vaporization is not activated. This is equivalent to *FLASH-METHOD-OGW.
*ON: H_2O vaporization is activated.

swr_h2ovap

Residual aqueous phase saturation for H_2O vaporization. If the aqueous phase saturation drops below swr_h2ovap, the vaporization ceases. The default value is 0.

satwcutoff

Aqueous phase saturation cutoff below which all geochemical reactions stop.

GHG Option - Aqueous Components *NC-AQUEOUS, *COMPNAME-AQUEOUS, *MW-AQUEOUS, *ION-SIZE-AQUEOUS, *CHARGE-AQUEOUS

PURPOSE:

These keywords are used to define the aqueous components.

FORMAT:

*NC-AQUEOUS	nc_aqu
*COMPNAME-AQUEOUS	cname_aqu
*MW-AQUEOUS	mw_aqu
*ION-SIZE-AQUEOUS	ion_size_aqu
*CHARGE-AQUEOUS	charge_aqu

DEFINITIONS:

*NC-AQUEOUS nc_aqu
Number of aqueous components excluding the gaseous solutes.

*COMPNAME-AQUEOUS cname_aqu
A total of nc_aqu alphanumeric character strings to identify the names of the nc_aqu aqueous components. One character string is required for each components. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes. *The primary aqueous components must be ordered first, followed by the secondary aqueous components. The number of secondary aqueous components is equal to the number of chemical equilibrium reactions.* This restriction is relaxed when *GEOCHEM_V2 is used.

*MW-AQUEOUS mw_aqu
Molecular weights of aqueous components (nc_aqu values).

*ION-SIZE-AQUEOUS ion_size_aqu
Ion size of aqueous species in Å; used in Debye-Hückel or B-dot activity model (nc_aqu values).

*CHARGE-AQUEOUS charge_aqu
Electrical charge of aqueous species; used in Debye-Hückel or B-dot activity model (nc_aqu values).

GHG Option - Mineral Components *NC-MINERAL, *COMPNAME-MINERAL, *MW-MINERAL, *MASSDENSITY-MINERAL

PURPOSE:

These keywords are used to define the mineral components.

FORMAT:

*NC-MINERAL	nc_mnr
*COMPNAME-MINERAL	cname_mnr
*MW-MINERAL	mw_mnr
*MASSDENSITY-MINERAL	mass_den_mnr

DEFINITIONS:

*NC-MINERAL nc_mnr
Number of mineral components.

*COMPNAME-MINERAL cname_mnr
A total of nc_mnr alphanumeric character strings to identify the names of the nc_mnr mineral components. One character string is required for each component. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes.

*MW-MINERAL mw_mnr
Molecular weights for mineral components (nc_mnr values).

*MASSDENSITY-MINERAL mass_den_mnr
Mass densities of mineral components in kg/m^3 (nc_mnr values).

GHG Option – Reactions *N-CHEM-EQUIL, *N-RATE-REACT, *STOICHIOMETRY,*CONCENTRATION-ORDER, '*GEOCHEM-V2, *REACTION-CHEM, *REACTION-RATE

PURPOSE:

These keywords are used to define the geochemical reactions.

FORMAT:

*N-CHEM-EQUIL	*n_chem_equil*
*N-RATE-REACT	*nrate_rate_react*
*STOICHIOMETRY	*stoich_array*
*CONCENTRATION-ORDER	*conc_order*
*GEOCHEM_V2	
*REACTION-CHEM	*chem_equilibrium_reaction*
*REACTION-RATE	*rate_reaction*

DEFINITIONS:

n_chem_equil

Number of chemical equilibirum reactions in the aqueous phase.

n_rate_react

Number of rate-dependent dissolution and precipitation reactions. Normally, *n_rate_react* = *nc_mnr* (number of minerals) since each mineral has its own dissolution and precipitation rate.

stoich_array

Stoichiometry of the chemical-equilibirum or rate dependent reactions. The reactions occur in the aqueous phase. The total nct components are ordered as follow: nc gaseous solutes, H_2O component, nc_aqu aqueous components, nc_mnr mineral components where

nct = nc + 1 + nc_aqu + nc_mnr.

For models with water vaporization (i.e., with keyword *H2O_INCLUDED), nc+1 is replaced by nc, since nc in that case also includes H_2O.

The stoichiometric coefficient is negative for reactants and positive for products. Reactions involving minerals must be written as dissolution reactions, i.e. minerals are reactants. Enter zeros for components that do not react. Note that a zero must be entered for each gaseous component that is <u>not</u> soluble in the aqueous phase.

conc_order

The concentration order is equal to the stoichiometry, except for the component H_2O where the value is zero. nct values are required. These values are optional for chemical equilibrium reactions and are not required for rate-dependent mineral dissolution/precipitation reactions.

*GEOCHEM_V2

This keyword indicates that certain GHG input data are to be entered in a simpler format. The input of *STOICHIOMETRY, *N-CHEM-EQUIL , *N-RATE-REACT , *ANNIH-MATRIX, and *CONCNTRATION-ORDER is not required when this keyword is used. The chemical equilibrium and mineral reactions are entered in a different format via keywords *REACTION-CHEM and *REACTION-RATE in a syntax discussed below.

Similarly for ion-exchange reactions *REACTION-IEX is used instead of *STOICH-IEX when *GEOCHEM_V2 is in effect.

chem_equilibrium_reaction

The chemical equilibrium reaction string following keyword *REACTION-CHEM from which the stoichiometry of chemical equilibrium reaction is to be obtained.

For instance, *REACTION-CHEM 'CO2' + 'H2O' = 'H+' + 'HCO3-'.

See explanation for syntax details.

rate_reaction

The mineral-dissolution and precipitaion reaction string following keyword *REACTION-RATE from which the stoichiometry of rate equilibrium reaction is to be obtained.

For instance, *REACTION-RATE 'Calcite' + 'H+' = 'Ca++' + 'HCO3-'

See explanation for syntax details.

DEFAULTS:

Required keywords when chemical equilibrium or mineral dissolution and precipitaion reactions are present. See conditions below.

CONDITIONS:

These keywords, if present, must appear in the COMPONENT PROPERTIES section.

Keywords *N-CHEM-EQUIL *N-RATE-REACT and *STOICHIOMETRY are required to specify number of chemical equilibrium, number of mineral precipitation and dissolution reactions, and stoichiometry of these reactions. These keywords and *CONCENTRATION-ORDER must not be used in presence of *GEOCHEM_V2.

Keywords *REACTION-CHEM and *REACTION-RATE are required to define stoichiometry of chemical equilibrium and mineral dissolution and precipitation reactions when keyword *GEOCHEM_V2 is present. These keywords must not be used in absence of *GEOCHEM_V2.

EXPLANATION:

*REACTION-CHEM or *REACTION-RATE keywords must be followed by chemical equilibrium or mineral dissolution and precipitation reaction in one of the following formats.

```
'CO2'  +  'H2O'  =  2  'H+' +  'CO3--'
'CO2' + 'H2O' = 2 'H+'  + 'CO3--'
'CO2' + 'H2O' = 2.0 'H+' + 'CO3--'
'CO2' + 'H2O' = 0.2D+01 'H+' + 'CO3--'
'CO2' + 'H2O' = 0.2E+01 'H+' + 'CO3--'
```

The component names must be quoted. At least one blank space is mandatory between a component, a sign and a number. One equal (=) sign is required for every reaction. For mineral precipitation/dissolution, the mineral component must to be entered on the reactant side. *REACTION-RATE and *REACTION-CHEM are defined after the component definitions.

Keyword *GEOCHEM_V2 orders the stoichiometry in the canonical form and creates equilibrium rate annihilation matrix (ERA) internally, as such input of the annihilation matrix (via *ANNIH-MATRIX) is not required. This provides flexibility of easily adding or removing a reaction from a given GEM geochemistry data input.

Keyword *OUTREACT in the INPUT-OUTPUT section optionally echoes out the reactions to the output file wherein the mass balance error in the stoichiometry is also printed if it is larger than 0.01%. The output can also be invoked via keyword *OUTPRN *RES *ALL in the INPUT-OUTPUT section.

GHG Option – Reactions *DER-CHEM-EQUIL, *DER-REACT-RATE, *YAQU-RATE-CUTOFF

PURPOSE:

These keywords are used to define derivative types and mole fraction cut-off for the geochemical reactions.

FORMAT:

*DER-CHEM-EQUIL	*ANALYTICAL | *NUMERICAL
*DER-REACT-RATE	*ANALYTICAL | *NUMERICAL
*YAQU-RATE-CUTOFF	*yaq_cutoff*

DEFINITIONS:

*DER-CHEM-EQUIL

Method for calculating the derivatives of the chemical equilibrium equation. It must be followed by *NUMERICAL or *ANALYTICAL for numerical or analytical derivatives respectively.

*DER-REACT-RATE

Method for calculating the derivatives of the mineral dissolution or precipitation rates. It must be followed by *NUMERICAL or *ANALYTICAL for numerical or analytical derivatives respectively.

*YAQU-RATE-CUTOFF *yaq_cutoff*

Mineral dissolution/precipitation rate are set to zero if the mole fraction of the gaseous component in the aqueous phase is less than *yaq_cutoff*. nc values are required.

GHG Option – Reaction Equilibrium and Rate Parameters

***LOG-CHEM-EQUIL-CONST, *LOG-CHEM-EQUIL-COEFS, *REACTIVE-SURFACE-AREA, *LOG-TST-RATE-CONSTANT, *REF-TEMP-RATE-CONST, *ACTIVATION-ENERGY, *LOG-TST-RATE-CONSTANT-ACTIV, *TST-ACTIVITY-POWER, *TST-PARAM1, *TST-PARAM2, *SPEC-REACT-SURFACE-AREA, *MIN-REACT-SURFACE-AREA, *SUPER-SATURATION-INDEX**

PURPOSE:

These keywords are used to define the parameters used for the aqueous phase chemical equilibrium reactions and the mineral dissolution and precipitation reactions.

FORMAT:

*LOG-CHEM-EQUIL-CONST	keq
*LOG-CHEM-EQUIL-COEFS	a0 a1 a2 a3 a4
*REACTIVE-SURFACE-AREA	area0
*SPEC-REACT-SURFACE-AREA	area_s
*MIN-REACT-SURFACE-AREA	area_min
*SUPER-SATURATION-INDEX	ssi
*LOG-TST-RATE-CONSTANT	k0_ref
*REF-TEMP-RATE-CONST	t_ref
*ACTIVATION-ENERGY	e_act
*LOG-TST-RATE-CONSTANT-ACTIV	ki
*TST-ACTIVITY-POWER	omega_i
*TST-PARAM1	tst_1
*TST-PARAM2	tst_2

DEFINITIONS:

*LOG-CHEM-EQUIL-CONST keq

log (K_{eq}), log (base 10) of chemical equilibrium constant for aqueous phase chemical equilibrium reactions or rate-dependent mineral dissolution and precipitation reactions.

*LOG-CHEM-EQUIL-COEFS a0 a1 a2 a3 a4

The log(K_{eq}) is calculated with a fourth order polynomial in T, i.e. $\log(K_{eq}) = a_0 + a_1T + a_2T^2 + a_3T^3 + a_4T^4$, where T is in °C. It must be followed by 5 values a_i, i = 0,1,…,4. *If this keyword is used, *LOG-CHEM-EQUIL-CONST is not required.*

*REACTIVE-SURFACE-AREA area0

$\hat{A}_0$, initial reactive surface area in m^2 / m^3 of bulk volume of rock.

*SPEC-REACT-SURFACE-AREA area_s

$\hat{A}_S$, initial specific reactive surface area in m^2 / m^3 of bulk volume of mineral.

area0 = area_s*(volume fraction of mineral)

*MIN-REACT-SURFACE-AREA area_min

$\hat{A}_{MIN}$, minimum specific reactive surface area in m^2 / m^3 of bulk volume of mineral.

This allows mineral precipitation even if the mineral is not present initially.

*SUPER-SATURATION-INDEX ssi

Super-saturation index (S_S) can be used to delay precipitation of minerals. Precipitation or dissolution is controlled by the saturation index (S). S < 1 gives dissolution and S > 1 gives precipitation. If a value of S_S is entered (S_S > 1) then there is no precipitation for 1 <= S <= S_S. [dimensionless]

This allows mineral precipitation even if the mineral is not present initially.

*LOG-TST-RATE-CONSTANT k0_ref

log k_0 [mol/(m^2 s)], standard TST rate constant.

*REF-TEMP-RATE-CONST t_ref

t^* (°C), reference temperature for rate constant. Optional, default = 25 °C.

*ACTIVATION-ENERGY e_act

E_a, activation energy in J/mol. Optional, default = 0.

*LOG-TST-RATE-CONSTANT-ACTIV ki

log k_i [mol/(m^2 s)], (nc + 1 + nc_aqu) values. Optional, default = 0.

*TST-ACTIVITY-POWER omega_i

ωi, (nc + 1 + nc_aqu) values. Optional, default = 0.

*TST-PARAM1 tst_1

ξ, power for Q/K_{eq}. Optional, default = 1.

*TST-PARAM2 tst_2

ζ, power for $| 1 - (Q/K_{eq})^{\xi} |$. Optional, default = 1.

*LOG-TST-RATE-CONSTANT

log k_0 [mol/(m^2 s)], standard TST rate constant.

EXPLANATION:

The aqueous phase chemical equilibrium reactions are governed by the following equation:

$$Q = K_{eq}$$

where Q is the activity product given by

$$Q = \prod_{i=1}^{nct} a_k^{\nu_i}$$

and K_{eq} is the chemical equilibrium constant.

The power (ν_i) used for calculating the activity product is set equal to the stoichiometric coefficient. The power for the component H_2O is taken as zero.

For aqueous phase chemical equilibrium reactions, the only data required is LOG-CHEM-EQUIL-CONST or LOG-CHEM-EQUIL-COEFS.

The mineral dissolution and precipitation reaction rate is calculated from Transition State Theory (TST). The equation for the rate is (Section 4 of Reference 1):

with

$$\hat{A} = \hat{A}_0 \cdot \frac{N_m}{N_{m0}} .$$

Here $\hat{A}$ and $\hat{A}_0$ are the reactive surface area at the current time and at time zero respectively. N_m and N_{m0} are respectively the current moles of mineral and the moles of mineral at time zero.

The rate constant k_0 at the current temperature, T (K), can be calculated from the rate constant k^* at a different temperature, T^* (K), with the following equation:

$$k_0 = k_0^* \cdot \exp\left[-\frac{E_a}{R}\left(\frac{1}{T} - \frac{1}{T^*}\right)\right]$$

where E_a is the activation energy [J/mol] and R is the gas constant (8.314 J/ mol K).

For most cases, only k_0 is used and all other k_i's are zero. As the reaction occurs in the aqueous phase, the aqueous phase saturation, S_w, is used as a factor in the calculation of the rate.

GHG Option - Equilibrium-Rate-Annihilation (ERA) Matrix

***ANNIH-MATRIX**

PURPOSE:

These keywords are used in the equilibrium rate annihilation (ERA) matrix option in GHG.

FORMAT:

*ANNIH-MATRIX

DEFINITIONS:

*ANNIH-MATRIX
: Equilibrium Rate Annihilation (ERA) matrix. This matrix has nca rows and (nca + ncheq) columns.

EXPLANATION:

The ERA matrix **E** is used to reduce the number of flow equations to be solved simultaneously because of the constraints associated with the chemical equilibrium reactions. Please refer to paper SPE 89474 for more details. The matrix **E** is:

$$E = \left\lfloor I_{nca} \quad \left(-\tilde{V}\right)_{nca \times ncheq} \right\rfloor$$

where $\mathbf{I}_{nca}$ is the nca identity matrix (nca = nc + 1 + nc_aqu - ncheq) and ncheq is the number of chemical-equilibrium reactions. The matrix $\tilde{V}$ contains stoichiometric coefficients for the gaseous components, H_2O and primary aqueous components in the chemical-equilibrium reactions.

ERA matrix is internally calculated with *GEOCHEM_V2, as such *ANNIH-MATRIX and associated input must not be used with *GEOCHEM_V2.

GHG Option – Setting of Initial Mineral Equilibrium Reaction Rates and Aqueous Molalities *EQUIL-REACT-RATE, *CHEM-EQUIL-SET

PURPOSE:

These keywords are used to set initial equilibrium reaction rates and aqueous molalities in GHG models.

FORMAT:

*EQUIL-REACT-RATE *ON | *OFF
*CHEM-EQUIL-SET *ON | *OFF

DEFINITIONS:

*EQUIL-REACT-RATE

Mineral dissolution/precipitation rates are normally not zero for the aqueous concentration entered. This keyword with the secondary keyword *ON will calculate a correction factor to the activity product so that the initial reaction rates for all minerals are zero. The secondary keyword *OFF will turn this option off.

*CHEM-EQUIL-SET

The initial molality of the aqueous components are entered with the keyword *MOLALITY-AQUEOUS and *MOLALITY-AQUEOUS-PRIMARY. With the keyword *MOLALITY-AQUEOUS, the molality of all aqueous components are entered, while with the keyword *MOLALITY-AQUEOUS-PRIMARY, only the molality of the primary aqueous components are entered. The molalities of the secondary components are calculated from the chemical-equilibrium-equations. The use of the keyword *MOLALITY-AQUEOUS was not recommended since chemical equilibrium is normally not satisfied. Actually, even with *MOLALITY-AQUEOUS, the molality of the secondary aqueous components are always calculated unless the following keywords are used in Initial Conditions Section *CALC-MOLALITYAQ-SECONDARY *OFF. By default, the secondary keyword for *CALC-MOLALITYAQ-SECONDARY is *ON.

With the 2013.12 release, the use of *CHEM-SET-EQUIL *ON will calculate a correction factor to the activity product so that chemical equilibrium will be honored even if molalities of all nc_aqu components are entered with the keyword *MOLALITY-AQUEOUS.

DEFAULTS:

Default for *EQUIL-REACT-RATE is *OFF.

Default for *CHEM-EQUIL-SET is *OFF.

CONDITIONS:

These keywords if present, must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords.

EXPLANATION:

***EQUIL-REACT-RATE for Mineral Reactions**

The rate law for the mineral dissolution and precipitation reaction is (Bethke, 1996):

$$r_\beta = \hat{A}_\beta \, k_\beta \left(1 - \frac{Q_\beta}{K_{eq,\beta}}\right), \quad \beta = 1,\ldots,R_{mn} \qquad (1)$$

where r_β is the rate, $\hat{A}_\beta$ is the reactive surface area for mineral β, k_β is the rate constant of mineral reaction β, $K_{eq,\beta}$ is the chemical equilibrium constant for mineral reaction β and Q_β is the activity product of mineral reaction β. The activity product Q_β is analogous to the activity product for aqueous chemical equilibrium reactions, i.e.

$$Q_\beta = \prod_{k=1}^{n_{aq}} a_k^{\nu_{k\beta}} \qquad (2)$$

Normally from input data the reaction rate r_β as calculated from Eq. (1) may not be zero at time = 0. To make sure that the reaction rate is zero a factor is computed at the initialization (time = 0.) which is stored for each grid block and each reaction, i.e.,

$$f = Q_\beta / K_{eq,\beta}, \quad \beta = 1,\ldots,R_{mn} \qquad (3)$$

At subsequent time steps, this factor is applied to balance the RHS of Eq. (1) with the mineral reaction rate. The action is taken when * EQUIL-REACT-RATE *ON keyword is specified by the user.

***CHEM-EQUIL-SET for Chemical Equilibrium Reactions**

The chemical equilibrium reactions are modeled with chemical equilibrium constants (Bethke, 1996). The governing equations for the chemical equilibrium reactions are:

$$Q_\alpha - K_{eq,\alpha} = 0, \quad \alpha = 1,\ldots,R_{aq} \qquad (4)$$

with

$$Q_\alpha = \prod_{k=1}^{n_{aq}} a_k^{\nu_{k\alpha}} \qquad (5)$$

where $K_{eq,\alpha}$ is the chemical equilibrium constant for the aqueous reaction α, a_k is the activity of component k, $\nu_{k,\alpha}$ are the stoichiometry coefficients and Q_α is the activity product.

Normally from the input data RHS of Eq. (4) may not be zero. In that case a factor is computed at the initialization which is stored for each grid block and each reaction, i.e.,

$$f = Q_\alpha / K_{eq,\alpha}, \quad \alpha = 1,\ldots,R_{aq} \qquad (6)$$

At subsequent time steps, this factor is applied to balance the activity product with the chemical equilibrium constants. The action is taken when *CHEM-EQUIL-SET *ON keyword is specified by the user. In such a case *YAQU-CUTOFF keyword should not be used.
See Appendix E for further details regarding treatment of chemical equilibrium reactions and mineral rate reactions in GEM.

GHG Option - Aqueous Phase Model (Optional)

***ACTIVITY-MODEL, *ACTIVCOEF-SOL, *ACTIVCOEF, *SALINITY-CALC, *AQFILL**

PURPOSE:

These keywords are used in the aqueous phase model option in GHG. Other aqueous phase model options are described in the "Aqueous Phase Properties" entry in this section of the User Guide.

FORMAT:

*ACTIVITY-MODEL	*IDEAL | *DEBYE-HUCKEL | *B-DOT
*ACTIVCOEF-SOL	*ON | *OFF
*ACTIVCOEF	
*SALINITY-CALC	*ON | *OFF
*AQFILL	*ON | *OFF

DEFINITIONS:

*ACTIVITY-MODEL

Activity model for the aqueous phase. It must be followed by either *IDEAL, *DEBYE-HUCKEL or B-DOT. For *IDEAL, the component activity coefficients are equal to unity. For the other cases, the activity coefficients are calculated from the Debye-Hückel or the B-dot model. By default, *IDEAL is used.

*ACTIVCOEF-SOL

In GEM versions prior to 2013.12, the activity coefficients of the soluble components such as CO2(aq) are set to 1. Thus the activities of soluble components are equal to their molalities. With the 2013.12 version, the use of *ACTIVCOEF-SOL *ON will set the activity coefficients of soluble components to be equal to the salting out coefficients. The default for *ACTIVCOEF-SOL is *OFF, which is consistent with previous versions of GEM.

*ACTIVCOEF

This keyword allows the input of activity coefficients for all components in the aqueous phase, the nc oil/gas components, H_2O and the nc_aqu aqueous components. Thus nc + 1 + nc_aqu real values are required. For components that are not soluble in the aqueous phase, enter 1. These activity coefficients will be used instead to the Debye-Huckel or the B-dot model. Note that the activity of all components except H_2O is equal to the product of the activity coefficient and the molality. The activity of H_2O is equal to the product of the activity coefficient and the mole-fraction of H_2O.

*SALINITY-CALC

Calculation of salinity. It must be followed by *ON or *OFF. If *OFF is entered, the salinity of the aqueous phase is assumed constant and could be specified with the keyword *SALINITY. If *ON is entered, the salinity is by

default calculated from the concentration of Cl^- in the aqueous phase. If Cl^- is not defined, Na^+ will be used. If salinity components are specified using *COMPNAME-SAL, conentration of salinity components will be used instead in salinity calculation.

*AQFILL

This keyword specifies that the aquifer is completely filled with water. It must be followed by *ON or *OFF. If this keyword is not used or *OFF is entered, a small trace of a hydrocarbon phase is always present in the aquifer. With *ON, the saturation of the aqueous phase is equal to unity. *If this keyword is used, the maximum number of gaseous component soluble in the aqueous phase is* 1.

GHG Option – Factors for CO2 Inventory (Optional)

***CO2-AQU-FACT, *CO2-MNR-FACT**

PURPOSE:

These keywords are used to input factors for the CO_2 inventory calculations in aqueous and mineral components.

FORMAT:

*CO2-AQU-FACT
*CO2-MNR-FACT

DEFINITIONS:

*CO2-AQU-FACT

This keyword is followed by nc_aqu values where nc_aqu is the number of aqueous components. These numbers represent the number of molecules of CO_2 in each of the aqueous components. For example, the factor corresponding to HCO_3^- is 1.

*CO2-MNR-FACT

This keyword is followed by nc_mnr values where nc_mnr is the number of minerals. These numbers represent the number of molecules of CO_2 in each of the minerals components. For example, the factor correspoding to Calcite ($CaCO_3$) is 1 whereas the number corresponding to Dolomite [$CaMg(CO_3)_2$] is 2.

DEFAULTS:

Optional keywords. Default for the aqueous and mineral component factors is zero.

CONDITIONS:

This keyword may appear in the COMPONENT PROPERTIES.

EXPLANATIONS:

These keywords are useful only when *INVENTORY-CO2 or any one or more of the *OUTSRF *WELL keywords *GHGGAS *GHGLIQ *GHGTHY *GHGSCRIT *GHGSOL *GHGAQU *GHGMNR have been specified in the INPUT/OUTPUT control section.

Polymer Degradation (Conditional) *POLY-DEGR-FREQFAC, *POLY-DEGR-EACT

PURPOSE:

Specify frequency factor and activation energy for the degradation of a polymer.

FORMAT:

*POLY-DEGR-FREQFAC	'comp_name'	freq_fac
*POLY-DEGR-EACT	'comp_name'	eact

DEFINITIONS:

comp_name
: Quoted name of the polymer component for which the data is entered.

freq_fac
: Frequency factor for an Arrhenius-type reaction rate constant expression for degradation of polymer

eact
: Activation energy for an Arrhenius-type reaction rate constant expression for degradation of polymer

DEFAULTS:

No defaults. If these keywords are absent, polymer will not degrade using the GEM polymer degradation model. If *POLY-DEGR-FREQFAC is specified but *POLY-DEGR-EACT is not, *POLY-DEGR-EACT is set to 0.0.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group. Used only when aqueous phase contains polymer components. *POLY-DEGR-FREQFAC is required to trigger this option.

EXPLANATION:

Polymer degradation can occur due to the mechanical, biological, thermal, and chemical mechanisms. The current version of GEM provides a simplified approach to model polymer degradation. It is useful when the information on the products of degradation is not available or when the main interest is in the rate at which the polymer will degrade and in turn affect the aqueous phase properties rather than in the products formed from degradation.

The degradation model assumes an elementary pseudo-reaction in which polymer degrades to form water:

$$\text{Polymer} \rightarrow \left(\frac{MW_p}{MW_{H_2O}} \right) H_2O$$

where MW_x is molecular weight of x (polymer or H_2O). In this model, the mass balance is maintained although the atom-balance is not for reasons explained above.

The rate expression for the reaction is:

$$r_p = -k_o \cdot e^{-E/RT} \cdot C_p$$

Where

r_p	is the rate of polymer degradation
k_o	is the frequency factor of the reaction (freqfac)
E	is the activation energy of the reaction (eact)
C_p	is the concentration of the polymer in the aqueous phase

The amount of water produced due to this pseudo-reaction will result in a mole balance error. However, the error is expected to be negligible since the concentration of the polymer injected is expected to be very small (typically less than 1200 ppm or 0.12 wt%). In practice, the selected polymer should exhibit low degradation.

GEM offers other keywords to specify a reaction in general. The keywords *POLY-DEGR-FREQFAC and * POLY-DEGR-EACT completely describe the polymer degradation model and no other reactions related keyword is required. Use of *STOICHIOMETRY or *REACTION-RATE keyword for polymer degradation modeling is not recommended.

Polymer Computation Options (Optional) *POLY-VISC-SAL, *POLY-VISC-SHEAR , *POLY-DEGR

PURPOSE:

Enable or disable polymer computations for viscosity and degradation.

FORMAT:

* POLY-VISC-SAL	*ON | *OFF
* POLY-VISC-SHEAR	*ON | *OFF
*POLY-DEGR	*ON | *OFF

DEFINITIONS:

*ON

In each case turn the feature specified by the main keyword on. See Explanation below for details of the features.

*OFF

In each case turn the feature specified by the main keyword off. See Explanation below for details of the features.

DEFAULTS:

If the keyword is not specified, the feature is turned on.

If the keyword is specified but *ON or *OFF is not, the feature is turned on.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group

EXPLANATION:

The keywords are used to turn the following features on or off:

*POLY-VISC-SAL	Effect of salinity on polymer viscosity using *VSSALTCMP
*POLY-VISC-SHEAR	Effect of shear on polymer viscosity using *SHEARTHIN, *SHEARTHICK, *SHEARTAB
*POLY-DEGR	Polymer degradation model using *POLY-DEGR-FREQFAC and *POLY-DEGR-EACT

If a feature is turned on but the required data are not specified, computations for the feature will not proceed.

If the data required for a feature are specified but the feature is turned off, computations feature will not proceed.

Ion-Exchange – Component Definitions *NC-IEX, *COMPNAME-IEX, *AQIONS-IEX, *REACTION-IEX, *STIOCH-IEX, *SCOEFF-IEX *LOG-SCOEFF-IEX

PURPOSE:

These keywords are used to define ion-exchange related input.

FORMAT:

*NC-IEX	*nc_iex*
*COMPNAME-IEX	*cname_iex*
*AQIONS-IEX	*cname_aqionx_iex*
*REACTION-IEX	*iex_reaction*
*STOICH-IEX	
*SCOEFF-IEX	
*LOG-SCOEFF-IEX	

DEFINITIONS:

*NC-IEX *nc_iex*

Number of ion-exchanger components.

*COMPNAME-IEX *cname_iex*

Name of ion-exchanger (*nc_iex* values).

A total of nc_iex alphanumeric quoted character strings to identify the names of the nc_iex ion-exchanger components. One character string is required for each component. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes.

For example 'Na-X', 'Ca-X2', 'Mg-X2'

*AQIONS-IEX *cname_aqionx_iex*

Name of the aqueous ions that are associated with the ion exchanger (nc_iex values).

A total nc_iex alphanumeric character strings to identify the aqueous ions is required. These strings must have been defined with the *COMPNAME-AQUEOUS keyword. For example if the exchangers are 'Na-X', 'Ca-X2' and 'Mg-X2' the following strings are required: 'Na+', 'Ca++' and 'Mg++'.

*REACTION-IEX *iex_reaction*

Ion exchange reaction string for the stoichiometry input for the ion-exchange reactions. The format is the same as *REACTION-CHEM and *REACTION-RATE except only two reactants and two products are allowed. The components have to be declared correctly in *COMPNAME-IEX and *AQIONS-IEX. This keyword must be used after *GEOCHEM_V2. Note that a blank space is required between every component, number and operator. The format is as follows:

*REACTION-IEX ‘Na+’ + 0.5 ‘Ca-X2’ = 0.5 ‘Ca++’ + ‘Na-X’

*STOICH-IEX

Stoichiometry coefficients for the ion-exchange reactions. This keyword must not be used in presence of *GEOCHEM_V2.

The input consists of 4 pairs of data, the name of the aqueous or ion-exchanger component followed by its stoichiometric coefficients (negative for reactants, positive for products). For example for the ion-exchange reaction

$$N_a^+ + \frac{1}{2}CaX_2 \leftrightarrow NaX + \frac{1}{2}C_a^{2+}$$

the input data is

```
'Na+'         -1.0
'Ca-X2'       -0.5
'Ca++'         0.5
'Na-X'         1.0
```

*SCOEFF-IEX

Table of selectivity coefficient functions of temperature for the ion-exchange reactions.

The selectivity coefficient for the ion-exchange reaction

$$N_a^+ + \frac{1}{2}CaX_2 \leftrightarrow NaX + \frac{1}{2}C^{2+}$$

is

$$K'_{Na\backslash Ca} = \frac{\zeta(NaX)\left[m\left(Ca^{2+}\right)\right]^{0.5}}{\left[\zeta(CaX_2)^{0.5}\, m\left(N_a^+\right)\right]} \times \frac{\left[\gamma\left(Ca^{2+}\right)\right]^{0.5}}{\gamma\left(N_a^+\right)}$$

where $\zeta(Na-X)$ and $\zeta(Ca-X_2)$ are the equivalent fraction of N_a^+ and Ca^{2+} on the exchanger respectively, m(.) the molality and γ(.) the activity coefficients. The table entries consist of

Selectivity coefficient at T_1, T_1

…

…

…

Selectivity coefficient at T_n, T_n

…

where T is the temperature in °C.

*LOG-SCOEFF-IEX

Table of $\log_{10}$(selectivity coefficient) functions of temperature for the ion-exchange reactions. Similar input format as that of *SCOEFF-IEX, i.e.,

log10(Selectivity coefficient at T_1), T_1

…

…

…

$\log_{10}$(Selectivity coefficient at T_n), T_n

…

where T is the temperature in °C.

DEFAULTS:

Optional keywords. Default is not to activate ion-exchange.

CONDITIONS:

These keywords, if present, must appear in the COMPONENT PROPERTIES section.

Use either *SCOEFF-IEX or *LOG-SCOEFF-IEX, not both.

EXPLANATIONS:

These keywords are useful in ion-exchange modeling. See Appendix-G for further details. The other relevant keywords are *CEC-IEX and *INTCOMP *EQVFRIEX in ROCK-FLUID DATA, *CONVERGE *CEQIEX in NUMERICAL METHODS CONTROL, and *EQVFRIEX in the INPUT/OUTPUT CONTROL sections.

Polymer Option – Component Definition *COMPNAME-POLYMER

PURPOSE:

This keyword identifies aqueous components in the model that are to be treated as polymer.

FORMAT:

*COMPNAME-POLYMER ‘cname_plm1’ ‘cname_plm2 … …

DEFINITION:

*COMPNAME-POLYMER

A maximum of nc_aqu alphanumeric character strings to identify the names of the polymer components. One character string is required for each component. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes. Here nc_aqu is the number of components in aqueous phase.

DEFAULTS:

Optional keyword. Default is not to activate polymer option in the model.

CONDITIONS:

This keyword, if present, must appear in the COMPONENT PROPERTIES section only after *MODEL, *NC, and *NC-AQUEOUS keywords.

Since polymer components are a subset of aqueous components, it is necessary that *NC-AQUEOUS, *COMPNAME-AQUEOUS and *MW-AQUEOUS appear in the data set.

Keyword *REACTION-CHEM allows geochemical reactions to be used in presence of polymer components, however in such case polymer component must not be defined as secondary (last part of *COMPNAME-AQUEOUS).

EXPLANATIONS:

This keyword identifies the aqueous components that are polymer. It requires at least one and at most nc_aqu quoted character-strings. If *COMPNAME-POLYMER identifies one or more aqueous components as polymer, the polymer-option is invoked in the model and other polymer-option related keywords can be read.

The keyword also serves in determining the number of polymer components in the system. For details about the polymer option in GEM, refer to Appendix-H.

Surfactant – Component Definition *COMPNAME-SURFACTANT

PURPOSE:

This keyword identifies aqueous components in the model that are to be treated as surfactant.

FORMAT:

*COMPNAME-SURFACTANT cname_surf .. ..

DEFINITIONS:

cname_surf .. ..

Name of surfactant component(s) in the aqueous phase

A maximum of nc_aqu alphanumeric character strings to identify the names of the polymer components. One character string is required for each component. A maximum of 8 characters is allowed for each component name. Enclose each string in single quotes. Here nc_aqu is the number of components in aqueous phase.

DEFAULTS:

Optional keyword. If the keyword is absent, none of the aqueous components will be treated as surfactant component.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group, after the *MODEL, *NC, and *NC_AQUEOUS keywords.

No more than the number of aqueous components may be specified as surfactant components.

Since surfactant components are a subset of aqueous components, it is necessary that *NC-AQUEOUS, *COMPNAME-AQUEOUS and *MW-AQUEOUS appear in the data set.

Keyword *REACTION-CHEM allows geochemical reactions to be used in presence of aqueous surfactant components, however in such case surfactant component must not be defined as secondary (last part of *COMPNAME-AQEEOUS).

EXPLANATION:

Surfactant components are required for certain processes, viz. foam, to model their effect on the physical and fluid flow characteristics of the reservoir. These components are assumed to remain in the aqueous phase and are not allowed to partition in any other phase. They flow with the aqueous phase and their concentration in a grid block may be used, for instance, as a basis to alter the relative permeabilities and capillary pressure of the grid-block.

The keyword also serves in determining the number of surfactant components in the system.

Example:

```
*COMPNAME-SURFACTANT 'SURF1'   'SURF2'   'SURF3
```

Resistance Factor Calculation (Optional) *RFCALC, *RF_EXPONENT

PURPOSE:

Specify computation of phase resistance factor caused by change in fluid porosity due to solid deposition and/or mineral precipitation or dissolution.

FORMAT:

*RFCALC	*OFF | *K-C | *POWER | *ADS-RRFT
*RF_EXPONENT	rfexp

DEFINITIONS:

*OFF

No resistance factor is calculated (rf = 1) and no change in porosity due to solid deposition and/or mineral precipitation or dissolution is accounted for.

*K-C

Resistance factor is calculated based on the Kozeny-Carman type formula

rf = $[\phi_n/\phi_k]$**rfexp*$[(1-\phi_k)/(1-\phi_n)]$**2,

where ϕ_n is the fluid porosity at the previous timestep and ϕ_k is fluid porosity at the current timestep.

*POWER

Resistance factor is calculated based on the power law formula

rf = $[\phi_n/\phi_k]$**rfexp.

rfexp

Exponent used in the above formula (dimensionless and defaulted to 3.0).

*ADS-RRFT

Resistance factor is calculated based on the value of residual resistance factor (*RRFT) specified for an adsorbing component in the aqueous phase in the rock-fluid section.

For a total of m adsorbing components in aqueous phase, the resistance factor is calculated as under where k and n represent current; and previous timestep values of porosity and adsorption quantities.

$$rf = \prod_{i}^{m}\left(\frac{A}{B}\right)$$

where,

$$A = 1 + \frac{(rrft(i) - 1)\omega_i^k/\omega_{i_max}}{\phi^k}$$

and,

$$B = 1 + \frac{(\mathrm{rrft(i)} - 1)\omega_i^n / \omega_{i_max}}{\phi^n}$$

ω_i = moles of adsorbed component per unit mass or rock

$\omega_{i,max}$ = maximum moles of adsorbed component i per unit mass of rock

DEFAULTS:

Optional keywords. The default is to use the power law formula for the solid deposition model and no resistance factor calculation for the mineral precipitation or dissolution in the GHG option. No change in porosity due to solid deposition and/or mineral precipitation or dissolution is accounted for when keyword *RFCALC is absent.

CONDITIONS:

These keywords must be in the COMPONENT PROPERTY keyword group, after the *MODEL and *NC keywords. *RFCALC should only be used when the solid model is activated (see *SOLIDMODEL) or when the mineral components are specified in the GHG option (see *NC-MINERAL).

EXPLANATION:

Conceptually permeability varies with fluid porosity due to its changes from solid deposition and/or mineral precipitation or dissolution:

$k_k = k_n / rf$,

where the resistance factor rf is modelled by the Kozeny-Carman equation or the power law relationship, and k_n and k_k refer to permeability at previous and current time steps respectively. The calculation of resistance factor is done recursively from timestep to timestep. Use of this option may over-ride the setting for *DERIVATIVE-METHOD.

It is important to note that change is permeability due to solid deposition; mineral precipitation or dissolution; or due to aqueous phase adsorption is accounted for in flow calculations by the phase resistance factors which can be visualized using *OUTSRF *GRID *RFO *RFG *RFW or corresponding special history keywords in the input/output section. The permeability value itself remains unaffected and therefore user will not see any change in permeability while visualizing.

Keyword *RFCALC was earlier referred to as *PERM-VS-POR, which is still accepted but is deprecated.

Thermal Option (Optional) *THERMAL, *ENTHCOEF

PURPOSE:

These keywords are used in the thermal option.

FORMAT:

```
*THERMAL    *ON
*ENTHCOEF      enth_a(1)  enth_b(1)  enth_c(1)  enth_d(1)  enth_e(1)  enth_f(1)
               enth_a(2)  enth_b(2)  enth_c(2)  enth_d(2)  enth_e(2)  enth_f(2)
               .
               .
               .
               enth_a(nc)  enth_b(nc)  enth_c(nc)  enth_d(nc)  enth_e(nc)  enth_f(nc)
```

DEFINITIONS:

*THERMAL *ON
Turns on thermal option.

*ENTHCOEF
Indicates input of ideal gas enthalpy coefficients.

enth_a enth_b enth_c enth_d enth_e enth_f
Coefficients for ideal enthalpy calculations. 6 coefficients are required per component. These could be generated from running WinProp.

EXPLANATION:

The ideal enthalpy at a given temperature T is calculated from a polynomial expression that takes the following form:

$$H_{ideal} = H_A + H_B * T + H_C * T^2 + H_D * T^3 + H_E * T^4 + H_F * T^5$$

where the temperature T is in °R. The ideal enthalpy coefficients H_A through H_F should be specified in units to give H_{ideal} in Btu/lb.

EOS Enthalpy Calculation Method (Optional) *ENTHALPY-MIXING

PURPOSE:

This keyword provides the user the option of using ideal mixing or non-ideal mixing for EOS enthalpy calculations that are applicable to the gas or oil phases.

FORMAT:

*ENTHALPY-MIXING	*IDEAL
	*NONIDEAL

DEFINITIONS:

*IDEAL
: Ideal mixing is used.

*NONIDEAL
: Nonideal mixing is used.

DEFAULT:

*NONIDEAL

Critical Delta Density Threshold(Optional) *CRITDELD

PURPOSE:

*CRITDELD is used to specify a threshold density difference between gas and liquid densities that is used to determine if the flash calculation should be skipped. This is intended to reduce the number of flash calculations done in the critical region, where the gas and liquid densities are very similar.

FORMAT:

*CRITDELD crdeld

DEFINITIONS:

* CRITDELD
Specifies the input of threshold density difference for flash region skipping

crdeld
Value of threshold density difference (kg/m^3 | lb/ft^3)

DEFAULTS:

Optional keyword. The default is 25 kg/m^3 | 1.56 lb/ft^3.

CONDITIONS:

This keyword must be in the COMPONENT PROPERTY keyword group.

EXPLANATION:

Critical region flash skipping is used to limit the number of flash calculations attempted at conditions where the flash calculation takes a large number of iterations to converge.

Specifying a smaller number for the threshold density difference will result in more flash calculations being done in the critical region. It is possible to turn off critical region flash skipping by specifying a negative number for the threshold density difference. This may be useful to get accurate phase splits near the saturation pressure for near-critical fluids.

Example:

```
*CRITDELD   -1
```

Rock-Fluid Data

Notes on Rock-Fluid Data

1. The default in GEM is to evaluate kro when three phases are present using Stone's second model as normalized by Aziz and Settari, with the krow and krog values which enter the Stone's second model formula being evaluated as functions of Sw and Sg, respectively. There are several other methods for computing Kro which can be invoked. The other models available are Stone's first model, the segregated model and the Linear Isoperm Method. Note that for both Stone's first and second models, krow and krog can be looked up as functions of So rather than Sw and Sg. Please see the entry under the *KROIL keyword for more information. The default calculation of kro in Stone's first model can be modified by user-specified tables of residual oil saturation versus water or gas saturation for drainage and imbibition cases.

   Oil wet systems

   For oil wet systems the intermediate wetting phase is water, not oil, in a three phase gas/oil/water system. Accordingly krwo (from 2 phase water/oil table) and krwg (from 2 phase connate oil/water gas table) that enter into empirical models such as Stone's models are calculated as a function of So and Sg respectively. Note that for oil wet systems there are no provisions for alternative means of looking up the two phase relative permeability, for example krwo as a function of Sw. Therefore the sub keywords SO/SWSG of KROIL should not be used for oil wet systems. For similar reasons do not use the keyword KRGAS since for 2 phase gas/water systems, the oil saturation will be zero and therefore the value to be assigned to krg cannot be looked up from the oil/water table.

2. GEM has the ability to describe phenomena such as multiple rock types and interpolation of curves based on specified variables (e.g. fluid compositions, solid deposition, and temperature), which implies that multiple sets of rock-fluid data may be required. The following gives an overview of the manner in which this data is entered. Here rock type #1 has interpolation between three sets of data which correspond to three values of the specified variable (for example, the solid deposition volume fractions); rock type #2 has no interpolation.

   ROCK TYPE 1 : Rock Type designator (*RPT)
   - Interpolation Definition (*INTCOMP)
   - Interpolation Set #1
     - Interpolation Set (local) designator (*KRINTRP)
     - Interpolation parameter (*INTCOMP_VAL)
     - Relative Permeability/Cap Pressure data (*SWT and *SLT, etc.)

Interpolation Set #2
Interpolation Set (local) designator (*KRINTRP)
Interpolation parameter (*INTCOMP_VAL)
Relative Permeability/Cap Pressure data (*SWT and *SLT, etc.)
Interpolation Set #3
Interpolation Set (local) designator (*KRINTRP)
Interpolation parameter (*INTCOMP_VAL)
Relative Permeability/Cap Pressure data (*SWT and *SLT, etc.)
ROCK TYPE #2: Rock Type designator (*RPT)
Relative Permeability/Cap Pressure data (*SWT and *SLT, etc.)
ASSIGN ROCK TYPES (*RTYPE rock_type_number)

3. For the water wet system: Hysteresis effects are allowed on the oil-water capillary pressures, Pcow = Po-Pw, gas-oil capillary pressures, Pcgo = Pg-Po, gas phase relative permeability, krg and oil phase relative permeability in 2 phase oil/water system, that is krow. Hysteresis is modelled using an approach similar to that of Killough, 1976. To invoke Pcow hysteresis enter both the drainage and imbibition Pcow columns in the *SWT tables and optionally epspc using *EPSPC. To invoke Pcgo hysteresis enter both the drainage and imbibition Pcgo columns in the *SGT/*SLT tables and optionally epspcg using *EPSPCG. To invoke krg hysteresis, enter the keyword *HYSKRG. To invoke krow hysteresis, enter the keyword *HYSKROW.

   For the oil wet system: Hysteresis effects are allowed on the water-oil capillary pressures, Pcwo = Pw-Po, gas-oil capillary pressures, Pcgo=Pg-Po, gas phase relative permeability, krg and water phase relative permeability in 2 phase oil/water system, that is krwo. Hysteresis is modelled using an approach similar to that of Killough, 1976. To invoke Pcwo hysteresis enter both the drainage and imbibition Pcwoi columns in the *SWT tables and optionally epspc using *EPSPC. To invoke Pcgo hysteresis enter both the drainage and imbibition Pcgo columns in the *SGT/*SLT tables and optionally epspcg using *EPSPCG. To invoke krg hysteresis, enter the keyword *HYSKRG. To invoke krwo hysteresis, enter the keyword *HYSKROW.

   Hysteresis (any model) should not be used together with velocity dependent relative permeability options.

   The WAG hysteresis model, (*HYS_MODEL *3PWAG) cannot be used with the interfacial tension dependent relative permeability option.

   The Land (*HYS_MODEL *LAND) and Carlson-Land (*HYS_MODEL *CARLSONANDLAND) hysteresis options can be used together with the interfacial tension dependent option (invoked with the keyword *SIGMA) if the system is specified to be water wet. Hysteresis for the non-wetting phase, gas, can be specified. For the oil phase, hysteresis can be specified for the oil curve in the oil-water system (krow). When the interfacial dependent relative permeability model is used (keyword *SIGMA) the oil and gas phase relative permeability are calculated by interpolating between the immiscible curves and miscible curves. Both immiscible and miscible curves utilize to some extent the user entered relative permeability tables (entered via the keywords *SGT/*SLT/*SWT).

For the immiscible case, gas phase relative permeability is a function of the gas phase saturation and looked up based on the user entered *SGT/*SLT table. For the immiscible case, the oil relative permeability in the water – oil two phase system (krow) is looked up from the entered *SWT table as a function of water saturation and the oil relative permeability in the two phase gas – oil system (krog) is looked up from the entered *SGT/SLT as a function of gas saturation. A final value for the oil phase immiscible curve relative permeability is calculated based on the krow and krog values and a three phase model such as stones model 1 or 2. If the keyword *HYSKRG is specified then hysteresis is applied to the gas phase immiscible/miscible relative permeability curves. Similarly if the *HYSKROW keyword is specified then hysteresis is applied to the krow immiscible/miscible relative permeability curves.

For the miscible case, the oil and gas phase are considered to be fully miscible, the system reduces to a two phase water hydrocarbon system. For the purpose of relative permeability calculations, the hydrocarbon phase can be considered to be either gas or oil. When the hydrocarbon phase is considered to be gas, the relative permeability is looked up from the user entered table (*SGT/*SLT) with the gas saturation equal to one minus the water saturation in the grid block. Hysteresis for this 'gas' phase miscible curve is turned on by specifying the keyword *HYSKRG. If a different value of the maximum trapped gas saturation for the miscible curve is required, then it can be specified with the keyword *HYSKRGM. In this case imbibition corresponds to increasing water saturation and drainage decreasing water saturation. If the hydrocarbon phase is considered to be oil, then the system reduces to two phase oil water. The oil phase relative permeability is looked up from the user entered table *SWT based on the existing water saturation in the grid block. Hysteresis for this 'oil' phase miscible curve can be turned on by specifying the keyword *HYSKROW. If a different value of the maximum trapped oil saturation for the miscible curve is required, then it can be specified with the keyword *HYSKROM. Imbibition corresponds to increasing water saturation and drainage decreasing water saturation. The final miscible relative permeability are constructed for the oil and gas phases by averaging the miscible 'gas'-water and 'oil'-water relative permeabilities to get a hydrocarbon phase relative permeability and then multiplying by the individual gas or oil phase saturation (see the EXPLANATION section of the *SIGMA keyword manual page for more details).

4. In very early versions of GEM all single phase hydrocarbon relative permeabilities were calculated using the oil relative permeability curve, krow(Sw), irrespective of whether the hydrocarbon phase was identified as a gas or an oil phase. This was done because the oil-water relative permeabilities were interpreted as the wetting/non-wetting curves. The *KRGAS keyword allows the user to adhere to this convention or to calculate krg directly from the gas-liquid curves.

   For the oil wet case, since kro is looked up as a function of So, the relative permeability assigned to single phase gas will always be equal to the value of oil relative permeability at connate oil (first entry in the kro column). Do not use the keyword KRGAS for oil wet cases.

5. The trapped oil option in GEM can be used to model the end-pore hydrocarbon saturation. The trapped oil is immobile but can expand and be compressed as pressure changes. The overall effect of the trapped oil is equivalent to reducing the recoverable pore volumes. In GEM, for simplicity, the trapped hydrocarbon is assumed to exist as an oil phase. Use *SORM to specify the trapped oil saturations for each grid block.
6. Interfacial tension effects on relative permeabilities and gas-oil capillary pressures can be modelled in GEM using *SIGMA. As the interfacial tension between the oil and gas phases decreases, the oil and gas relative permeabilities become linear functions of their saturations, and the gas-oil capillary pressure approaches zero.
7. Relative permeability tables may be scaled to different saturation end points for each grid block. This is done by entering the following eight end points for each grid block.

   *BSWCRIT: Water wet: The critical water saturation. Oil wet: The critical oil saturation.

   *BSGCRIT: The critical gas saturation.

   *BSORW: Water wet: The residual oil saturation from the oil relative permeability curve – 2 phase water/oil table. Oil wet: The residual water saturation from the water curve – 2 phase oil/water table.

   *BSORG: Water wet: The residual oil saturation from the liquid (connate water +oil) relative permeability curve – 2 phase liquid/gas table. Oil wet: The residual water saturation from the liquid (connate oil + water) phase relative permeability – 2 phase liquid/gas table.

   *BSWCON: Water wet: The connate water saturation. Oil wet: The connate oil saturation.

   *BSGCON: The connate gas saturation.

   *BSOIRW: Water wet: Irreducible oil saturation in 2 phase oil/water system. Oil wet: The irreducible water saturation in 2 phase oil/water system.

   *BSLCON: The connate liquid saturation in 2 phase gas-liquid system. Water wet: Equal to connate water + irreducible oil from 2 phase liquid/gas table. Oil wet: Equal to the connate oil saturation plus the irreducible water saturation from 2 phase liquid/gas table.
8. Relative permeability tables may also be scaled / shifted to different relative permeability end points for each grid block (vertical scaling). This is done by entering one or more of the following end points for each grid block:

   *BKRWIRO: Water wet: Water rel. perm. at irreducible oil saturation in the oil-water table (the last row in the table). Oil wet: Oil relative permeability at irreducible water in the oil water table (the last row in the table).

   *BKROCW: Water wet: Oil rel. perm. at connate water (oil-water table). Oil wet: The water relative permeability at connate oil (oil-water table)

   *BKRGCL: Gas rel. perm at connate liquid (liquid-gas table)

*BKROGCL: Water wet: Oil rel. perm at connate gas (liquid-gas table). Oil wet: The water relative permeability at connate gas.

*PCWMAX – Water wet: Oil water (Po-Pw) capillary pressure at connate water. Oil wet: The water-oil (Pw-Po) capillary pressure at connate oil

*PCGMAX – Gas oil capillary pressure at connate liquid.

*PCOW_SHIFT – Water wet: Amount that oil water (Po-Pw) capillary pressure is shifted. Oil wet: The amount that water-oil (Pw-Po) capillary pressure is shifted

*JFWMAX – Water wet: Oil water J function at connate liquid. Oil wet: Water oil J function at connate oil

*JFGMAX – Gas oil J function at connate liquid

*JFW_SHIFT – Water wet: Amount that oil water J function is shifted. Oil wet: The amount that water oil J function is shifted

*BKRWRO – Water relative permeability at residual oil (oil-water table)

*BKROCRW – Oil relative permeability at critical water (oil-water table)

*BKRGRL – Gas relative permeability at residual liquid (liq.-gas table)

*BKROGCRG – Oil relative permeability at critical gas (liq.-gas table)

10. Relative permeability tables may be scaled to different saturation and relative permeability end points for each relative permeability table. This is done by entering the corresponding end points for each table, i.e.,

    Saturation:

    *TSWCON, *TSWCRIT, TSOIRW, *TSORW, *TSGCON, *TSGCRIT, *TSLCON, *TSORG, *TSOIRG; and

    Relative Permeability:

    *TKRWIRO, *TKROCW, *TKRGCL, *TKROCG, *TKRWRO, *TKROCRW, *TKRGRL, *TKROGCRG.

    If for a particular grid-block both block-based (say, *BKROW) and table-based (say, *TKROW) end points are assigned, the value specified with block-based (say, *BKROW) end point will be used.

11. Different rock compressibilities can be entered for each rock type using *CROCK.
12. J Function can be input in place of capillary pressure. See the keywords *JFUNC, *SRFTNW and *SRFTNG for details.
13. Modelling oil wet reservoirs: The oil wet option of GEM is turned on by adding the keyword OILWET to the RPT <table number> sequence. The user may assign a particular region to be oil wet and another region to be water wet. Doing so could result in peculiar results and possibly unstable run since the relative permeabilities can change suddenly as fluid moves from an oil wet to water wet grid block or vice versa. For initialization using DEPTH_AVE or BLOCK_CENTER COMP all initialization regions must use either the oil wet or water wet tables. Please refer to the notes under the tutorial section for more extensive discussion on the topic of modeling oil wet reservoirs. For the oil wet case the independent variable for SWT table look up is the

oil saturation. The intermediate phase is now water and not oil. Therefore the first column of the SWT table should be the oil saturation, the second column the oil relative permeability, the third column the water relative permeability. The optional fourth column should be values of the water oil capillary pressure, non-wetting phase pressure minus the wetting phase pressure (Pcwo = Pw-Po). The fourth column corresponds to the capillary pressure measured under drainage conditions. The fifth column also optional specifies values of Pcwoi, water oil imbibition capillary pressure. Therefore for given oil saturation the drainage capillary pressure should always be greater than or equal to the imbibition capillary pressure. For the SGT/SLT table the third column for oil wet is the relative permeability of water in the presence of two mobile phases, gas and water and also in the presence of immobile or connate oil phase. The definitions of other columns of the SGT/SLT table are unchanged from the water wet case.

NOTE: Starting with GEM versions labeled 2007.xx, the default is to use the new relative permeability scaling. This is a change in behavior from the 2005 and 2006 versions which used the old scaling by default. The old scaling method can be selected by specifying *RPT *SCALING-OLD, however, meeting any of the following conditions will trigger the new scaling algorithm:

1. One or more of the following new keywords appear in the data set: *BSWCON, *BSGCON, *BSLCON, *BSOIRW, *BKRWIRO, *BKROCW, *BKRGCL, *BKROGCL, *BKRWRO, *BKRGRL, *BKROCRW, *BKROGCRG, or *PCWMAX, *PCGMAX.
2. *SCALING-NEW is selected for some instance of the *RPT keyword (see the *RPT help section) or
3. *KRPERF keyword appears (see the *KRPERF help section).

Note that *RPT *SCALING-NEW and *RPT *SCALING-OLD cannot be mixed in a single GEM data set.

To reproduce the behavior of data sets that were run on GEM 2006.xx or earlier without SCALING-NEW specified, *RPT *SCALING-OLD must be used.

Start of Rock-Fluid Property Input (Required) *ROCKFLUID

PURPOSE:

*ROCKFLUID indicates the start of the rock-fluid data.

FORMAT:

*ROCKFLUID

DEFAULTS:

Required keyword. No default.

CONDITIONS:

This keyword must be the first keyword in the ROCK- FLUID DATA keyword group. ROCK-FLUID DATA must follow immediately after component properties.

EXPLANATION:

Tables giving krw (Sw), krow (Sw) and Pcow (Sw) and krg(Sg), krog(Sg), and Pcog(Sg) for the water wet case or giving kro(So), krwo(So), Pcwo(So) and krg(Sg), krwg(Sg) and Pcog(Sg) for the oil wet case are entered in this section. These tables are entered under the *RPT keyword.

The method for computing the intermediate phase oil relative permeability may also be specified in this section. For the water wet case the intermediate phase is the oil phase and for the oil wet case the intermediate phase is the water phase. The available methods are Stone's Second Model as modified by Aziz and Settari, which is the default, Stone's First Model as modified by Aziz and Settari, the segregated model and the Linear Isoperm method. The method for computing the 3-phase kro (water wet) or krw (oil wet) is specified under the *KROIL keyword.

The relative permeability of water (water wet case) in a three phase system is the same as that in a two phase water-oil system, and is a function of Sw (water saturation) only. The relative permeability of oil (oil wet case) in a three phase system is the same as that in a two phase water-oil system, and is a function of So (oil saturation) only. The relative permeability of gas in the three phase system is equal to the gas relative permeability in the two phase liquid-gas system, and is a function of Sg (gas saturation) only.

Connate water saturations, critical gas saturations, and residual oil saturations are initialized by scanning the table entries. Hysteresis effects can be included as detailed later in this chapter.

Tabular Relative Permeabilities *RPT

PURPOSE:

*RPT indicates that this set of relative permeability curves will be defined by table entries.

FORMAT:

```
*RPT  set_number        (*DRAINAGE)
                        (*IMBIBITION) (*PCOW | *PCOG | *BOTH)
                        (*SCALING-OLD|*SCALING-NEW)
                        (*OILWET)
```

DEFINITIONS:

set_number

Set_number for this set of water-oil and liquid- gas relative permeability tables. This is the number used with *RTYPE to assign relative permeability curves to grid blocks.

*DRAINAGE, *IMBIBITION

Hysteresis subkeywords for capillary pressure -- if *DRAINAGE is specified the drainage capillary pressure curve is used initially; if *IMBIBITION is specified the imbibition curve is used initially. At most one of these subkeywords may be entered; if neither is entered *DRAINAGE is assumed. Hysteresis effects may also be included for the relative permeabilities, but for relative permeabilities the drainage curve is ALWAYS used initially.

*PCOW

Sub-keyword of the *IMBIBITION keyword. If present, the imbibition oil-water capillary pressure curve and the drainage oil-gas capillary pressure curve are used initially.

*PCOG

Sub-keyword of the *IMBIBITION keyword. If present, the drainage oil-water capillary pressure curve and the imbibition oil-gas capillary pressure curve are used initially.

*BOTH

Sub-keyword of the *IMBIBITION keyword. If present, the imbibition oil-water capillary pressure curve and the imbibition oil-gas capillary pressure curve are used initially. This is the same behavior as when *IMBIBITION is not followed by a sub-keyword

*SCALING-NEW

This new model uses eight end points, while the older end point scaling model used four end points. The new model gives an improved representation of end point scaling. Beginning with version 2007.xx, this is the default method.

*SCALING-OLD

This keyword allows the user to revert back to the older end point scaling option employed in GEM for releases 2004.xx and earlier. While the new model uses eight end points, the old model used four end points.

*OILWET

This keyword indicates that the entries of the SWT and SGT table should be interpreted according to format corresponding to oil wet data requirements.

DEFAULTS:

Required keyword. No default.

If neither *SCALING-OLD nor *SCALING-NEW appears, *SCALING-NEW is assumed. To reproduce the behavior of data sets that were run on GEM 2006.xx or earlier without SCALING-NEW specified, *RPT *SCALING-OLD must be used. If OILWET is missing then water wet reservoir is assumed.

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group. It must be after the *ROCKFLUID keyword before the specification of relative permeabilities and capillary pressure data. One *RPT keyword is necessary for each rock type whose properties are entered in tabular form.

EXPLANATION:

A specified method (normalized Stone's Model 2, normalized Stone's Model 1, Segregated model, or the Linear Isoperm Method) is used to calculate the three phase relative permeabilities from the two-phase curves that are input under *RPT.

There must be both a water-oil relative permeability table and a liquid-gas relative permeability table immediately following this keyword.

Set_number assigns an integer index to the curve set being entered. Each set of curves characterizes one rock type in the reservoir. Later, a rock type is assigned to each grid block in the reservoir. If only one set of relative permeability curves is used, the set_number may be omitted since the default is 1.

If there is more than one rock type, the first relative permeability set must have a set_number of one and set_numbers must increase CONSECUTIVELY for subsequent table sets. Set_numbers are assigned to grid blocks with the *RTYPE keyword.

The *RPT keyword must be immediately followed by a *SWT keyword and table, then a *SLT or *SGT keyword and table.

Interpolation Component (Optional) *INTCOMP, *CRVWGT

PURPOSE:

Define variable for interpolation among multiple relative permeability table sets for a given rock type.

FORMAT:

*INTCOMP	*variable or phase*	(*comp_name*)
*CRVWGT	*crvwgt*	

DEFINITIONS:

variable or phase

Sub-keyword indicating the interpolation variable or phase from which the component's composition will be taken:

*OIL	h-c component mole fraction in oil phase
*GAS	h-c component mole fraction in gas phase
*WATER	h-c component and H_2O mole fraction in aqueous phase
*GLOBAL	global mole fraction
*MAX	maximum of oil, gas and water mole fractions
*AQUEOUS	aqueous phase molality (gmol / kg H_2O)
*FOAM	Mole fraction in aqueous phase of a surfactant component. (comp_name is required and it refers to a valid surfactant component name). A non-zero composition of the surfactant component, in presence of gas phase, can potentially form foam in the grid-block.
*MINERAL	porosity change due to mineral (no *comp_name* needed)
*SOLID	solid deposition volume fraction (no *comp_name* needed)
*TEMPER	temperature (C|F, no *comp_name* needed)
*EQVFRIEX	equivalent fraction of *comp_name* (gmol/m^3 × charge of ion)
*EQVFRIEXN	equivalent fraction of *comp_name* (gmol/m^3 × charge of ion) determined using maximum CEC over all the grid-blocks
*CAPNOW	Oil-water capillary number calculated using o-w interfacial tension specified via *IFTTABLE (i.e., a table of surfactant composition vs. oil-water IFT).

comp_name

Quoted name of component upon whose composition the rock-fluid interpolation will depend. If required, it must match one of the component names specified in the COMPONENT PROPERTY keyword group for the corresponding phase.

crvwgt

Curvature change parameter for the interpolation between two sets of rock-fluid data.

DEFAULTS:

If *INTCOMP is absent from a given rock type, interpolation will NOT be enabled. Linear interpolation will be conducted between two adjacent relativity permeability table sets if *INTCOMP is entered but *CRVWGT is absent (*crvwgt* defaulted to 1).

CONDITIONS:

The keywords must be in the ROCK-FLUID DATA keyword group. They must be placed after each *RPT keyword if there are more than one rock types exist.

EXPLANATION:

Under special circumstances, the assumption that rock-fluid properties are functions only of fluid saturations and saturation histories is not sufficient to accurately describe observed flow behavior, such as the wettability change due to solid deposition or endpoint modification in low salinity. In these cases, the ability to interpolate basic relative permeability and capillary pressure data as functions of other variables can prove very useful. Because of the flexibility in the choice of interpolation variables and the fact that arbitrary tabular data relative permeability and capillary pressure can be employed, a wide variety of phenomena can be handled. Keyword *INTCOMP activates such interpolation among the multiple relative permeability table sets (input via keyword *KRINTRP) for the current rock type under *RPT.

The curvature change parameter *crvwgt* allows additional flexibility in interpolating between sets of curves if experimental evidence requires it. Normally default values of 1 are recommended unless a poor history match is observed. Non-default values imply that the slopes of the interpolated relative permeability curves do not change at the same rate as the endpoint values. The interpolation formula is given as follows:

$$k_r = wgt * k_r(j) + (1-wgt) * k_r(j+1)$$

$$wgt = wgt0 ** crvwgt$$

where the current interpolation value falls between two adjacent rock-fluid datasets j and j+1, and $wgt0 \in [0, 1]$ is the distance weighting parameter. k_r refers to the relative permeabilities and capillary pressure.

Interpolation Set Number and Parameters (Optional)

***KRINTRP, *INTCOMP_VAL, *DTRAPW, *DTRAPN**

PURPOSE:

Indicate interpolation set number along with its value of interpolation parameters.

FORMAT:

*KRINTRP	*nset*	(*COPY *old_rock old_set*)
*INTCOMP_VAL	*intval*	
*DTRAPW	*dtrapw*	
*DTRAPN	*dtrapn*	

DEFINITIONS:

*KRINTRP *nset*

Interpolation set number, local to the current rock-fluid rock type. Values start at 1 for each new rock type and increase by 1 for each additional interpolation set. For example, rock type #1 might have local set numbers 1 and 2 while rock type #2 might have local set numbers 1, 2 and 3.

*COPY *old_rock old_set*

Optional keyword that allows a rock-fluid data set to be defined incrementally from another previously defined set.

old_rock is a rock type number previously defined via *RPT.

old_set is a local interpolation set number previously defined via *KRINTRP.

intval

Value of the interpolation parameter of each interpolation set. Physical meaning of *intval* (including its unit) depends on the interpolation variable definition set by *INTCOMP. Refer to Explanation below for usage.

dtrapw

Value of wetting phase interpolation parameter for current interpolation set of the current rock type. Physical meaning of *dtrapw* (including its unit) depends on the interpolation variable definition set by *INTCOMP. Refer to Explanation below for usage.

dtrapn

Value of non-wetting phase interpolation parameter for current interpolation set of the current rock type. Physical meaning of *dtrapn* (including its unit) depends on the interpolation variable definition set by *INTCOMP. Refer to Explanation below for usage.

DEFAULTS:

For a rock type, if *KRINTRP is absent or there is only one interpolation set defined then there is no rock-fluid interpolation, but for *COPY purposes the data is accessible as

interpolation set #1. If *KRINTRP is specified, then the value of interpolation parameter must be supplied.

If *DTRAPW is specified but *DTRAPN is not, value of *DTRAPN is set equal to that of *DTRAW. If *DTRAPN is specified but *DTRAPW is not, value of *DTRAPW is set equal to that of *DTRAPN.

CONDITIONS:

The keywords must be in the ROCK-FLUID DATA keyword group. They must be placed after a *RPT or *INTCOMP keyword. *KRINTRP must be placed after a *RPT or *INTCOMP keyword. *INTCOMP_VAL, *DTRAPW or *DTRAPN must be placed after a *KRINTRP keyword.

EXPLANATION:

At least two sets of rock-fluid data must be present before interpolation can be done. The minimum requirement for interpolation inside a rock type is:

```
*RPT 1                  ** current rock type
*INTCOMP *AQUEOUS 'Ca++'
*CRVWGT  1.0

*KRINTRP 1              ** Set #1 of rock #1
*INTCOMP_VAL 0.0

*SWT . . .
*SLT . . .

*KRINTRP 2 *COPY 1 1    ** Set #2 of rock #1
*INTCOMP_VAL 1.0e-5
*SWT …                  ** different *SWT only
```

Interpolation parameters

Values of *INTCOMP_VAL, *DTRAPW and *DTRAPN must be monotonically increasing or decreasing, and cannot be too close to each other ($> 10^{-10}$). Only the first table set of each rock type will be used for the reservoir initialization, unless the user specifies the saturation of each grid block through keyword *USER_INPUT in the INITIAL CONDITIONS keyword group. Only the first relative permeability set of a specified rock type will be used for the phase mobility calculation in the well equation if the user has assigned different rock types (or endpoint values) for well layers other than those of the corresponding reservoir blocks (e.g. via keyword *PERF).

*DTRAPW and *DTRAPN are introduced with the empirical foam model in GEM although they can also be in place of *INTCOMP_VAL. If *intval* is specified (via *INTCOMP_VAL) then internally, *intval* = *dtrapw* = *dtrapn*. While either one or both of *DTRAPW and *DTRAPN may be used for a rock type, *INTCOMP_VAL must not be used together with either *DTRAPW or *DTRAPN.

Relative permeability interpolation is done with two consecutive water-oil and liquid-gas relative permeability table sets (*SWT and *SGT/*SLT tables). The computed interpolation value for a grid block is compared to the maximum of the *dtrapw* and *dtrapn* values for each set. If the computed value is less than the maximum of the *dtrapw* and *dtrapn* values of the set, then that set is selected as the first of the *SWT and *SGT/*SLT tables. If the computed

value is outside the intervals formed from all the values of *dtrapw* and *dtrapn* for a rock type, no interpolation is done. Depending on relative values and monotonicity, either the first set or the last set is selected for computing the relative permeability values.

Foam Component Name (Conditional) *FMCOMP

PURPOSE:

Designate a component, a surfactant in aqueous phase, whose composition will be used in dimensionless foam parameter calculations for *FOAM-MODEL *MULTREL.

FORMAT:

*FMCOMP *comp_name*

DEFINITIONS:

comp_name

Name of a component in aqueous phase (a surfactant) whose composition is used in dimensionless foam parameter (FM) calculations.

DEFAULTS:

If *FMCOMP is absent from a given rock type, that has *FOAM-MODEL *MULTREL specified, dimensionless foam parameter will not be calculated.

CONDITIONS:

The keyword must be in the ROCK-FLUID DATA keyword group. It must be placed after *RPT keyword if there are more than one rock types. The keyword is required for *FOAM-MODEL *MULTREL.

Keyword *FMCOMP should not be used with *FOAM-MODEL *INTREL.

EXPLANATION:

Keyword *FMCOMP is used for a rock-type that uses empirical foam option without the use relative permeability sets for interpolation, i.e., *FOAM-MODEL *MULTREL. Only one *FMCOMP keyword may be used per rock-type.

Since it is a surfactant component, *COMPNAME-SURFACTANT must be specified in the Component Properties section of the data file.

Interfacial Tension Table *IFTTABLE, *INTLOG, *INTLIN

PURPOSE:

Define interfacial tension between gas and aqueous phase for use in the empirical foam model or,

Define interfacial tension between oil and aqueous phase for use in the non-foam model.

FORMAT:

```
*IFTTABLE
                cift        sigft
                ..          ..
                ..          ..
*INTLIN
*INTLOG
```

DEFINITIONS:

cift
: Mole fraction in the aqueous phase of the component (typically a surfactant) specified using *INTCOMP *AQUEOUS '*component_name*'

sigft
: Gas-water or oil-water interfacial tension for empirical foam or non-foam model respectively [dyne/cm, i.e., mN/m]

*INTLIN
: Linear interpolation is used when doing a table lookup of composition values, *cift*. This is the default

*INTLOG
: Logarithmic interpolation is used when doing a table lookup of composition values, *cift*.

DEFAULTS:

The *IFTTABLE is absent, then it is assumed that interpolation parameters *DTRAPW, *DTRAPN, or *INTVAL_COMP correspond to the mole fraction itself defined via *INTCOMP.

*INTLIN is the default until overridden by *INTLOG.

CONDITIONS:

This keyword must appear in the Rock-Fluid keyword group and applies to a rock type.

The *IFTTABLE is required for calculating the capillary number dependence in the empirical foam option calculations. One table is needed for each rock type region in which contribution of capillary number in the empirical foam calculations is to be applied.

For empirical foam models where capillary number dependence is not desired, use of *IFTTABLE is not necessary.

In non-foam models, where capillary number based relative permeability interpolation is requested via *CAPNOW option of *INTCOMP, specification of *IFTTABLE is required, but unlike foam option, the second column of the table indicates oil-water interfacial tension.

The *cift*-vs.-*sigft* table must have at least two entries. Each entry must start on a new line.

EXPLANATION:

Interfacial table input is used for computing the contribution of interfacial tension (factors F3 and F4) in the dimensionless foam interpolation calculations. The gas-water capillary number is defined by

$$N_{c,w} = \frac{\mu_g v_g}{\sigma_{gw}}$$

where σ_{gw} is the gas-water interfacial tension, *sigft*, μ_g is gas viscosity and v_g is the gas velocity.

The oil-water capillary number is calculated when *INTCOMP *CAPNOW is specified using oil-wat interfacial tension from the IFT table,

$$N_{c,w} = \frac{\mu_o v_o}{\sigma_{ow}}$$

where σ_{ow} is the oil-water interfacial tension, *sigft*, μ_o is oil viscosity, and v_o is the oil phase velocity.

Foam-Model Definition *FOAM-MODEL

PURPOSE:

Define the type of foam model to be used for relative permeability alterations.

FORMAT:

*FOAM-MODEL *INTREL | *MULTREL

DEFINITIONS:

*INTREL

Based on the value of dimensionless foam-interpolation parameter (FM) in the grid-block, relative permeability is determined by interpolation between two adjacent relative permeability sets. This is the default.

*MULTREL

The gas phase relative permeability is altered by multiplying it by the dimensionless foam-parameter.

DEFAULTS:

The *FOAM-MODEL is absent, then *FOM-MODEL *INTREL is assumed.

CONDITIONS:

This keyword must appear in the Rock-Fluid keyword group and applies to a rock type.

The *FOAM-MODEL *MULTREL is required if the intent is to modify gas relative permeability of the gridblock by multiplying it by dimension-less foam parameter (FM).

EXPLANATION:

Dimensionless foam parameter (FM) is used to alter relative permeabilities in a grid block as a consequence of foam formation using an empirical foam model in GEM. *FOAM-MODEL *INTREL allows user to input two or more relative permeability sets that are generated under experimental conditions reflecting varying dimensionless foam parameter values (specified via *INTCOMP_VAL, *DTRAPN, and/ or *DTRAPW). During simulation, the dimensionless foam parameter is computed for each grid block which is then used for interpolating relative permeabilities between two adjacent relative permeability sets.

On the other hand, *FOAM-MODEL *MULTREL alters the relative permeability to gas by simply multiplying the dimensionless foam parameter to it. Unlike *INTREL, no interpolation is done in this case. Only one set of relative permeability is needed for a rock-type. However, if multiple sets are present in the rock-type only the FIRST set is used. This option does not require use of any of the interpolation keywords like *INTCOMP, *INTCOMP_VAL, *DTRAPN, or *DTRAPW. The surfactant component for foam generation must be specified via keyword *FMCOMP.

See keyword *FMKRW that could optionally be used to alter relative permeability to water when *FOAM-MODEL *MULTREL is in effect.

Basic Foam Interpolation Parameters *FMSURF, *FMCAP, *FMOIL, *FMGCP, *FMOMF, *FMSALT, *FMMOB, *FMPERM1, *FMPERM2, *EPSURF, *EPCAP, *EPOIL, *EPGCP, *EPOMF, *EPSALT, *FLOIL, *FLSALT

PURPOSE:

Assign basic foam interpolating parameters.

FORMAT:

*FMSURF	*fmsurf*
*FMCAP	*fmcap*
*FMOIL	*fmoil*
*FMGCP	*fmgcp*
*FMOMF	*'comp_name' fmomf*
*FMSALT	*fmsalt*
*FMMOB	*fmmob*
*FMPERM1	*fmperm*1
*FMPERM2	*fmperm*2
*EPSURF	*epsurf*
*EPCAP	*epcap*
*EPOIL	*epoil*
*EPGCP	*epgcp*
*EPOMF	*epomf*
*EPSALT	*epsalt*
*FLOIL	*floil*
*FLSALT	*flsalt*

DEFINITIONS:

fmsurf

Critical component mole fraction of surfactant component in the aqueous phase (surfactant component name specified via keywords *INTCOMP *AQUEOUS ..) used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

fmcap

Reference rheology capillary number value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

fmoil

Critical oil saturation value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

fmgcp

Critical generation capillary number value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

'*comp_name*'

An EOS component name that is expected to be in oil phase. Component '*comp_name*' is a component entered via *COMPNAME in the Component Properties keyword group.

fmomf

Critical oil mole fraction for component '*comp_name*' used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

fmsalt

Critical salt mole fraction value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

fmmob

Reference foam mobility reduction factor used in dimensionless foam interpolation calculation. The minimum allowed value is 0, and the suggested maximum is 100,000.

*fmperm*1

Permeability dependence parameter 1. The maximum value is 10^{10}. A value of 0 or less will make the dimensionless foam parameter independent of permeability.

*fmperm*2

Permeability dependence parameter 2. The maximum value is 10^{10}. A value of 0 or less will make the dimensionless foam parameter independent of permeability.

epsurf

Exponent for composition contribution to dimensionless foam interpolation calculation. The allowed range is -4 to 4. The default is 0, which makes foam interpolation independent of composition.

epcap

Exponent for capillary number contribution to dimensionless foam interpolation calculation. The allowed range is -10 to 10.The default is 0, which makes foam interpolation independent of capillary number.

epoil

Exponent for oil saturation contribution to dimensionless foam interpolation calculation. The allowed range is 0 to 5. The default is 0, which makes foam interpolation independent of oil saturation.

epgcp

Exponent for generation cap. no. contribution to dimensionless foam interpolation calculation. The allowed range is -10 to 10.The default is 0, which makes foam interpolation independent of capillary number.

epomf

Exponent for the contribution of the oil phase component *comp_name* to the dimensionless interpolation calculation.. The allowed range is 0 to 5. The default is 0, which makes foam interpolation independent of oil mole fraction of component '*comp_name*'.

epsalt

Exponent for salt contribution to dimensionless foam interpolation calculation. The allowed range is -4 to 4. The default is 0, which makes foam interpolation independent of salt mole fraction.

floil

Lower oil saturation value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

flsalt

Lower salt mole fraction value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

DEFAULTS:

If any of the above keywords is absent its value defaults to zero. If an exponent keyword is absent or it is specified but its value is entered as zero, the interpolation factor is considered independent of the corresponding contributing factor. For example, if *EPOIL is not specified or is specified as zero, the interpolation factor is independent of F2.

CONDITIONS:

The foam interpolation option is possible only if both *INTCOMP and *FMMOB are present in Rock-Fluid data section, and surfactant component name is defined via *COMPNAME-SURFACTANT in the Component Properties section.

Foam option is not supported in presence of chemical equilibrium reactions.

EXPLANATION:

The basic foam interpolation option interpolates between sets of relative permeability curves via the dimensionless interpolation factor, FM, is computed using the following formula:

$$FM = \frac{1}{1 + FMMOB * F1 * F2 * F3 * F4 * F5 * F6 * F7 * FDRY}$$

Where

F1	=	(Mole fraction(icprel) / *fmsurf*) ***epsurf*
F2	=	((*fmoil* – S_o) / (*fmoil* – *floil*))***epoil*
F3	=	(*fmcap* / capillary number)***epcap*
F4	=	((capillary number - *fmgcp*) / *fmgcp*)***epcpg*
F5	=	((*fmomf* – x_{comp_name}) / *fmomf*)***epomf*
F6	=	((x_{salt} – *flsalt*) / (*fmsalt* – *flsalt*))***epsalt*
F7	=	(1/*fmperm*1)*ln((*permav*/*fmperm*2)+1)
FDRY		See "Foam Dry-Out Interpolation Parameters"

In the above nomenclature, icprel refers to the component specified using *INTCOMP *FOAM … for *FOAM-MODEL *INTREL and using *FMCOMP for *FOAM-MODEL *MULTREL, and '*comp_name*' refers to the oil phase component specified with the *FMOMF keyword.

The gas-water capillary number is defined as:

$$N_{c,w} = \frac{\mu_g v_g}{\sigma_{gw}}$$

where σ_{gw} is the gas-water interfacial tension, μ_g is the gas viscosity and v_g the gas velocity.

The permav in expression F7 is geometric mean of permeabilities in x-, y-, and z-directions in the gridblock.

The factor FM is an inverse mobility reduction factor which varies between FM = 1 (no foam) and FM ~ 0 (strongest foam).

The reference foam mobility reduction factor FMMOB is that achieved at measured values of surfactant concentration *fmsurf*, capillary number (flow rate) *fmcap* above two times *fmgcp*, oil saturation $S_o = 0$, and oil mole fraction of component '*comp_name*', i.e., $x_{comp_name} = 0$. The normal range of FMMOB is 5 to 100, depending on the strength of created foam.

The remaining factors account empirically for the effects of surfactant concentration, the positive effect of salt, the detrimental effect of oil and oil composition (component '*comp_name*'), and flow velocity (both generation and shear thinning effects) on foam mobility, essentially scaling FMMOB. Typical values are *fmsurf* = 0.00001 mole fraction, *fmcap* = 0.0001, and *fmoil* = 0.2, while *epsurf* = *epoil* = 1 and *epcap* = 0.5 are appropriate exponents. Additionally *fmgcp* = 10^{-6} and *fmomf* = 0.2, while *epgcp* = *epomf* = 1. Setting an exponent value to zero disables the corresponding contribution.

The simplest application of the foam interpolation option is to rescale gas relative permeability, that is, from k_{rg} to $FM \cdot k_{rg}$ which is available via *FOAM-MODEL *MULTREL.

To account for increased gas trapping with foam, employ higher critical gas saturation in the input foam relative permeability curves.

If the gas saturation in the grid block is zero, the value of the interpolating factor is set to 1 (no foam).

If an exponent keyword is absent or it is specified but its value is entered as zero, the interpolation factor is considered independent of the corresponding contributing factor. For example, if *EPOIL is not specified or is specified as zero, the interpolation factor is independent of F2.

If *fmsurf* = 0 and/or *epsurf* = 0, the interpolation factor is independent of F1.

If *fmoil* ~ *floil* and/or *epoil* = 0, the interpolating factor is independent of F2.

If capillary number ~ 0 and/or *epcap* = 0, the interpolating factor is independent of F3.

If *fmgcp* = 0 and/or *epgcp* = 0, the interpolating factor is independent of F4.

If *fmomf* = 0 and/or *epomf* = 0, the interpolating factor is independent of F5.

If *fmsalt* ~ *flsalt*, and/or *epsalt* = 0 the interpolating factor is independent of F6.

If either *fmperm*1 = 0, or, *fmperm*2 = 0, (or negative) the interpolating factor is independent of F7.

See *SALINITY, *SALINITY-CALC and *SALINR keywords in Component Properties; and Initialization Data sections of the User Guide for specification of salinity in individual grid-blocks. Salt mole fraction, x_{salt}, is derived from salinity.

During computations, if any of the six contributing factors F1 … F6, or FDRY is zero, FM will equal 1 (no foam).

The contribution of parameters F1 through F6 and FDRY to FM can be output via sub-keywords *FMC1SURF, *FMC2COIL, *FMC3CAPN, *FMC4GCAPN, *FMC5OMF, *FMC6SALT, *FMC7PERM, and *FMCDRYOUT of *OUTPRN *GRID; *OUTSRF *GRID; or *OUTSRF *SPECIAL in the Input Output and/or in the Well and Recurrent Data sections. The relative permeability interpolation parameter FM can similarly be output via *KRINTER keyword.

Foam Dry-Out Interpolation Parameters *SFSURF, *SFCAP, *SFOIL, *SFSALT, *SFDRY, *SFBET, *EFSURF, *EFCAP, *EFOIL, *EFSALT, *SLOIL, *SLSALT

PURPOSE:

Assign foam dry-out interpolating parameters.

FORMAT:

*SFSURF	*sfsurf*
*SFCAP	*sfcap*
*SFOIL	*sfoil*
*SFSALT	*sfsalt*
*SFDRY	*sfdry*
*SFBET	*sfbet*
*EFSURF	*efsurf*
*EFCAP	*efcap*
*EFOIL	*efoil*
*EFSALT	*efsalt*
*SLOIL	*sloil*
*SLSALT	*slsalt*

DEFINITIONS:

sfsurf

Critical surfactant component mole fraction value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1.

sfcap

Reference rheology capillary number value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1.

sfoil

Critical oil saturation value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1.

sfsalt

Critical salt mole fraction value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1.

sfdry

Maximum dry-out value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1.

sfbet

Reference dry-out slope used in dimensionless foam dry-out calculation. The allowed range is -10^{-5} to 10^{5}. The suggested minimum value is 0, and the suggested maximum value is 10^{5}.

efsurf

Exponent for composition contribution to dimensionless foam dry-out calculation. The allowed range is -4 to 4. The default is 0, which makes foam interpolation independent of composition.

efcap

Exponent for capillary number contribution to dimensionless foam dry-out calculation. The allowed range is -10 to 10.The default is 0, which makes foam interpolation independent of capillary number.

efoil

Exponent for oil saturation contribution to dimensionless foam dry-out calculation. The allowed range is 0 to 5. The default is 0, which makes foam interpolation independent of oil saturation.

efsalt

Exponent for salt contribution to dimensionless foam dry-out calculation. The allowed range is -4 to 4. The default is 0, which makes foam interpolation independent of salt mole fraction.

sloil

Lower oil saturation value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1.

slsalt

Lower salt mole fraction value used in dimensionless foam dry-out calculation. The allowed range is 0 to 1

DEFAULTS:

If all of these foam dry-out keywords are absent, then no dry-out effects are modeled.

CONDITIONS:

The foam dry-out option is available only if the basic foam interpolation option is used (keywords *FMMOB, etc.).

EXPLANATION:

The foam dry-out option multiplies the basic foam interpolation factor FMMOB (described earlier) by factor FDRY which is calculated as

$$\mathrm{FDRY} = 0.5 + \frac{\arctan\left(sfbet\left(\mathrm{S_w} - \mathrm{SF}\right)\right)}{\pi}$$

where,

$\mathrm{SF} = \max\left(\mathrm{Q1}, \mathrm{Q2}, \mathrm{Q3}, \mathrm{Q4}\right)$, with,

$\mathrm{Qn} = \mathrm{Gn}*\left(1 - sfdry\right) + sfdry, \quad \mathrm{n} = 1,2,3,4$, and

G1 = ((*sfsurf*–Mole Fraction (icprel)) / *sfsurf*)**efsurf*

G2 = (S_o– *sfoil*) / abs(*sfoil-sloil*))**efoil*

G3 = (s*fcap* / (Capillary Number))*******efcap*

G4 = (*sfsalt* –Salt Mole Fraction)/(*sfsalt* – *slsalt*))*******efsalt*

S_w = Water saturation in the grid block

S_o = Oil saturation in the grid block

For Capillary Number, see the explanation section of 'Basic Foam Interpolation Parameters'.

SF is a dry-out factor which varies between SF = 1 (no foam) and SF = 0 (no dry-out). Factor FDRY rescales the basic foam interpolation over a limited saturation range.

Reference dry-out factor SFDRY is the value obtained at measured values of surfactant concentration *sfsurf*, capillary number (flow rate) *sfcap*, and $S_o = 0$. The normal range of SFDRY is 0.01 to 0.05 above irreducible water saturation, depending on the strength of created foam and the porous media structure. Scaling factor *sfbet* is assumed to be constant and depend only on the porous media structure.

The remaining factors account empirically for the effects of surfactant concentration, the detrimental effect of oil, and flow velocity (shear thinning effects) on foam dry-out, essentially scaling SFDRY. Typical values are *sfsurf* = 0.00001 mole fraction, *sfcap* = 0.0001 and *sfoil* = 0.2, while *efsurf* = *efoil* = 1 and *efcap* = 0.5 are appropriate exponents. Setting an exponent value to zero disables the corresponding contribution.

See *SALINITY, *SALINITY-CALC and *SALINR keywords in Component Properties; and Initialization Data sections of the User Guide for specification of salinity in individual grid-blocks. Salt mole fraction is derived from salinity.

The application of the foam dry-out option is to rescale foam relative permeability back to the (no foam) gas relative permeability over the saturation range 0 to SF.

Factor for Foam Water Relative Permeability (Optional)

***FMKRW**

PURPOSE:

Define a factor to be used for altering the water-relative permeability of a rock-type, when empirical foam calculations with *FOAM-MODEL *MULTREL are in effect.

FORMAT:

*FMKRW fmkrw

DEFINITIONS:

fmkrw

Factor used to determine water-relative permeability multiplier for empirical foam option.

DEFAULTS:

If *FMKRW is absent from a given rock type, that has *FOAM-MODEL *MULTREL specified, the water relative permeability will not be modified.

CONDITIONS:

The keyword must be in the ROCK-FLUID DATA keyword group. If required it must be placed after *RPT keyword if there are more than one rock types. The keyword is required for *FOAM-MODEL *MULTREL.

It should not be used with *FOAM-MODEL *INTREL.

The value of *fmkrw* must lie between 0.2 and 1. If it is beyond these limits a fatal error message is issued.

EXPLANATION:

Keyword *FMKRW is used for a rock-type that uses empirical foam option without the use relative permeability sets for interpolation, i.e., *FOAM-MODEL *MULTREL. Only one *FMKRW keyword may be used per rock-type.

The water relative permeability multiplier, fkrw, is calculated using following expression.

$$\text{fkrw} = fmkrw + (1 - fmkrw) * \text{FM}$$

where, FM is the dimensionless foam parameter.

Water-Oil Relative Permeability Table (Conditional) *SWT

PURPOSE:

*SWT indicates the start of the water-oil relative permeability table. The lines entered after *SWT are interpreted as rows of the table until a keyword is encountered.

TABLE:

*SWT (*SMOOTHEND {*LINEAR | *QUAD | *CUBIC |*QUADGEM |
*POWERQ m_{rw} e_{rw} m_{row} e_{row} |
*POWERC m_{rw} e_{rw} m_{row} e_{row} | *ON | *OFF})

For water wet:

Sw	krw	krow	(Pcow)	(Pcowi)
:	:	:	:	:

For oil wet:

So	kro	krwo	(Pcwo)	(Pcwoi)
:	:	:	:	:

DEFINITIONS:

*SMOOTHEND

Optional keyword indicating the type of interpolation to be used for the table intervals where k_{rw} and k_{row} go from zero to non-zero.

*LINEAR

Use sub keyword *LINEAR for linear interpolation between zero and non-zero values of water-oil relative permeability curves. This used to be the default interpolation option before GEM 2014.11.

*QUAD

Use sub keyword *QUAD for quadratic interpolation between zero and non-zero values of water-oil relative permeability curves. This is similar to *QUAD option in IMEX and STARS.

*QUADGEM

Use sub keyword *QUADGEM for quadratic interpolation between zero and non-zero values of water-oil relative permeability curves using a formula which is slightly different from *QUAD. This is the default interpolation option in GEM.

*CUBIC

Use sub keyword *CUBIC for cubic interpolation between zero and non-zero values of water-oil relative permeability curves.

*POWERQ

Use sub keyword *POWERQ for power-law interpolation between zero and non-zero values of water-oil relative permeability curves, defaulting to quadratic smoothing if the value of the exponent (e_{rw}|e_{row}) is outside the

range 1.5 to 4.0. It must be immediately followed by four numbers, m_{rw}, e_{rw}, m_{row}, and e_{row}.

*POWERC

Use sub keyword *POWERC for power-law interpolation between zero and non-zero values of water-oil relative permeability curves, defaulting to cubic smoothing if the value of the exponent (erw|erow) is outside the range 1.5 to 4.0. It must be immediately followed by four numbers, m_{rw}, e_{rw}, m_{row}, and e_{row}.

m_{rw}

Multiplier used in *POWERQ or *POWERC options for k_{rw} curve. There is no default value.

e_{rw}

Exponent used in *POWERQ or *POWERC options for k_{rw} curve. Value entered should be between 1.5 and 4.0. If the value is outside the range, then smoothing option *QUAD will be used instead if *POWERQ is specified or *CUBIC smoothing instead if *POWERC is specified. There is no default value.

m_{row}

Multiplier used in *POWERQ or *POWERC options for k_{row} curve. There is no default value.

e_{row}

Exponent used in *POWERQ or *POWERC options for k_{row} curve. Value entered should be between 1.5 and 4.0. If the value is outside the range, then smoothing option *QUAD will be used instead if *POWERQ is specified or *CUBIC smoothing instead if *POWERC is specified. There is no default value.

*ON

Specifying sub keyword *ON is equivalent to using *QUADGEM.

*OFF

Specifying sub keyword *OFF is equivalent to using *LINEAR.

Sw

Water saturation (fraction). There is no default value.

krw

Relative permeability to water (fraction) at the given water saturation. There is no default value.

krow

Relative permeability to oil (fraction) in the presence of the given water saturation. There is no default value. This column should not be included with the use of the *KRCBM keyword for coal bed methane modeling.

Pcow

Oil-Water capillary pressure (kPa | psi | kPa) on the drainage curve. This column is optional; when omitted it is defaulted to zero and no capillary pressure effects are included in the simulation.

Pcowi

Oil-Water capillary pressure (kPa | psi | kPa) on the imbibition curve. This column is optional; if Pcow is not entered then Pcowi must also not be entered (the fourth column is always assumed to be Pcow). If Pcowi is not entered it is set equal to Pcow by default. Hence if the user does not enter Pcowi, no hysteresis effects are included in capillary pressure; to include hysteresis effects, the user enters values in this column.

So

Oil saturation (fraction), for oil wet case. There is no default value.

kro

Relative permeability to oil (fraction) at the given oil saturation, for oil wet case. There is no default value.

krwo

Relative permeability to water (fraction) in the presence of the given oil saturation, for oil wet case. There is no default value.

Pcwo

Water-oil capillary pressure (kPa | psi | kPa) on the drainage curve. This is the value corresponding to the non-wetting phase (water) minus the wetting phase (oil) pressure. Pcwo = Pw-Po. This column is optional; when omitted it is defaulted to zero and no capillary pressure effects are included in the simulation. For oil wet case, measured under drainage conditions.

Pcwoi

Water-oil capillary pressure (kPa | psi | kPa) on the imbibition curve for the oil wet case. This is the value corresponding to the non-wetting phase (water) minus the wetting phase (oil) pressure. Pcwoi = Pwi-Poi. This column is optional; if Pcwo is not entered then Pcwoi must also not be entered (the fourth column is always assumed to be Pcwo). If Pcwoi is not entered it is set equal to Pcwo by default. Hence if the user does not enter Pcwoi, no hysteresis effects are included in capillary pressure; to include hysteresis effects, the user enters values in this column.

DEFAULTS:

*SWT is a conditional keyword.

*SMOOTHEND is an optional keyword. If *SMOOTHEND is absent then *SMOOTHEND *QUADGEM is used as default.

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group, after the *RPT keyword. One *SWT is required for each *RPT.

If sub keywords *POWERQ or *POWERC of *SMOOTHEND are selected, then it must be followed by four real numbers. These are the multiplying coefficient and exponent of the water curve followed by the coefficient and exponent for the oil in oil/water curve. If the value of the exponent is outside the allowed range of 1.5 to 4.0 then for *POWERQ, quadratic smoothing will be used instead. If the value of the exponent is outside the allowed range of 1.5 to 4.0 then for *POWERC, cubic smoothing will be used instead.

EXPLANATION:

The data entered with the *SWT keyword refer to the rock type set number specified by *RPT.

Optional keyword *SMOOTHEND allows end point smoothing of the water and oil curves, k_{rw} and k_{row} based on the specified method by the sub-keyword that follows. *QUADGEM and *QUAD differ in certain implementation details. *QUAD implementation is closer in form to what is available in simulator IMEX and STARS. For GEM *QUADGEM, which is the default, is recommended.

Denoting S_{wc} as the critical water saturation, in the user entered table designated as the ipt water saturation entry, that is $S_{wc}=S_w(ipt)$, smoothing is applied for a water saturation S_w that is greater than S_{wc}, but less than the next table entry following the $S_{wc} = S_w(ipt)$ entry, that is $S_w(ipt) < S_w < S_w(ipt+1)$.

<u>Linear Smoothing:</u>

Following form is used for *LINEAR for the case of the water relative permeability curve, k_{rw}.

$$k_{rw} = \left(\frac{S_w - S_{wc}}{S_w(ipt+1) - S_{wc}} \right) k_{rw}(ipt+1)$$

<u>Quadratic Smoothing:</u>

Following form is used for *QUAD for the case of the water relative permeability curve, k_{rw}.

Let, $k_1 = (S_w - S_{wc})/(S_w(ipt+1) - S_{wc})$

then, $k_{rw} = k_{rw}(ipt+1)\, k_1^{\,2}$

<u>Quadratic GEM Smoothing:</u>

Following form is used for *QUADGEM for the case of the water relative permeability curve, k_{rw}.

Let, $a = \min((S_w(ipt+1) - S_{wc}), 0.05) + S_{wc}$

and, $q = k_{rw}(ipt+1)\left(\frac{a - S_{wc}}{S_w(ipt+1) - S_{wc}} \right)$

then, $k_{rw} = q\left(\frac{S_w - S_{wc}}{a - S_{wc}} \right)^2$

Cubic Smoothing:

Following form is used for *CUBIC for the case of the water relative permeability curve k_{rw}.

Let, $S_{wp} = S_w - S_{wc}$

$$p = \frac{1}{S_w(ipt+1) - S_{wc}}$$

$$q_1 = \frac{k_{rw}(ipt+1) - k_{rw}(ipt)}{S_w(ipt+1) - S_{wc}}$$

$$q_2 = \frac{k_{rw}(ipt+2) - k_{rw}(ipt+1)}{S_w(ipt+2) - S_w(ipt+1)}$$

$$t = q_1 - pk_{rw}(ipt)$$

$$c_a = p^2(q_2 - 2t)$$

$$c_b = p(3t - q_2)$$

then, $k_{rw} = S_{wp}{}^2(S_{wp}c_a + c_b)$.

Power Law Smoothing:

The power law smoothing (*POWERQ, *POWERC) for the case of the water relative permeability curve uses following form where m_{rw} and exponent e_{rw} are user specified multiplier and exponent respectively.

$$k_{rw} = m_{rw}(S_w - S_{wc})^{e_{rw}}$$

The oil relative permeability smoothing uses similar forms.

Capillary pressure curves determine the height of the transition zones calculated by the *VERTICAL option.

The water-oil capillary pressure is defined as

Pcow = Po – Pw for the water wet case, and

Pcwo = Pw – Po for the oil wet case.

For the water wet case:

Since water is assumed to be the wetting phase, the capillary pressure should decrease monotonically with increasing water saturation. However, the Pcow value at Sw=1 can be nonzero.

Entries must be in order of increasing water saturation. The first table entry is taken to be the connate water saturation.

It is not unusual to encounter water zones in simulation studies, thus it is recommended that the last table entry have $S_w = 1$, with $K_{rw} = 1$ and $K_{row} = 0$ (see FIGURE 7 in Appendix B).

Examples:

```
*SWT
** Sw     krw   krow
   0.12   0.0   1.0
   0.85   0.6   0.0
   1.00   1.0   0.0
```

<u>For the oil wet case:</u>

Since oil is assumed to be the wetting phase, the capillary pressure should decrease monotonically with increasing oil saturation. However, the Pcwo value at So=1 can be nonzero.

Entries must be in order of increasing oil saturation. The first table entry is taken to be the connate oil saturation.

It is not unusual to encounter water zones in simulation studies, thus it is recommended that the last table entry have So = 1-Swcon, with Kro = 1 and Krwo = 0 (see FIGURE 7 in Appendix B).

Examples:

```
*SWT
** So     kro   krwo
   0.12   0.0   1.0
   0.85   0.6   0.0
   0.90   1.0   0.0
```

The acceptable range of values for capillary pressure is:

	SI **kPa**	**Field** **psi**	**Lab** **kPa**
min	-1.0E+4	-1450.0	-1.0E+4
max	1.0E+4	1450.0	1.0E+4

Liquid-Gas Relative Permeability Table (Conditional)

***SLT, *SGT**

PURPOSE:

*SLT indicates the start of a liquid-gas relative permeability table dependent on liquid saturation.

*SGT indicates the start of a liquid-gas relative permeability table dependent on gas saturation.

TABLE:

For the water wet case:

```
*SLT (smoothend_option)
Sl        krg       krog      (Pcog)                (Pcogi)
 :         :         :         :                     :
*SGT (smoothend_option)
Sg        krg       krog      (Pcog)                (Pcogi)
 :         :         :         :                     :
```

For the oil wet case:

```
*SLT (smoothend_option)
Sl        krg       krwg      (Pcog)                (Pcogi)
 :         :         :         :                     :
*SGT (smoothend_option)
Sg        krg       krwg      (Pcog)                (Pcogi)
 :         :         :         :                     :
```

where for both water-wet and oil wet cases:

smoothend_option:

*SMOOTHEND {*LINEAR | *QUAD | *CUBIC |*QUADGEM |
*POWERQ m_{rg} e_{rg} m_{rog} e_{rog} |
*POWERC m_{rg} e_{rg} m_{rog} e_{rog} | *ON | *OFF}

DEFINITIONS:

*SMOOTHEND

Optional keyword indicating the type of interpolation to be used for the table intervals where k_{rg} and k_{rog} go from zero to non-zero.

*LINEAR

Use sub keyword *LINEAR for linear interpolation between zero and non-zero values of water-oil relative permeability curves.

*QUAD

Use sub keyword *QUAD for quadratic interpolation between zero and non-zero values of gas-oil relative permeability curves. This is similar to *QUAD option in IMEX and STARS.

*QUADGEM

Use sub keyword *QUADGEM for quadratic interpolation between zero and non-zero values of gas-oil relative permeability curves using a formula which is slightly different from *QUAD. This is the default interpolation option in GEM.

*CUBIC

Use sub keyword *CUBIC for cubic interpolation between zero and non-zero values of gas-oil relative permeability curves.

*POWERQ

Use sub keyword *POWERQ for power-law interpolation between zero and non-zero values of gas-oil relative permeability curves, defaulting to quadratic smoothing if the value of the exponent (erg|erog) is outside the range 1.5 to 4.0. It must be immediately followed by four numbers, m_{rg}, e_{rg}, m_{rog}, and e_{rog}.

*POWERC

Use sub keyword *POWERC for power-law interpolation between zero and non-zero values of gas-oil relative permeability curves, defaulting to cubic smoothing if the value of the exponent (erg|erog) is outside the range 1.5 to 4.0. It must be immediately followed by four numbers, m_{rg}, e_{rg}, m_{rog}, and e_{rog}.

m_{rg}

Multiplier used in *POWERQ or *POWERC options for k_{rg} curve. There is no default value.

e_{rg}

Exponent used in *POWERQ or *POWERC options for k_{rg} curve. Value entered should be between 1.5 and 4.0. If the value is outside the range, then smoothing option *QUAD will be used instead if *POWERQ is specified or *CUBIC smoothing instead if *POWERC is specified. There is no default value.

m_{rog}

Multiplier used in *POWERQ or *POWERC options for k_{rog} curve. There is no default value.

e_{rog}

Exponent used in *POWERQ or *POWERC options for k_{rog} curve. Value entered should be between 1.5 and 4.0. If the value is outside the range, then smoothing option *QUAD will be used instead if *POWERQ is specified or *CUBIC smoothing instead if *POWERC is specified. There is no default value.

*ON

Specifying sub keyword *ON is equivalent to using *QUADGEM.

*OFF

Specifying sub keyword *OFF is equivalent to using *LINEAR.

Sl

Total liquid saturation (fraction) -- i.e. oil saturation plus connate water saturation for the water wet case and water saturation plus connate oil saturation for the oil wet case. When *SLT is entered, gas saturation is assumed to be (1.0 - Sl)

Sg

Gas saturation (fraction). When *SGT is entered, the krg and krog are the relative permeabilities with the presence of connate water or connate oil (oil wet). Thus krog must be zero at Sg = (1.0 - Swr), and may be zero for smaller values of Sg.

krg

Relative permeability to gas (fraction) for the given saturation.

krog

Relative permeability to oil (fraction) in the presence of gas and connate water for the given saturation. .This column should not be included with the use of the *KRCBM keyword for coal bed methane modeling.

krwg

Relative permeability to water (fraction) in the presence of gas and connate oil for the given saturation. (oil wet case)

Pcog

Gas-oil capillary pressure (kPa | psi | kPa). If this column is not entered it is assumed to be zero and capillary pressure effects are not included in the simulation.

Pcogi

Imbibition Gas-oil capillary pressure (kPa | psi | kPa). This column is optional; Pcogi (column 5) must be entered alongside Pcog (column 4). If Pcogi is not entered, it is set equal to Pcog and gas-oil capillary pressure hysteresis is not modeled.

DEFAULTS:

*SLT or *SLT are conditional keyword.

*SMOOTHEND is an optional keyword. If *SMOOTHEND is absent then *SMOOTHEND *QUADGEM is used as default.

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group, after the *RPT keyword.

Either *SLT or *SGT is required with *RPT, but only one of the two can be entered for a given *RPT entry.

If sub keywords *POWERQ or *POWERC of *SMOOTHEND are selected, then it must be followed by four real numbers. These are the multiplying coefficient and exponent of the gas curve followed by the coefficient and exponent for the oil in gas/oil curve. If the value of the exponent is outside the allowed range of 1.5 to 4.0 then for *POWERQ, quadratic smoothing

will be used instead. If the value of the exponent is outside the allowed range of 1.5 to 4.0 then for *POWERC, cubic smoothing will be used instead.

EXPLANATION:

The data entered with these two keywords refer to the rock type set_number specified by *RPT.

Optional keyword *SMOOTHEND allows end point smoothing of the gas and oil curves, k_{rg} and k_{rog} based on the specified method by the sub-keyword that follows. *QUADGEM and *QUAD differ in certain implementation details. *QUAD implementation is closer in form to what is available in simulator IMEX and STARS. For GEM *QUADGEM, which is the default, is recommended.

Denoting S_{gc} as the critical water saturation, in the user entered table designated as the ipt gas saturation entry, that is $S_{gc}=S_w(ipt)$, smoothing is applied for a water saturation S_g that is greater than S_{gc}, but less than the next table entry following the $S_{gc} = S_g(ipt)$ entry, that is $S_g(ipt) < S_g < S_g(ipt+1)$.

<u>Linear Smoothing:</u>

Following form is used for *LINEAR for the case of the gas relative permeability curve, k_{rg}.

$$k_{rg} = \left(\frac{S_g - S_{gc}}{S_g(ipt+1) - S_{gc}} \right) k_{rg}(ipt+1)$$

<u>Quadratic Smoothing:</u>

Following form is used for *QUAD for the case of the gas relative permeability curve, k_{rg}.

Let, $k_1 = (S_g - S_{gc}) / (S_g(ipt+1) - S_{gc})$

then, $k_{rg} = k_{rg}(ipt+1)\, k_1^{\,2}$

<u>Quadratic_GEM Smoothing:</u>

Following form is used for *QUADGEM for the case of the gas relative permeability curve, k_{rg}.

Let, $a = \min((S_g(ipt+1) - S_{gc}), 0.05) + S_{gc}$

and, $q = k_{rg}(ipt+1) \left(\frac{a - S_{gc}}{S_g(ipt+1) - S_{gc}} \right)$

then, $k_{rg} = q \left(\frac{S_g - S_{gc}}{a - S_{gc}} \right)^2$

<u>Cubic Smoothing:</u>

Following form is used for *CUBIC for the case of the gas relative permeability curve k_{rg}.

Let, $S_{gp} = S_g - S_{gc}$

$$p = \frac{1}{S_g(ipt+1) - S_{gc}}$$

$$q_1 = \frac{k_{rg}(ipt+1) - k_{rg}(ipt)}{S_g(ipt+1) - S_{gc}}$$

$$q_2 = \frac{k_{rg}(ipt+2) - k_{rg}(ipt+1)}{S_g(ipt+2) - S_g(ipt+1)}$$

$$t = q_1 - pk_{rg}(ipt)$$

$$c_a = p^2(q_2 - 2t)$$

$$c_b = p(3t - q_2)$$

then, $k_{rg} = S_{gp}{}^2(S_{gp}c_a + c_b)$.

<u>Power Law Smoothing:</u>

The power law smoothing (*POWERQ, *POWERC) for the case of the gas relative permeability curve uses following form where m_{rg} and exponent e_{rg} are user specified multiplier and exponent respectively.

$$k_{rg} = m_{rg}(S_g - S_{gc})^{e_{rg}}$$

The oil relative permeability smoothing uses similar forms.

Capillary pressure curves determine the height of the transition zones calculated by the *VERTICAL option.

The gas-oil capillary pressure is defined as

Pcog = Pg - Po

Since liquid is assumed to be the wetting phase, the capillary pressure should decrease monotonically with increasing liquid saturation. However, the Pcog value at Sl = 1 or Sg = 0 can be nonzero.

For *SLT, table entries must be in order of increasing liquid saturation. The last entered value for Sl should be 1. The last entered krog value is at Sl = 1 for the water wet case and the last krwg is at Sl=1 for the oil wet case. It is equal to kro at connate water saturation for the water wet case and equal to krw at connate oil for the oil wet case. Thus the last entered krog value in an *SLT table should equal the first entered value of krow in the water-oil relative permeability table for the same set number for the water wet case and the last entered krwg value in an SLT table should be equal to the first entered value of krwo in the water-oil relative permeability table for the same set number for the oil wet case.

For *SGT, table entries must be in order of increasing gas saturation. The first entered value for Sg should be 0. The first entered krog value in an *SGT table (krog at Sg=0) should equal the first entered value of krow in the water-oil relative permeability table (krow at the connate water saturation) for the water wet case. The first entered krwg value in an *SGT table (krwg

at Sg=0) should equal the first entered value of krwg in the water-oil relative permeability table (krwo at connate oil saturation) for the oil wet case.

See Fig. 7 in Appendix B.

Example case for water wet:

```
*SWT
**  Sw   krw   krow
0.12     0.0   1.0
0.85     0.6   0.0
1.00     1.0   0.0
*SLT
**  Sl   krg    krog
0.12    0.984   0.0
0.3     0.942   0.0
0.4     0.87    0.0001
0.5     0.72    0.001
0.6     0.41    0.021
0.7     0.19    0.09
0.8     0.075   0.35
0.9     0.0     0.997
1.0     0.0     1.0
```

The example corresponds to Krocw=1.0 with Swr=0.12, Sgr=0.1, Sorw=0.15 and Sorg=(0.3-Swr)=0.18 See FIGURE 7 in Appendix B for typical relative permeability curves.

Example case for oil wet:

```
*SWT
**  So   kro   krwo
0.12     0.0   1.0
0.85     0.6   0.0
0.90     1.0   0.0
*SLT
**  Sl   krg    krwg
0.12    0.984   0.0
0.3     0.942   0.0
0.4     0.87    0.0001
0.5     0.72    0.001
0.6     0.41    0.021
0.7     0.19    0.09
0.8     0.075   0.35
0.9     0.0     0.997
1.0     0.0     1.0
```

The example corresponds to Krwo(at connate oil)=1.0 with Socon=0.12, Sgr=0.1, Swro=0.15 and Swrg=(0.3-Socon)=0.18

The acceptable range of values for capillary pressure is:

	SI **kPa**	**Field** **psi**
min	-1.0E+4	-1450.0
max	1.0E+4	1450.0

Rock Compressibility for Each Rock Type (Optional) *CROCK

PURPOSE:

*CROCK specifies the rock compressibility for the rock type specified by *RPT

FORMAT:

*CROCK cpor dcpor ref_pressure

DEFINITIONS:

ref_pressure
: Reference pressure (kPa | psi)

cpor
: Pressure dependence of formation porosity (rock compressibility) (1/kPa | 1/psi)

dcpor
: Pressure dependence of the rock compressibility (1/kPa ** 2 | 1/psi ** 2)

DEFAULTS:

Optional keyword. The default for the rock type is the values as specified by *CPOR, *DCPOR and *PRPOR in the reservoir description section.

CONDITIONS:

This keyword must be in the ROCK-FLUID PROPERTY keyword group after each *RPT keyword.

EXPLANATION:

The compressibility entered here refers to the rock type set_number specified by *RPT.

The functional form used in the calculation of porosity por for the rock type is:

por = pori * [1 + cpor * (p - ref_pressure) + dcpor * (p - ref_pressure) ** 2]

where pori is the porosity when p = ref_pressure. pori is input using the *POR keyword. p is the pressure in the oil phase.

Example:

```
CROCK 3.0E-6 0.0 14.7
```

Hysteresis Parameters (Optional) *EPSPC, *EPSPCG, *HYSKRG, *HYSKROW, *HYSKRGM, *HYSKROM,*HYSKROG

PURPOSE:

These keywords signal entry of the hysteresis parameters.

FORMAT:

*EPSPC	epspc
*EPSPCG	epscg
*HYSKRG	sgrmax
*HYSKROW	sormax (water-wet) or swrmax (oil-wet)
*HYSKRGM	sgrmaxm
*HYSKROM	sormaxm
*HYSKROG	sorgmax

DEFINITIONS:

epspc

Dimensionless real number which determines the transition between the imbibition and drainage curves for oil-water capillary pressure. Typical values of epspc lie between 0.05 and 0.1.

epspcg

Dimensionless real number which determines the transition between the imbibition and drainage curves for gas-oil capillary pressure. Typical values of epspcg lie between 0.05 and 0.1.

sgrmax

Maximum residual gas saturation (dimensionless); in practice this is an adjustable parameter which determines the imbibition krg curve as a function of the given drainage curve. sgrmax must lie between zero and one minus the connate water saturation; values of 0.3 - 0.4 are typical.

sormax

Sormax is the maximum possible imbibition residual oil saturation (Oil-water table). It is the endpoint of an imbibition branch which breaks off from the oil relative permeability drainage curve at the maximum possible kro (at So = 1.0-Swcon). This value is used to evaluate the shape and path of all imbibition curves which leave the drainage curve at any saturation. sormax must be greater than Sorw and less than 1.0-Swcon. Normally sormax is significantly larger than Sorw.

sorgmax

sorgmax is the maximum possible imbibition residual oil saturation (Oil-gas table). It is the endpoint of an imbibition branch which breaks off from the oil relative permeability drainage curve at the maximum possible kro (at So = 1.0-Sgcon). This value is used to evaluate the shape and path of all imbibition

curves which leave the drainage curve at any saturation. sorgmax must be greater than Sorg and less than 1.0-Sgcon. Normally sorgmax is significantly larger than Sorg.

swrmax

Swrmax is the maximum possible imbibition residual water saturation (Oil-water table). It is the endpoint of an imbibition branch which breaks off from the water relative permeability drainage curve at the maximum possible krw (at Sw = 1.0-Socon). This value is used to evaluate the shape and path of all imbibition curves which leave the drainage curve at any saturation. Swrmax must be greater than Swcrit and less than 1.0-Socon. Swrmax is normally significantly larger than Swor (residual water saturation, water curve in 2 phase water/oil system). Oil wet case.

sgrmaxm

Maximum residual gas saturation (dimensionless); in practice this is an adjustable parameter which determines the imbibition krg curve as a function of the given drainage curve. sgrmaxm must lie between critical gas saturation and one minus the connate water saturation; values of 0.3 - 0.4 are typical. This value applies when the interfacial dependent relative permeability model is invoked. It applies to case where the miscible curve is being considered and the hydrocarbon phase is considered to be the gas phase.

sormaxm

Sormaxm is the maximum possible imbibition residual oil saturation (Oil-water table). It is the endpoint of an imbibition branch which breaks off from the oil relative permeability drainage curve at the maximum possible kro (at So = 1.0-Swcon). This value is used to evaluate the shape and path of all imbibition curves which leave the drainage curve at any saturation. Sormaxm must be greater than Sorw and less than 1.0-Swcon. Sormaxm is normally significantly larger than Sorw. This value applies when the interfacial dependent relative permeability model is invoked. It applies to case where the miscible curve is being considered and the hydrocarbon phase is considered to be the oil phase.

DEFAULTS:

If *HYSKRG is not entered the hysteresis effect on relative permeabilities is not included in the simulation. If *HYSKRG is entered without the real number sgrmax following it the default is

sgrmax = Sgr + 0.5 * (1-Swr-Sgr),

This default value corresponds to the midpoint of the allowable range.

If *HYSKROW is not entered the hysteresis effect on oil relative permeability (krow) is not included in the simulation. If *HYSKROW is entered without the real number sormax following it,

sormax = Sorw + 0.5 * (1-Swcon-Sorw),

This default value corresponds to the midpoint of the allowable range.

If *HYSKROW is not entered the hysteresis effect on water relative permeability (krwo) is not included in the simulation for the oil wet case. If *HYSKROW is entered without the real number swrmax following it,

swrmax = Swcrit + 0.5 * (1-Socon-Swcrit),

This default value corresponds to the midpoint of the allowable range.

If *HYSKROG is not entered the hysteresis effect on oil relative permeability (krog) is not included in the simulation. If *HYSKROG is entered without the real number sorgmax following it, sorgmax is calculates as:

sorgmax = Sorg + 0.5 * (1-Sgcon-Sorg-Swcon),

This default value corresponds to the midpoint of the allowable range.

If the *HYSKRGM is not entered the default is to take the value entered via *HYSKRG

If the *HYSKROM is not entered the default is to take the value entered via *HYSKROW

The water-oil capillary pressure hysteresis effect is included if the fifth column (tpcowi/tpcwoi) of the tables introduced by *SWT is specified. The default for epspc is 0.1. If *EPSPC is entered without a real number following it, epspc = 0.1 is assumed.

The oil-gas capillary pressure hysteresis effect is included if the fifth column (tpcogi) of the tables introduced by *SLT (or *SGT) is specified. The default for epspcg is 0.1. If *EPSPCG is entered without a real number following it, epspcg = 0.1 is assumed.

CONDITIONS:

These keywords must be in the ROCK-FLUID DATA keyword group, after the *RPT keyword. Do NOT use the hysteresis option when the interfacial tension effects on the relative permeabilities are included.

EXPLANATION:

The epspc, epspcg, sgrmax and sormax refer to the rock type set_number specified by *RPT.

If more than one rock type is present then *EPSPC, *EPSPCG, *HYSKRG and *HYSKROW must be specified before the next *RPT keyword. After the next *RPT keyword, a new set of *EPSPC, *EPSPCG, *HYSKRG, *HYSKROW and *HYSKROG can be specified for the new rock type set_number.

When hysteresis is included, the water-oil capillary pressure is expressed as follows. If the capillary pressure is initially on the drainage curve (decreasing Sw) and Sw begins to increase, then pcow is calculated from the drainage-to-imbibition scanning curve

pcow(Sw) = (f)pcow(Im,Sw) + (1-f)pcow(Dr,Sw);

here f lies between zero and one, and is given by

$$f = \left(\frac{Sw(max) - Sw(hyst) + epspc}{Sw(max) - Sw(hyst)}\right)\left(\frac{Sw - Sw(hyst)}{Sw - Sw(hyst) + epspc}\right)$$

where Sw(hyst) is the value which Sw had when the reversal occurred and Sw(max) is the maximum attainable water saturation, 1-Sorw. pcow remains on this scanning curve until Sw falls below Sw(hyst), when it reverts to the drainage curve. If the capillary pressure is initially on the imbibition curve (increasing Sw) and Sw begins to decrease, then pcow is calculated from the imbibition-to-drainage scanning curve:

pcow(Sw) = (f)pcow(Dr,Sw) + (1-f)pcow(Im,Sw),

where

$$f = \left(\frac{Sw(hyst) - Swr + epspc}{Sw(hyst) - Swr}\right)\left(\frac{Sw(hyst) - Sw}{Sw(hyst) - Sw + epspc}\right)$$

with Swr the connate water saturation. pcow remains on this scanning curve until Sw exceeds Sw(hyst), when pcow reverts to the imbibition curve. Define dSmax to be the maximum amount by which water saturation changes in one continuous drainage or imbibition cycle. If epspc is much larger than dSmax, the transition between curves is gradual, being nearly linear in Sw; if epspc is much smaller than dSmax the transition is sudden, with the full transition to the second curve occurring when Sw differs from Sw(hyst) by only a small amount. The typical value of 0.05 to 0.1 for epspc generally leads to a moderate transition.

When Pcog hysteresis is included, the oil-gas capillary pressure is expressed as follows. Sl below refers to liquid saturation in the oil (liquid) – gas table. If the capillary pressure is initially on the drainage curve (decreasing Sl) and Sl begins to increase, then pcog is calculated from the drainage-to- imbibition scanning curve

pcog(Sl) = (f) x pcog(Im,Sl) + (1-f) x pcog(Dr,Sl);

here f lies between zero and one, and is given by

$$f = \left(\frac{Sl(max) - Sl(hyst) + epspcg}{Sl(max) - Sl(hyst)}\right)\left(\frac{Sl - Sl(hyst)}{Sl - Sl(hyst) + epspcg}\right)$$

where Sl(hyst) is the value which Sl had when the reversal occurred and Sl(max) is the maximum attainable liquid saturation 1-Sgcon. Pcog remains on this scanning curve until Sl falls below Sl(hyst), where it reverts to the drainage curve. If the capillary pressure is initially on the imbibition curve (increasing Sl) and Sl begins to decrease, then pcog is calculated from the imbibition-to-drainage scanning curve:

pcog(Sl) = (f) x pcog(Dr,Sl) + (1-f) x pcog(Im,Sl),

where

$$f = \left(\frac{Sl(hyst) - Slr + epspcg}{Sl(hyst) - Slr}\right)\left(\frac{Sl(hyst) - Sl}{Sl(hyst) - Sl + epspcg}\right)$$

with Slr the connate liquid saturation. Pcog remains on this scanning curve until Sl exceeds Sl(hyst), where pcog reverts to the imbibition curve. Let dSmax be the maximum amount by which water saturation changes in one continuous drainage or imbibition cycle. If epspcg is much larger than dSmax, the transition between curves is gradual, being nearly linear in Sl; if epspcg is much smaller than dSmax the transition is sudden, with the full transition to the second curve occurring when Sl differs from Sl(hyst) by only a small amount. The typical values of 0.05 to 0.1 for epspcg generally lead to a moderate transition. epspcg cannot be less than 1.0d-05.

The hysteresis effect on krg is calculated as follows:

krg(Sg) = krg(Dr;Sg),during drainage;

krg(Sg) = krg(Dr;Sg(shifted)), during imbibition;

where

Sg(shifted) = Sgr+(Sg-Sgrh)(Sgh-Sgr)/(Sgh-Sgrh).

krg is assumed to lie on the drainage curve initially; Sgh is the value of Sg when the shift to imbibition occurs and Sgrh is the value of Sgr corresponding to Sgh via Land's equation,

$$\frac{1}{Sgrmax - Sgr} - \frac{1}{Sgmax - Sgr} = \frac{1}{Sgrh - Sgr} - \frac{1}{Sgh - Sgr}$$

where Sgmax = 1 - Swr. Sgrmax has the value of the user-entered parameter sgrmax.

The hysteresis effect on krow is calculated as follows:

krow(Sw) = kro(Dr;Sw),during drainage;
krow(Sw) = kro(Dr;Sw(shifted)), during imbibition;

where

Sw(shifted)=1.0 - Sorw+(1-Sw-Sorh)(Soh-Sorw)/(Soh-Sorh).

krow is assumed to lie on the drainage curve initially; Soh is the value of So when the shift to imbibition occurs and Sorh is the value of Sorw corresponding to Soh via Land's equation,

$$\frac{1}{Sormax - Sorw} - \frac{1}{Somax - Sorw} = \frac{1}{Sorh - Sorw} - \frac{1}{Soh - Sorw}$$

where Somax = 1 - Swcon. Sormax has the value of the user-entered parameter via keyword *HYSKROW.

The hysteresis effect on krog is calculated as follows:

krog(Sg) = krog(Dr;Sg),during drainage;
krog(Sg) = krog(Dr;Sg(shifted)), during imbibition;

where

Sg(shifted)=1.0–Swcot- Sorg+(1-Sg-Sorh)(Soh-Sorg)/(Soh-Sorh).

krog is assumed to lie on the drainage curve initially; Soh is the value of So when the shift to imbibition occurs and Sorh is the residual (trapped) oil saturation value corresponding to Soh via Land's equation,

$$\frac{1}{Sorgmax - Sorg} - \frac{1}{Somax - Sorg} = \frac{1}{Sorh - Sorg} - \frac{1}{Soh - Sorg}$$

where Somax = 1 – Swcon-Sgcon. Sorgmax has the value of the user-entered parameter via the keyword *HYSKROG.

Note: Keyword *HYSKROW was earlier known as *HYSKRO which is still supported but is deprecated.

Example:

```
*EPSPC  0.05 *EPSPCG 0.05
*HYSKRG 0.4
*HYSKROW 0.4
*HYSKROG 0.4
```

WAG Relative Permeability Hysteresis Modelling (Optional)

***HYS_MODEL, *ALPHAKRG , *ARESOILKRG ,*SWT3**

PURPOSE:

Keyword *HYS_MODEL allows the user to select between available options for modeling relative permeability hysteresis for the non-wetting phase (gas) and the wetting phase (water). Keywords ALPHAKRG and ARESOILKRG allow the user to specify parameters of the three phase WAG hysteresis model for the gas phase. Keyword SWT3 allows the user to enter the 'three phase' water relative permeability curve. This curve as well as the two phase flow curve entered with the keyword *SWT are utilized in modeling water phase relative permeability hysteresis. These keywords must be specified in the Rock-Fluid section of the GEM data file.

FORMAT:

```
*HYS_MODEL        *3PWAG      *GAS
                              *WATER
                              *GASANDWATER
                  *LAND
                  *CARLSONANDLAND
*ALPHAKRG         alpha
*ARESOILKRG       a
```

TABLE:

```
*SWT3
Sw        krw3
 :         :
```

DEFINITIONS:

HYS_MODEL

Allows the user to include relative permeability hysteresis for the gas and/or water phase and to select the specific hysteresis model for the gas phase. In the case of the gas phase either the Land (original hysteresis model implemented in GEM), the Carlson and Land model or the three-phase WAG hysteresis model may be selected. For the water phase only the three-phase WAG hysteresis model is available.

3PWAG

This keyword indicates that three-phase WAG hysteresis model is to be applied. The user can choose to apply this model to the gas phase alone, to the water phase alone or to both gas and water phases. Please note that if this model is to be applied to the gas phase then the maximum trapped gas saturation must also be entered using the keyword *HYSKRG. If the model is to be applied for the water phase then a second relative permeability curve, a three phase flow curve must be entered using the keyword *SWT3.

LAND

The selection of this keyword will result in the hysteresis model originally developed in GEM being applied for the gas phase. This model uses the Land trapping model. Please note that if this model is to be applied for the gas phase then the maximum trapped gas saturation must also be entered using the keyword *HYSKRG.

CARLSONANDLAND

The selection of this keyword will result in the Carlson hysteresis model being applied for the gas phase. This model uses the Land trapping model. Please note that if this model is to be applied to the gas phase then the maximum trapped gas saturation must also be entered using the keyword *HYSKRG.

The Carlson and Land model is described by Equations 1, 2 and 3 below (see discussion on the 3 phase WAG model). Unlike the WAG model however no distinction is made on the direction of saturation change while on a scanning curve. The same flowing saturation is used to calculate relative permeability for gas saturation increasing or decreasing while on a scanning curve.

GAS

The selection of this sub keyword following 3PWAG indicates that the user wants to use the WAG hysteresis model for the gas phase and no hysteresis for the water phase. Please note that the maximum trapped gas saturation must also be entered using the keyword *HYSKRG.

WATER

The selection of this sub keyword following 3PWAG indicates that the user wants to use the WAG hysteresis model for the water phase. If the maximum trapped gas saturation is ALSO entered using the keyword *HYSKRG, then the Land model will be applied for gas phase hysteresis. A three phase water relative permeability table must also be entered using the keyword SWT3.

GASANDWATER

The selection of this sub-keyword following 3PWAG indicates that the user wants to use the WAG hysteresis model for both gas and water phases. The maximum trapped gas saturation must also be entered using the keyword *HYSKRG, and a three phase water relative permeability table must be entered using the keyword SWT3.

ALPHAKRG

This keyword is used to enter the value of the reduction exponent used in calculating drainage (increasing gas saturation) relative permeability. This parameter is shown as the symbol “α” in equation (4) presented below under the discussion paragraph.

ARESOILKRG

This keyword is used to enter the value of the parameter “a” in equation (5) presented below under the discussion paragraph. It is used in modifying the residual oil saturation required as input to the STONE 1 oil phase relative

permeability model. This parameter is only relevant if STONE1 is specified following KROIL.

SWT3

This keyword is used to enter a three phase relative permeability curve for the water phase. Following this keyword a 2 column table of water saturation versus water relative permeability is required. The table must start on a new line. The range of water saturation must span the same range as the two phase table entered with the keyword *SWT. The first and last water saturation values must be the same as those entered with the keyword SWT. This is because the three phase table is scaled in the same manner as the two phase table. However the end points used in scaling are inferred from the two phase table. The lines entered after *SWT3 are interpreted as rows of the table until a keyword is encountered.

alpha

Parameter of the gas phase hysteresis model. This value is the reduction exponent used in calculating drainage (increasing gas saturation) relative permeability. This parameter is shown as the symbol "α" in equation (4) presented below under the discussion paragraph.

a

Parameter of the gas phase hysteresis model. It is used in modifying the residual oil saturation. The residual oil saturation is an input parameter to the STONE 1 three phase oil phase relative permeability model. Equation (5) below shows how this parameter is used to modify the residual oil saturation. This parameter is only used if STONE1 is specified following KROIL keyword.

Sw

Water saturation (fraction). There is no default value.

krw3

Three-phase relative permeability to water (fraction) at the given water saturation. There is no default value.

DEFAULTS:

Optional keyword. Defaults are:

HYS_MODEL *LAND
ALPHAKRG $\alpha = 1.0$
ARESOILKRG $a = 1.0$

CONDITIONS:

This option should be used only when the reservoir is water wet.

These keywords must appear in the Rock-Fluid Data keyword group.

For gas phase hysteresis the keyword *HYSKRG is also required.

EXPLANATION:

The cornerstone of the quantitative prediction of multiphase flow in porous media is the concept of relative permeability, which accounts for the reduction in flow of a given phase due to the mutual interaction of the different flowing phases. In general the relative permeability of a given phase is a function of the saturations as well as many other rock and fluid descriptors such as wettability, flow rate, interfacial tension and saturation histories. The dependence of the relative permeability on the saturation path or history is referred to as hysteresis.

The hysteresis models used in GEM when the keywords HYSKRG or HYSKROW are specified can be thought of as two phase models where the relative permeability of a given phase is a function of its own saturation and saturation history. For a given phase, a bounding drainage curve is required as well as Land's parameter which essentially allows the bounding imbibition as well as all scanning curves to be calculated. If a saturation direction reversal occurs once on a scanning curve then the same scanning curve is retraced until a bounding curve is reached. Therefore an assumption of reversibility of scanning curves is also built in for the two phase model.

The two phase hysteresis models have been found to be inadequate especially in modeling processes such as water alternating gas injection (WAG) where simultaneous flow of all three phases occurs. For these processes more rigorous models are required where the relative permeability of a given phase is a function of the saturation history of not only its own phase but of one other phase as well. In addition the assumption of the reversibility of drainage and imbibition scanning curves needs to be removed. This is consistent with observed behavior, for example it has been observed that the gas relative permeability is higher when displacing pure oil than when displacing a water oil mixture following a water flood. These types of models are known as three-phase hysteresis models.

Three-phase hysteresis model implemented in GEM

The three phase hysteresis model implemented in GEM is based on the work of Larsen and Skauge [1]. Notation in the equations below follows this reference.

The gas phase (non-wetting phase) hysteresis model follows the theory of Land and Carlson. The water phase (wetting phase) model interpolates between two-phase and three-phase relative permeability curves, where the three-phase water relative permeability is interpreted as the relative permeability after gas flooding. The final element of the model is the reduction of the minimal oil saturation (Som) used in Stone's first model for three-phase oil relative permeability.

For the non-wetting phase (gas) consider a typical drainage process (increasing gas saturation) reaching a maximum gas saturation Sgm followed by an imbibition process (decreasing gas saturation) leading to a trapped gas saturation Sgr The trapped gas saturation is given by equation 1:

$$S_{gr} = S_{gcrit} + \frac{\left(S_{gm} - S_{gcrit}\right)}{\left(1 + C\left(S_{gm} - S_{gcrit}\right)\right)} \qquad (1)$$

Where:

C is Land's parameter, calculated by GEM to be consistent with Sgcrit and the maximum S_g entered in the SLT or SGT table, and the Sgrmax value entered with HYSKRG.

Sgcrit is the critical gas saturation.

The gas relative permeability on the drainage to imbibition scanning curve is given by:

$$K_{rg}\left(S_g\right) = K_{rg}^{drain}\left(S_{gf}\right) \qquad (2)$$

The free gas saturation Sgf is calculated from Land's equation:

$$S_{gf} = S_{gcrit} + \frac{1}{2}\left\{\left(S_g - S_{gr}\right) + \sqrt{\left(S_g - S_{gr}\right)^2 + \frac{4}{C}\left(S_g - S_{gr}\right)}\right\} \qquad (3)$$

If the gas saturation once again decreases, then a secondary drainage curve will be followed. The secondary drainage curves are calculated using the following equation:

$$K_{rg}^{drain} = \left[K_{rg}^{input} - K_{rg}^{input}\left(S_g^{start}\right)\right]\left[\frac{S_{wcon}}{S_w^{start}}\right]^{\alpha} + \left[K_{rg}^{imb}\left(S_g^{start}\right)\right] \qquad (4)$$

Where:

K_{rg}^{drain} is the relative permeability calculated for the secondary drainage

K_{rg}^{input} is the input relative permeability at Sg

$K_{rg}^{input}\left(S_g^{start}\right)$ is the input relative permeability at the gas saturation at the start of the secondary drainage curve

S_{wcon} is the connate water saturation

S_w^{start} is the water saturation at the start of the secondary drainage curve.

$K_{rg}^{imb}\left(S_g^{start}\right)$ is the relative permeability at the start of the secondary drainage process (that is the K_{rg} at the end of the imbibition curve)

α is the reduction exponent input with the ALPHAKRG keyword.

The term raised to the exponent α in equation (4) accounts for reduction of the gas phase relative permeability in the presence of mobile water. Equation (4) implies that the gas phase relative permeability is a function of the history of both the gas and water phase saturations.

Further imbibition curves are handled by transforming the saturation using Land's equation (3) with the trapped gas saturation Sgr defined as the maximum trapped gas over all the WAG cycles. The transformed saturation is then looked up on the secondary drainage curve.

If Stone's first model for three-phase oil relative permeability model is used, the minimal residual oil saturation Som may be modified to account for trapped gas saturation:

$$Som_{mod} = Som - \left(a\left(S_g - S_{gf}\right)\right) \qquad (5)$$

Where "a" is input by the user with the keyword ARESOILKRG.

It has been observed that water mobility following a gas flood is significantly reduced relative to its mobility in the original oil-water system. In addition to the usual two phase relative permeability curve (oil-water flow) another curve, for a three phase flow scenario,

that represents the water relative permeability following a gas flood is required. This is the curve input with the SWT3 keyword.

For an imbibition process (Sw increasing) the relative permeability is interpolated between the two and three phase curves as follows:

$$K_{rw}^{imb} = K_{rw}^{W2}\left(1 - \frac{S_g^I}{Sg_{max}}\right) + K_{rw}^{W3}\left(\frac{Sg^I}{Sg_{max}}\right) \qquad (6)$$

Where:

K_{rw}^{imb} is the calculated imbibition relative permeability

K_{rw}^{W2} is the relative permeability at Sw read from the two-phase curve

K_{rw}^{W3} is the three-phase relative permeability at Sw

Sgmax is the maximum attainable gas saturation, 1 − Swcon − Soirg

S_g^I is the gas saturation at the start of the increasing water saturation (imbibition) process

Swcon is the connate water saturation

Soirg is the irreducible oil in the gas-liquid table

A subsequent drainage relative permeability will be calculated by interpolation between the imbibition curve and either the three-phase curve or the two-phase curve, depending on the gas saturation.

Plots of key hysteresis model gas saturations

The user has the option of writing out to the RESULTS output system for plotting certain key hysteresis related variables. This includes the gas saturation corresponding to the reversal point from drainage to imbibition (*SGHYS). This is one of the end points of the drainage to imbibition scanning curve, corresponding to zero trapped gas saturation. The other end point corresponds to the largest gas saturation at which the gas relative permeability is zero or the maximum trapped gas saturation (*SGRHYS). At any other value of the gas saturation on the scanning curve the dynamic trapped gas saturation (*SGDTHY) will be some value less than the maximum trapped gas saturation.

To write out the gas saturation(s) add the above keywords to the list of grid variables selected for plotting, for example:

*OUTSRF *GRID *SG *KRG *SGHYS *SGDTHY *SGRHYS

References:

1. Larsen, J.A and Skauge A, “Methodology for Numerical Simulation with Cycle-Dependent Relative Permeabilities”, SPE Journal, June 1998. SPE Paper # 38456.

Rock Type (Optional) *RTYPE

PURPOSE:

*RTYPE indicates the start of input of rock types. A relative permeability curve set_number is assigned to each grid block (see *RPT).

ARRAY:

```
*RTYPE  set_number_array
```

DEFAULTS:

Optional keyword. The default is *RTYPE *CON 1, where 1 is a value for set_number; this uses the first relative permeability curve set for all grid blocks.

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group, after all the two phase relative permeability curve tables or curve data have been entered.

EXPLANATION:

If more than one set of corresponding water-oil and liquid-gas relative permeability tables is input, then a set of curves must be assigned to each grid block.

*RTYPE indicates the input of an array of grid values. All array reading option subkeywords are valid. See the section on ARRAY READING OPTIONS for more details.

Trapped Oil Saturation (Optional) *SORM

PURPOSE:

*SORM indicates that a trapped oil saturation is to be read for each grid block.

ARRAY:

*SORM

DEFAULTS:

Optional keyword. By default, the trapped oil saturations are set to 0.0

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group, after all the two phase relative permeability curve tables or curve data have been entered.

EXPLANATION:

The trapped oil with volume corresponding to these saturations is not recoverable. The trapped oil is immobile and always remains as an oil phase. It can expand and be compressed as pressure changes. The overall effect of the trapped oil is equivalent to reducing the recoverable pore volumes.

In each grid block, SORM must be less than the residual oil saturation in the grid block assigned through relative permeability curves and after taking into account any endpoint scaling. An error message is issued if this condition is violated.

When oil saturation (*SO) output is requested, the values include the trapped as well as non-trapped oil. However, the output of other related grid block properties, like, component oil composition (*X), component global composition (*Z), and component moles per grid block volume (*RMOL) consider only non-trapped oil.

While computing recoveries, the initial oil volume in place includes trapped as well as non-trapped oil. Similarly ‘Mole Percent Hydrocarbon Recovered’ is calculated with ‘Total-HC-originally-in-place’, that also includes trapped oil moles.

Interfacial Tension Effects on Relative Permeability (Optional)

***SIGMA, *IFTMISPT**

PURPOSE:

*SIGMA indicates that relative permeabilities and capillary pressures are dependent on interfacial tension; the parameters for calculating the dependence follow the keyword.

Use of keyword *IFTMISPT ensures calculation of relative permeabilities and capillary pressures that change smoothly from two-phase to single-phase and vis.-a-versa, during miscibility transitions.

FORMAT:

```
*SIGMA  eksig  epsig  sigma0
*IFTMISPT (sigmis_thr)
```

DEFINITIONS:

eksig
: Exponent for gas-oil relative permeabilities (dimensionless)

epsig
: Exponent for gas capillary pressure (dimensionless)

sigma0
: Reference interfacial tension (mN/m) for relative permeability calculations

sigmis_thr
: Threshold interfacial tension (mN/m) for the gas and oil phases in a grid block to be considered to be miscible. Block is miscible if sigma < sigmis_thr.

DEFAULTS:

Optional keywords.

By default, (i.e. if the *SIGMA keyword is not present) interfacial tension has no effect on relative permeability and capillary pressure.

If *IFTMISPT is specified then the calculation is designed to ensure smoothly varying relative permeabilities and capillary pressures as a grid block transitions from two phase miscible state to a single phase state or vis.-a-versa.

If *SIGMA alone is entered, the default values are

eksig	= 0.1
epsig	= 1.0
sigma0	= 0.01

If *IFTMISPT is entered without a value, sigmis_thr, is defaulted to 0.001.

CONDITIONS:

These keywords, if present, must be in the ROCK-FLUID DATA keyword group. The *SIGMA keyword is a prerequisite for the *IFTMISPT keyword.

EXPLANATION:

The Land (*HYS_MODEL *LAND) and Carlson-Land (*HYS_MODEL *CARLSONANDLAND) hysteresis options can be used together with the interfacial tension dependent option (invoked with the keyword *SIGMA) if the system is specified to be water wet. Hysteresis effects on the immiscible/miscible curves, krg and krow can be invoked by the keywords *HYSKRG and *HYSKROW. Maximum trapped gas saturation for the miscible water – 'gas' system can be entered via *HYSKRGM or defaulted based on the value entered with *HYSKRG. Maximum trapped oil saturation for the miscible water – 'oil' system can be entered via *HYSKROM or defaulted based on the value entered with *HYSKRO.

For the immiscible case, gas phase relative permeability is a function of the gas phase saturation and looked up based on the user entered *SGT/*SLT table. For the immiscible case, the oil relative permeability in the water – oil two phase system (krow) is looked up from the entered *SWT table as a function of water saturation and the oil relative permeability in the two phase gas – oil system (krog) is looked up from the entered *SGT/SLT as a function of gas saturation. A final value of the relative permeability for the oil phase immiscible curve is calculated based on the krow and krog values and a three phase model such as Stone's model 1 or 2.

For the miscible case, the oil and gas phase are considered to be fully miscible, the system reduces to a two phase water hydrocarbon system. For the purpose of relative permeability calculations, the hydrocarbon phase can be considered to be either gas or oil. When the hydrocarbon phase is considered to be gas, the relative permeability is looked up from the user entered table (*SGT/*SLT) with the gas saturation equal to one minus the water saturation in the grid block. In this case imbibition corresponds to increasing water saturation and drainage decreasing water saturation. If the hydrocarbon phase is considered to be oil, then the system reduces to two phase oil water. The oil phase relative permeability is looked up from the user entered table *SWT based on the existing water saturation in the grid block. Imbibition corresponds to increasing water saturation and drainage decreasing water saturation. The final miscible relative permeability are constructed for the oil and gas phases by averaging the miscible 'gas'-water and 'oil'-water relative permeabilities to get a hydrocarbon phase relative permeability and then multiplying it by the individual gas or oil phase saturation.

The purpose of including the interfacial tension effect on relative permeabilities is to force the relative permeability curves for gas and oil to become linear functions of the respective saturations when the phases become nearly indistinguishable (and hence the interfacial tension between the two phases approaches zero). Denoting the modified kro and krg by krot and krgt respectively, the expressions for krot and krgt are as follows:

$$\mathrm{krot} = \mathrm{f} * \mathrm{kro} + (1 - \mathrm{f}) * \mathrm{krh} \frac{\mathrm{So}}{(1 - \mathrm{Sw})}$$

$$\mathrm{krgt} = \mathrm{f} * \mathrm{krg} + (1 - \mathrm{f}) * \mathrm{krh} \frac{\mathrm{Sg}}{(1 - \mathrm{Sw})}$$

where

krh	=	0.5 * (krow(Sw) + krg(Sg = 1 - Sw)) ;
f	=	1, if sigma > sigma0;
	=	(sigma/sigma0) ** eksig, otherwise ;
sigma	=	Interfacial tension between oil and gas in dynes/cm, calculated using the MacLeod- Sugden correlation (Reid et al., 1977).

When eksig is large compared to one, the transition to the linear function occurs suddenly as soon as sigma drops below sigma0; when eksig is small compared to one the transition is delayed and occurs suddenly when sigma drops to a small fraction of sigma0. When eksig = 1 the transition is a linear function of sigma and is gradual.

Similarly the interfacial tension effect on the gas-oil capillary pressure forces the pressure difference between the two phases to approach zero as the phases become indistinguishable. Denoting the modified pcog by pcogt, the expression for pcogt is as follows:

pcogt	=	pcog, if sigma > sigmap0;
	=	pcog * (sigma/sigmap0) ** epsig, otherwise.

When epsig is large compared to one, the transition to zero capillary pressure occurs suddenly as soon as sigma drops below sigmap0; if epsig is small compared to one, the transition is delayed and occurs suddenly when sigma drops to a small fraction of sigma0. When epsig = 1 the transition is a linear function of sigma and occurs gradually.

There is a need to ensure that the calculated relative permeabilities and capillary pressure change in a smooth and continuous way as a grid block transitions from two phases (oil & gas) at or near miscible state to a single phase state (oil or gas). From a rigorous stand point a value of the interfacial tension can only be calculated if two equilibrium phases exist in a grid block. To ensure that the relative permeability assigned to a single phase grid block which has transitioned from a miscible two phase state (grid block for which sigma < sigmis_thr) varies in a smooth way, a “pseudo” value of the interfacial tension is calculated for the single phase grid block, when keyword *IFTMISPT is used.

The value of the pseudo interfacial tension for a single phase grid block is calculated using the composition and properties of a reference phase. It can be output to the Simulation Results grid and special history as well as to the output file. See sub-keyword *PSIG for *OUTSRF and *OUTPRN in the Input / Output Control section.

The *IFTMISPT option may be useful for situations where the *SIGMA model experiences difficulty in convergence due to frequent change from two- to single-phase and or vis.-a-versa and there by use of incoherent relative permeabilities and capillary pressures. However, the option needs to be used with caution as it may increase computation time as well as some storage overhead due to additional calculations and use of tracking arrays.

Velocity Dependent Relative Permeability Option (Optional)

***VELDEPRP**

PURPOSE:

*VELDEPRP indicates that relative permeability is dependent on phase velocity and interfacial tension; the parameters for calculating the dependence follow the keyword. This option is designed to model gas condensate systems where the reduction in relative permeability due to condensate dropout is mitigated by low interfacial tension and/or the high fluid velocities, especially in the near well region. The dependence of the relative permeability on the capillary number (CN) has been validated through experimental measurements.

FORMAT:

*VELDEPRP (*MODEL1)	(*GAS)	$T_g\ N_{cbg}\ \tau_g\ S_{rg}^{high}\ k_{rg}^{o\ high}\ \alpha_g$
	(*CONDENSATE)	$T_c\ N_{cbg}\ \tau_c\ S_{rc}^{high}\ k_{rc}^{o\ high}\ \alpha_c$
	(*WATER)	$T_w\ N_{cbw}\ \tau_w\ S_{rw}^{high}\ k_{rw}^{o\ high}\ \alpha_w$
or		
VELDEPRP (*MODEL2)	(*GAS)	$m_g\ N_{cbg}\ n_g$
	(CONDENSATE)	$m_c\ N_{cbg}\ n_c$

DEFINITIONS:

*MODEL1

Selection of the model published by Pope et al. Note that parameters for all three phases are required. Keywords *GAS, *CONDENSATE and *WATER followed by appropriate values are ALL required. This is NOT compatible with previous versions of GEM which only required *GAS and *CONDENSATE keywords. To avoid aqueous phase relative permeability from being revised based on capillary number, enter a large value for the base capillary number, for example N_{cbw} = 1.0E+20.

*MODEL2

With this selection the relative permeability is calculated by interpolating between the base and straight line miscible curves via a scaling function.

T_g, T_c, T_w

Parameters of the Pope et al. model for the gas, condensate and aqueous phases respectively (no defaults).

$N_{cbg}\ N_{cbc} N_{cbw}$

Base or threshold capillary number for the gas, condensate and aqueous phase respectively. If the internally calculated capillary number is less than the base capillary number then the relative permeability is equal to the entered base relative permeability, that is, there is no capillary number effect (no defaults).

τ_g, τ_c, τ_w

Parameters of the Pope et al. model for the gas, condensate and aqueous phase respectively, generally all will have a value of one although some data in the literature can be fit better using τ as a second parameter.

S_{rg}^{high}, S_{rc}^{high}, S_{rw}^{high}

Gas, condensate and aqueous phase residual saturations in the high capillary number limit, parameters of the Pope at al model. Generally these values are equal to zero.

$k_{rg}^{o\ high}$, $k_{rc}^{o\ high}$, k_{rw}^{ohigh}

End point relative permeability for the gas, condensate and aqueous phase respectively in the high capillary number limit, parameters of the Pope at al model. Generally these values are equal to one.

α_g, α_c, α_w

Exponent used in calculating the $\left[\frac{k_{rl}}{k_{rl}^{o}}\right]^{low}$ term appearing in equation (4) of the Pope et al. model for the gas, condensate and aqueous phases respectively.

m_g,m_c

Parameters for the gas and condensate phases respectively used in the equation to calculate the residual saturation ratio as a function of the capillary number, model2 (no defaults).

n_g,n_c

Parameters for the gas and condensate phases respectively used in calculating the scaling function. The scaling function is equal to the ratio of capillary numbers raised to the power $1/n_l$, model2 (no defaults).

DEFAULTS:

Optional keyword. By default, (i.e. if the *VELDEPRP keyword is not present) capillary number has no effect on relative permeability. The default values for the Pope et al. model are:

$\tau_g, \tau_c, \tau_w = 1.0$

$k_{rg}^{o\ high}$, $k_{rc}^{o\ high}$, $k_{rw}^{ohigh} = 1.0$

S_{rg}^{high}, S_{rc}^{high}, $S_{rw}^{high} = 0.0$

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group. DO NOT use this capillary number model together with the hysteresis model.

EXPLANATION:

The decrease in well productivity for gas condensate reservoirs once the pressure falls below the dew point has been reported in the literature particularly for high gas rate wells. This loss in productivity is caused by the buildup of condensate saturation in the near wellbore region. For near critical fluids cases the interfacial tension (IFT) between the gas and condensate phases is low which often mitigates the decrease in relative permeability due to condensate dropout. However it has been found that in general the relative permeability is a function of not only interfacial tension but fluid velocity as well. A number of models utilizing a dimensionless parameter, the capillary number (CN), which is a ratio of viscous to capillary number on a pore scale, and includes both fluid velocity and IFT have been proposed to correlate laboratory experimental data. The key requirement of such models is that they provide a relationship between the residual saturation and capillary number. All such models require in addition to base relative permeability tables, that is relative permeability as a function of saturation at low capillary number, at least one other parameter per phase obtained by matching experimental data obtained over a range of capillary numbers. In GEM two models proposed in the literature have been implemented. The first is the model by Pope et al. which is essentially a generalized Corey function type approach. This model requires at a minimum one parameter per phase. The second model involves interpolating between base and miscible curves via a scaling function. The scaling function is a function of capillary number. This approach requires a minimum of two parameters per phase. The capillary number has been defined based on the potential gradient of the displacing phase, that is, for condensate, gas is the displacing phase:

$$N_{cc} = \frac{k\nabla\Phi_g}{\sigma} \qquad (1a)$$

For the gas phase, the condensate phase is the displacing phase.

$$N_{cg} = \frac{k\nabla\Phi_c}{\sigma} \qquad (1b)$$

For the aqueous phase, capillary number has been assumed to equal the value for the gas phase.

Where

N_{cc}	=	capillary number condensate, dimensionless
k	=	permeability, m^2
$\nabla\Phi_g$	=	potential gradient for the gas phase, $\Phi = P + \rho_g gh$
σ	=	interfacial tension between gas and oil phases N/m
N_{cg}	=	capillary number gas, dimensionless
$\nabla\Phi_c$	=	potential gradient for the condensate phase, $\Phi = P + \rho_c gh$

For gas, condensate or water phase if the capillary number N_{cl} is less than the entered base capillary number N_{cb} then the relative permeability is equal to the base value. The base relative permeability curves are entered via tables with the relative permeability a function of saturation alone, see *SGT and *SWT keywords.

A brief description of the two models follows:

Pope et al. model:

The residual saturation for a phase l (l = gas or condensate) is calculated based on the capillary number as shown below:

$$S_{lr} = \min\left(S_l, S_{lr}^{high} + \frac{S_{lr}^{low} - S_{lr}^{high}}{1 + T_l (N_{cl})^{\tau_l}} \right) \qquad (2)$$

In the above equation:

S_{lr}	=	residual saturation phase (gas or condensate), dimensionless
S_l	=	saturation of phase l, dimensionless
S_{lr}^{high}	=	residual saturation in the high capillary number limit, usually zero
S_{lr}^{low}	=	residual saturation in the low (base) capillary limit
T_l	=	parameter of the model, obtained by fitting experimental data
N_{cl}	=	capillary number, dimensionless
τ_l	=	parameter of the model, obtained by fitting experimental data, usually equal to one

Mass transfer can reduce the value of S_l to values less than S_{lr} which requires a minimum to be taken in the above equation. The next step is to establish the functional dependence of the end point relative permeability as a function of capillary number and residual saturations:

$$k_{rl}^{o} = k_{rl}^{o\ low} + \frac{S_{l'r}^{low} - S_{l'r}}{S_{l'r}^{low} - S_{l'r}^{high}} \left(k_{rl}^{o\ high} - k_{rl}^{o\ low} \right) \qquad (3)$$

Where $S_{l'r}$ is the residual saturation of the conjugate phase for example condensate is the conjugate phase for gas and:

k_{rl}^{olow}	=	end point relative permeability for the base (low capillary number limit) case
k_{rl}^{ohigh}	=	end point relative permeability in the high capillary number limit, this value is usually equal to zero

The final step is to calculate the relative permeability of each phase as a function of saturation.

$$\log k_{rl} = \log k_{rl}^{o} + \log \bar{S}_l + \frac{\log \left(\frac{k_{rl}}{k_{rl}^{o}} \right)^{low} - \log \bar{S}_l}{1 + T_{l'} (N_{cl'})^{\tau_{l'}}} \qquad (4)$$

Where k_{rl} is the relative permeability to phase l.

In equation (4) the term $\left[\frac{k_{rl}}{k_{rl}^{o}} \right]^{low}$ is equal to $\bar{S}_l^{\alpha_l}$ where αl is a parameter of the model.

The normalized saturation $\bar{S}_l$ in the above equation are defined as follows, where n_p is the number of phases present.

$$\bar{S}_l = \frac{S_l - S_{lr}}{1 - \sum_{1}^{n_p} S_{lr}} \qquad (5)$$

Interpolation between base and 'miscible' curves:

Given a phase l (l =gas, condensate) capillary number, N_{cl}, calculate a normalized capillary number, that is $N_{cnl} = N_{cbl}/N_{cl}$.

As the capillary number rises above the base value the residual saturation S_{lr}^{low}, decreases.

In this model this effect is modelled through a saturation scaling parameter, X_l:

$$X_l = 1 - \exp(-m_l N_{cnl}) \qquad (6)$$

Where m_1 is a parameter of the model, it is determined by matching experimental data and is required to be entered by the user. Given a base or low capillary number residual saturation, S_{lr}^{low}, the effect of the CN dependency is to change the residual saturation $S_{lr} = X_1 \; S_{lr}^{low}$.

The next step is to construct a miscible or straight line relative permeability curve using the following equation:

$$k_{rml} = \frac{\bar{S}_l - X_1 \bar{S}_{lr}^{low}}{1 - X_1 \bar{S}_{lr}^{low}} \qquad (7)$$

$\bar{S}_l$ is the normalized or HCPV saturation of phase 1 or $\bar{S}_l = S_1 / (1-S_w)$.

The CN modified relative permeability curve for phase l is now calculated by interpolating between base and miscible curves using a scaling function:

$$k_{rl} = N_{cnl}^{1/n_l} \; k_{rl}^{low} + \left(1 - N_{cnl}^{1/n_l}\right) k_{rml} \qquad (8)$$

where n_l is a parameter of the model and must be entered by the user.

Single Phase Gas Relative Permeability Flag (Optional)

***KRGAS**

PURPOSE:

*KRGAS indicates that a single-phase gas relative permeability calculation flag follows.

FORMAT:

*KRGAS (*KRG)
(*KRO)

DEFINITIONS:

*KRG

Subkeyword which indicates that the gas relative permeability curve, krg(Sg), is used as krg when only a single hydrocarbon phase is present and that phase is identified as a gas phase. This is the default.

*KRO

Subkeyword which indicates that the oil relative permeability curve, krow(Sw), is assigned to krg when only a single hydrocarbon phase is present and that phase is identified as a gas phase.

DEFAULTS:

Optional keyword. Default is *KRGAS *KRG.

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group. If *KRGAS is entered, exactly one of *KRG or *KRO must be entered. Do not use KRGAS if reservoir is declared as being oil wet.

EXPLANATION:

In early versions of GEM all single phase hydrocarbon relative permeabilities were calculated using the oil relative permeability curve, krow(Sw), irrespective of whether the hydrocarbon phase was identified as a gas or an oil phase. This was done because the oil-water relative permeabilities were interpreted as the wetting/non-wetting curves. The *KRGAS keyword allows the user to continue to adhere to this convention.

Example:

```
*KRGAS  *KRO
```

Method for Evaluating 3-Phase Kro (Optional) *KROIL

PURPOSE:

*KROIL introduces subkeywords which indicate how Kro is evaluated. The options are Stone's first and second models, as modified by Aziz and Settari; the Segregated model; and the Linear Isoperm method. For Stone's models the krow and krog values may be looked up as functions either of Sw and Sg, respectively, or of 1-So and 1-Swr-So.

FORMAT:

```
*KROIL    (*STONE1)          (*SO)
                             (*SWSG)
          (*STONE2)          (*SO)
                             (*SWSG)
          (*LINEAR_ISOPERM)
          (*SEGREGATED)
```

DEFINITIONS:

STONE1

Subkeyword which indicates that Stone's First Model (see below) is to be used to compute Kro.

STONE2

Subkeyword which indicates that Stone's Second Model (see below) is to be used to compute Kro.

LINEAR_ISOPERM

Subkeyword which indicates that the Linear Isoperm Model (see below) is to be used to compute Kro.

SEGREGATED

Subkeyword which indicates that the Segregated Model (see below) is to be used to compute Kro.

SWSG

Subkeyword of STONE1 and STONE2 which indicates that krow is looked up as a function of Sw and krog is looked up as a function of Sg for use in the computation of kro. *STONE2 *SWSG is the default when *KROIL does not appear in the data set. When *KROIL *STONE2 or *KROIL *STONE1 appear with no following subkeyword, the default method is *SWSG. When *KROIL appears with no following subkeyword, the default is *KROIL *STONE2 *SWSG .

SO

Subkeyword of STONE1 and STONE2 which indicates that krow is looked up as a function of 1 - So and krog is looked up as a function 1 - Swr - So for use in the computation of kro with Stone's second model.

DEFAULTS:

Optional keyword. If *KROIL is not encountered in the data set, the default is *KROIL *STONE2 *SWSG. If *KROIL is encountered with no following subkeyword, the default is *KROIL *STONE2 *SWSG . If *KROIL *STONE1 is encountered with no following subkeyword, the default is *KROIL *STONE1 *SWSG. If *KROIL *STONE2 is encountered with no following subkeyword, the default is *KROIL *STONE2 *SWSG.

CONDITIONS:

If it appears, this keyword must be in the ROCK-FLUID DATA keyword group. It can appear anywhere within this group. Do not use this option for the oil wet case.

EXPLANATION:

In STONE'S SECOND MODEL for kro as modified by Aziz and Settari, the oil relative permeability

is computed as

kro = krocw *(((kro/krocw)+ krw)((krog/krocw)+ krg)-krw-krg).

krw and krg are always looked up as functions of Sw and Sg, respectively. krow and krog are read from water-oil and gas-liquid relative permeability tables, respectively. In GEM, krow is tabulated as a function of Sw and krog is tabulated as a function of Sg. If *KROIL *STONE2 *SWSG is in effect and values of Sw, So, and Sg are given, krow is looked up in the table as krow(Sw) and krog is looked up as krog(Sg). When *KROIL *STONE2 *SO is in effect, krow is looked up from the table as krow(1 - So) and krog is looked up as krog(1-Swr-So). *SO corresponds to having krow and krog looked up directly as functions of So.

In some situations, the user may know krow and krog as functions of So rather than of Sw and Sg. GEM requires that the krow and krog curves be entered as functions of Sw and Sg, respectively, but the user may enter krow(Sw = 1-So) as krow(So) and krog(Sg = 1-Swr-So) as krog(So) and specify *KROIL *STONE2 *SO. Here Swr is the connate water saturation for the given *SWT table. The effect is exactly the same as having krog and krow tabulated directly as functions of So.

In STONE'S FIRST MODEL, as modified by Aziz and Settari, kro is computed as

$$\text{kro} = \frac{\text{Sostar} * \text{krow} * \text{krog}}{\text{krocw} * (1 - \text{Swstar}) * (1 - \text{Sgstar})}$$

where

Sostar	=	(So - Som)/(1 - Swcrit - Som)
Swstar	=	(Sw - Swcrit)/(1 - Swcrit - Som)
Sgstar	=	Sg/(1 - Swcrit - Som) ;

for the "minimal" value Som of the oil saturation, GEM uses the linear function of Sg proposed by Fayers and Matthews (SPEJ April 1984, 224-232):

Som(Sg) = (1 - a(Sg)) * Sorw + a(Sg) * Sorg,

where

a(Sg) = Sg / (1 - Swcrit - Sorg).

As in Stone's second model, when the subkeyword *SO is specified, krow(1-So) and krog(1-Swcrit-So) are used in the formula; otherwise, krow(Sw) and krog(Sg) are used.

The LINEAR ISOPERM model is based on a proposal by Baker (paper SPE 17369, 1988). In this method, kro(Sg, Sw) is defined in the region

Swcrit < Sw < 1 - Sorw, 0 < Sg < (1-Sorw-Sw)*(1-Sorg-Swcrit)/(1-Sorw-Swcrit),

by specifying curves (isoperms) along which kro assumes a constant value. In particular, the isoperms are assumed to be straight line segments.

For example, if

kro(Sg=0,Sw) = krow(Sw) = 0.2 for Sw = Sw2

and

kro(Sg, Swcrit) = krog(Sg) = 0.2 for Sg = Sg2,

then all points (Sg, Sw) on the line segment joining (0,Sw2) to (Sg2,Swcrit),

(Sg, Sw) = (0,Sw2) + b * (Sg2,Swcrit-Sw2), 0<b<1,

also have kro(Sg, Sw) = 0.2 .

The *SEGREGATED model corresponds to a model which uses a segregated assumption for the gas and water. So is assumed to be constant throughout the block and the water in the gas zone is assumed to be equal to the connate water saturation. Using volume fraction arguments and some algebraic manipulation, it follows that:

$$\mathrm{kro} = \frac{(\mathrm{Sg} * \mathrm{krog} + (\mathrm{Sw} - \mathrm{Swcon}) * \mathrm{krow})}{(\mathrm{Sg} + \mathrm{Sw} - \mathrm{Swcon})}$$

where krog is the oil relative permeability for a system with oil, gas and connate water (created as in *SWSG but looked up as in *SO), and krow is the oil relative permeability for a system with oil and water only (created as in *SWSW but looked up as in *SO). This model was also proposed by Baker (SPE/DOE Paper 17369).

Examples:

To specify that the linear isoperm method should be used, the user may enter the following line in the ROCK/FLUID DATA section of the data set:

```
*KROIL *LINEAR_ISOPERM
```

To specify that the modified Stone's first model with krow and krog looked up as functions of So, the user may include the following line in the ROCK/FLUID DATA section:

```
*KROIL *STONE1 *SO
```

Saturation Endpoints for each Grid Block (Optional)

***BSWCON, *BSWCRIT, *BSOIRW, *BSORW, *BSGCON, *BSGCRIT, *BSLCON, *BSORG, *BSOIRG**

PURPOSE:

These keywords specify the end points of the relative permeabilities tables of each grid block. The table curves are scaled based on these end points. Use of any of these keywords forces GEM to use new endpoint scaling; see "Notes on Rock Fluid".

ARRAY:

*BSWCON	BS_{wcon}
*BSWCRIT	BS_{wcrit}
*BSOIRW	BS_{oirw}
*BSORW	BS_{orw}
*BSGCON	BS_{gcon}
*BSGCRIT	BS_{gcrit}
*BSLCON	BS_{lcon}
*BSORG	BS_{org}
*BSOIRG	BS_{oirg}

DEFINITIONS:

BS_{wcon}

Connate water saturation of a grid block. BS_{wcon} is the first entry in the *SWT table. 1- BS_{wcon} is the maximum oil saturation possible in the *SWT table.

Oil wet: Connate oil saturation. 1- BS_{wcon} is the maximum possible water saturation in the *SWT table.

BS_{wcrit}

Critical water saturation of a grid block. BS_{wcrit} is the saturation at which water first becomes mobile in the two phase oil-water system. It is always equal to or greater than BS_{wcon}.

Oil wet: Critical oil saturation. It is the saturation at which oil first becomes mobile in the two phase oil-water system. It is always equal to or greater than BS_{wcon}.

BS_{oirw}

Irreducible oil saturation of a grid block. 1- BS_{oirw} is the last water saturation entry in the *SWT table. BS_{oirw} is the minimum oil saturation possible in the *SWT table. 1 BS_{oirw} is the maximum water saturation in the *SWT table.

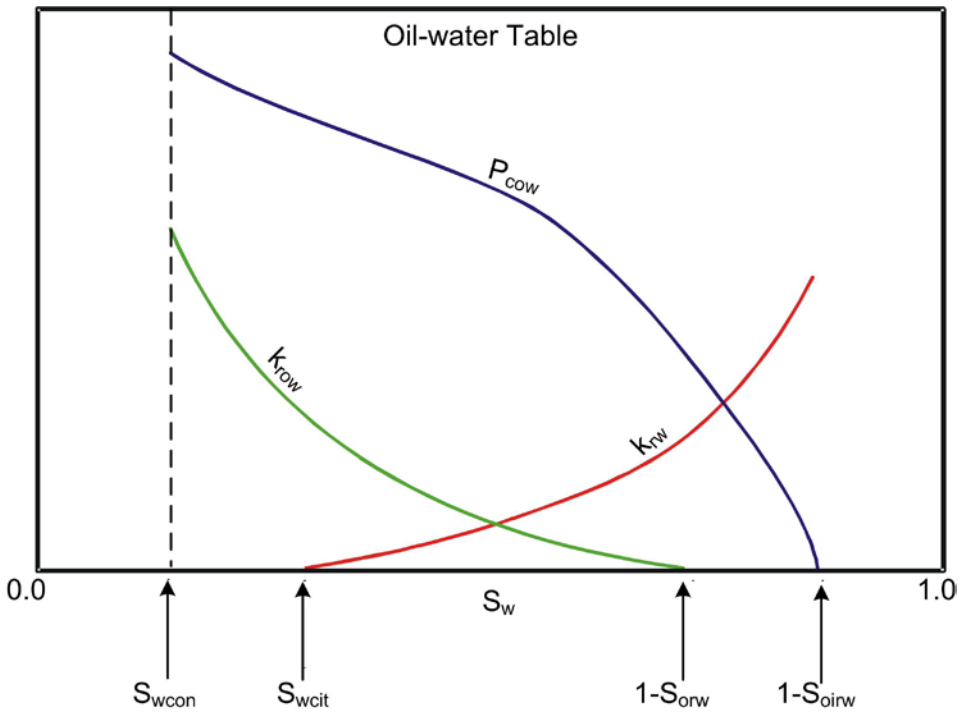
Oil wet: Irreducible (or connate) water saturation of a grid block. 1- BS_{oirw} is the last oil saturation entry in the *SWT table. BS_{oirw} is the minimum water saturation possible in the *SWT table.

BS_{orw}

Residual oil saturation of a grid block. BS_{orw} is the oil saturation at which oil first becomes mobile in the two phase oil-water table. It is always equal to or greater than BS_{oirw}.

Oil wet: Residual water saturation of a grid block. BS_{orw} is the water saturation at which water first becomes mobile in the two phase oil-water table. It is always equal to or greater than $BS_{oirw.}$

Note: Graph shows the water wet case



BS_{gcon}

Connate gas saturation of a grid block. BS_{gcon} is the minimum gas saturation in the two phase gas-liquid system. It is the first entry in a *SGT table and 1- BS_{gcon} is the last entry in a *SLT table.

BS_{gcrit}

Critical gas saturation of a grid block. BS_{gcrit} is the gas saturation at which gas first becomes mobile. It is always equal to or greater than BS_{gcon}.

BS_{lcon}

Connate liquid saturation of a grid block in the gas-liquid table. BS_{lcon} includes connate water (BS_{wcon}). It is the minimum liquid saturation in the two phase gas-liquid system and the first entry in the *SLT table and the last entry in the *SGT table.

Oil wet: Connate liquid saturation of a grid block in the gas liquid table. BS_{lcon} is the sum of connate oil saturation and irreducible water saturation in the two phase gas-liquid (connate oil + water) system. It is the first entry in the *SLT table and the last entry in the *SGT table.

BS_{org}

Residual oil saturation of a grid block in the two phase gas liquid system. It does not include connate water, thus BS_{org} + BS_{wcon} = residual liquid saturation.

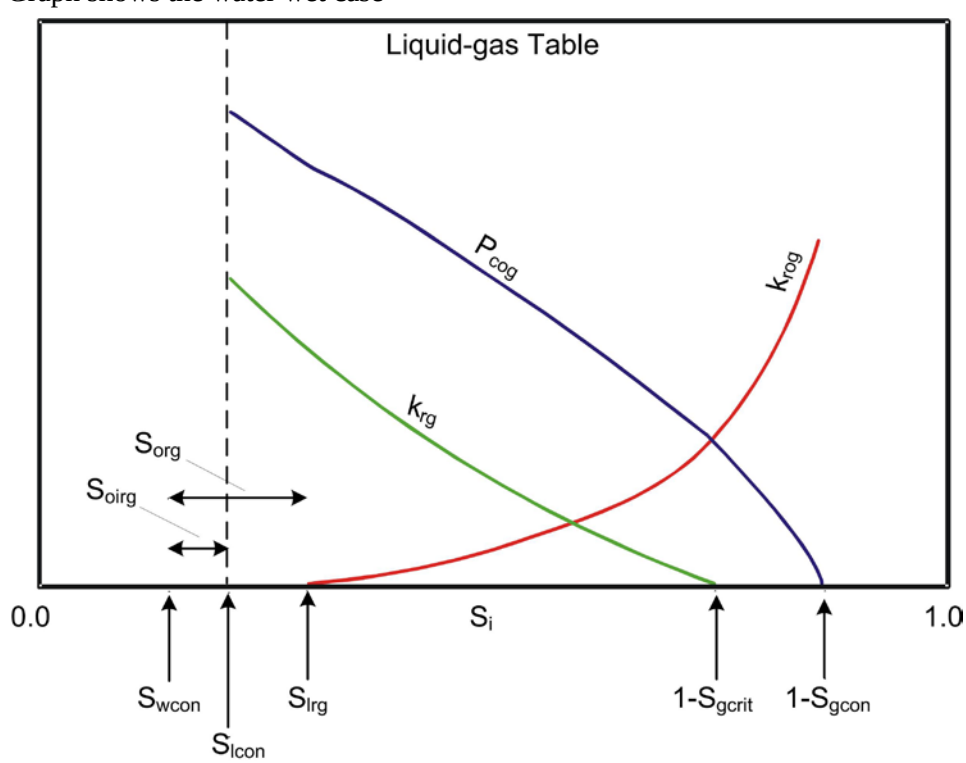
Oil wet: Residual water saturation of a grid block in the two phase gas-liquid (connate oil + water) system.

BS_{oirg}

Irreducible oil saturation in a two phase oil-gas system of a grid block. 1-BS_{oirg} - BS_{wcon} is the last gas saturation entry in the *SGT table. BS_{oirg} + BS_{wcon} is the minimum oil saturation possible in the *SLT table. 1- BS_{oirg} - BS_{wcon} is the maximum gas saturation in the *SGT table.

Oil wet: Irreducible water saturation of a grid block in the two phase water-gas system. Note: Water wet: Specify either the irreducible oil saturation (with BS_{oirg}) or connate liquid with keyword BS_{lcon}. If BS_{oirg} is specified then connate liquid is calculated as the sum of BS_{oirg} + BS_{wcon}. If BS_{lcon} is specified then BS_{oirg} is calculated as BS_{lcon} - BS_{wcon}.

Note: Graph shows the water wet case



where S_{lrg} is the residual liquid in the gas-liquid table.

S_{oirg} is the irreducible oil in the gas liquid table.

$$S_{lrg} = S_{org} + S_{wcon}$$

$$S_{oirg} = S_{lcon} - S_{wcon}$$

DEFAULTS:

Optional keyword. The default is to use the end points from the relative permeability tables corresponding to the *RTYPE region to which the grid block belongs. Any or all of the above keywords may be omitted. In addition, when used in Well and Recurrent Data section it is not necessary to specify the end point values for all the grid blocks. The grid blocks that are not specified will have the default values of the end points (from tables corresponding to the blocks *RTYPE region).

CONDITIONS:

These keywords can appear in the Rock-Fluid Data section and/or in the Well and Recurrent Data section as all of the end points can be altered during the run.

Note: Prior to GEM 2016.10, in place of keywords *BSWCON, *BSWCRIT, *BSOIRW, *BSORW, *BSGCON, *BSGCRIT, *BSLCON, *BSORG, *BSOIRG, keywords *SWCON, *SWCRIT, *SOIRW, *SORW, *SGCON, *SGCRIT, *SLCON, *SORG, *SOIRG respectively were used as block-based saturation end-points. The latter are still acceptable but are deprecated and may not be supported in future releases.

EXPLANATION:

The end point keywords indicate the input of an array of grid values. All array reading option subkeywords are valid. See the section on Array Reading Options for more details.

The relative permeabilities and capillary pressures are looked up on their respective tables after the lookup saturation (either S_w or S_l) have been scaled appropriately.

For the oil-water table and for the water wet case:

$$K_{rw}, S'_w = S'_{wcrit} + (S_w - S_{wcrit}) * \left(\frac{1 - S'_{oirw} - S'_{wcrit}}{1 - S_{oirw} - S_{wcrit}} \right)$$

$$K_{row}, S''_w = S'_{wcon} + (S_w - S_{wcon}) * \left(\frac{1 - S'_{orw} - S'_{wcon}}{1 - S_{orw} - S_{wcon}} \right)$$

$$P_{cow}, S'''_w = S'_{wcon} + (S_w - S_{wcon}) \left(\frac{1 - S'_{oirw} - S'_{wcon}}{1 - S_{oirw} - S_{wcon}} \right)$$

Sw is the blocks water saturation

S'_w, S''_w, S'''_w are the saturations used to interpolate the original user entered tables with respect to K_{rw}, K_{row} and P_{cow} respectively.

S_{wcrit}, S_{wcon}, S_{oirw} and S_{orw} are the user entered grid block end point values (from the keywords of the same name)

$S'_{wcrit}, S'_{wcon}, S'_{oirw}$ and S'_{orw} are the original table end point values (calculated directly from the relative permeability tables)

For the gas-liquid table and for the water wet case:

$$K_{rog},\ S'_l = S'_{lrg} + (S_l - S_{lrg}) * \left(\frac{1 - S'_{lrg} - S'_{gcon}}{1 - S_{lrg} - S_{gcon}} \right)$$

$$K_{rg},\ S''_l = S'_{lcon} + (S_l - S_{lcon}) * \left(\frac{1 - S'_{lcon} - S'_{gcrit}}{1 - S_{lcon} - S_{gcrit}} \right)$$

$$P_{cog},\ S'''_l = S'_{lcon} + (S_l - S_{lcon}) \left(\frac{1 - S'_{lcon} - S'_{gcon}}{1 - S_{lcon} - S_{gcon}} \right)$$

$$S_l = S_o + S_w$$

$$S_{lrg} = S_{wcon} + S_{org}$$

S'_l, S''_l, S'''_l are the saturations used to interpolate the original user entered tables with respect to K_{rog}, K_{rg} and P_{cog} respectively.

S_{org}, S_{lcow}, S_{gcon} and S_{gcrit} are the user entered grid block end point values (from the keywords of the same name)

$S'_{org}, S'_{lcon}, S'_{gcon}$ and S'_{gcrit} are the original table end point values (calculated directly from the relative permeability tables)

Note that each relative permeability curve is now scaled with respect to different end points which define the range of each curve. The table below applies to the water wet case.

K_{rw}	$S_{wcrit} \le S_w \le 1 - S_{oirw}$,	thus K_{rw} is scaled by S_{wcrit} and S_{oirw}
K_{row}	$S_{wcon} \le S_w \le 1 - S_{orw}$,	thus K_{row} is scaled by S_{wcon} and S_{orw}
P_{cow}	$S_{wcon} \le S_w \le 1 - S_{oirw}$,	thus P_{cow} is scaled by S_{wcon} and S_{oirw}
K_{rog}	$S_{lrg} \le S_l \le 1 - S_{gcon}$,	thus K_{rog} is scaled by S_{wcon}, S_{org} and S_{gcon}.
K_{rg}	$S_{lcon} \le S_l \le 1 - S_{gcrit}$,	thus K_{rg} is scaled by S_{lcon} and S_{gcrit}
P_{cog}	$S_{lcon} \le S_l \le 1 - S_{gcon}$,	thus P_{cog} is scaled by S_{lcon} and S_{gcon}

Leverett J Function (Optional) *JFUNC, *SRFTNW, *SRFTNG

PURPOSE:

Replaces water-oil and/or oil-gas capillary pressures entered with Leverett J Functions.

FORMAT:

*JFUNC	(*BOTH)
	(*WATER)
	(*GAS)

ARRAY:

*SRFTNW
*SRFTNG

DEFINITIONS:

*JFUNC

Activate J function option. All capillary pressure entries in relative permeability tables are now interpreted as J Functions. All rock types are effected by a single *JFUNC keyword.

*BOTH

Both water-oil and oil-gas capillary pressures are interpreted as J functions for all rock types defined (Default).

*WATER

Only the water-oil capillary pressures are interpreted as J functions.

*GAS

Only the oil-gas capillary pressures are interpreted as J functions.

*SRFTNW

Defines water-oil surface tension of each grid block to be used in the J Function Option. In the calculation of P_{cow} when using the J function option, the capillary pressure is a function of J function, porosity, permeability and surface tension (or surface tension * Cosine (Contact Angle)). This grid block array can be used to enter a different surface tension term for each block in the reservoir (dyne/cm).

*SRFTNG

Defines oil-gas surface tension of each grid block to be used in the J Function Option. In the calculation of P_{cog} when using the J function option, the capillary pressure is a function of J function, porosity, permeability and surface tension (or surface tension * Cosine (Contact Angle)). This grid block array can be used to enter a different surface tension term for each block in the reservoir (dyne/cm).

DEFAULTS:

Optional keyword, capillary pressure table entries are interpreted as capillary pressures unless *JFUNC is encountered, If *JFUNC is found, some capillary pressure tables (based on subkeywords) may be read in as J functions. If *JFUNC is encountered without an optional subkeyword, *JFUNC *BOTH is assumed.

Surface tension arrays *SRFTNW and *SRFTNG are used only when the J function option is active. When the J function option is in use, the default values are 1.0. This implies that the surface tension effects are constant with grid block and are already multiplied into the J function terms entered in the *SWT, *SGT, and *SLT tables.

CONDITIONS:

This keyword must be in the Rock-Fluid Data section.

EXPLANATION:

The J function option allows the user to scale capillary pressure to account for differences in block porosity and permeability. When used with the *SRFTNW and *SRFTNG keywords to enter individual block surface tension terms (water-oil surface tension using *SRFTNW and oil-gas surface tension using *SRFTNG), the user can additionally scale capillary pressure based on block location. When enabled, J functions replace either or all capillary pressures in the relative permeability table reading options (*SWT, *SGT, *SLT).

The effect of block dependent porosity, permeability and surface tension is accounted for in the Gravity Equilibrium initialization options and will produce spatially varying transition zones which are a result of spatially varying capillary pressures.

Defining:

ϕ	Block Porosity at Reference Pressure
K	Block Permeability (md), normally $\sqrt{K_x * K_y}$ unless $\sqrt{K_x * K_y}$ is zero, then max(K_x, K_y, K_z) is used
$J_{(SW)}$	Dimensionless J function entered in the *SWT table. This replaces the capillary pressure column for Pcow
$J_{(Sg)}$ or $J_{(Sl)}$	Dimensionless J function entered in the *SGT or *SLT table. This replaces the capillary pressure column for Pcog
*SRFTNW	Oil-water Surface tension or Surface tension * Cosine (Contact Angle) (dyne/cm)
*SRFTNG	Oil-gas Surface tension or Surface tension * Cosine (Contact Angle) (dyne/cm)
CF	Conversion Factor, for Capillary pressure in kPa = 31.8316 for Capillary pressure in Psi = 4.61678

Then the true capillary pressure for the oil-water table is:

$$P_{cow} = J_{(Sw)} * SRFTNW * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

Similarly for the oil-gas table:

$$P_{cog} = J_{(Sg)} * SRFTNG * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

or

$$P_{cog} = J_{(Sl)} * SRFTNG * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

Maximum Relative Permeability/Capillary Pressure Values for Each Grid Block (Optional) *BKRWIRO, *BKROCW, *BKRGCL, *BKROGCG, *BKRWRO, *BKROCRW, *BKRGRL, *BKROGCRG, *PCWMAX, *PCGMAX, *JFWMAX, *JFGMAX

PURPOSE:

These keywords specify the end point values of the relative permeabilities tables of each grid block. The table curves are scaled based on these end points. Use of any of these keywords forces GEM to use new endpoint scaling; see "Notes on Rock Fluid".

ARRAY:

*BKRWIRO	Bk_{rwiro}
*BKROCW	Bk_{rocw}
*BKRGCL	Bk_{rgcl}
*BKROGCG	Bk_{rogcg}
*BKRWRO	Bk_{rwro}
*BKROCRW	Bk_{rocrw}
*BKRGRL	Bk_{rgrl}
*BKROGCRG	Bk_{rogcrg}
*PCWMAX | *JFWMAX	PCWMAX | JFWMAX
*PCGMAX | *JFGMAX	PCGMAX | JFGMAX

DEFINITIONS:

Bk_{rwiro}

Relative permeability value of water at the irreducible oil saturation in the oil-water table (the last row in the table).

Oil wet: Relative permeability value of oil at the irreducible water saturation in the oil water table (last row in the *SWT table)

Bk_{rocw}

Relative permeability value of oil at the connate water saturation in the oil-water table (the first row in the table).

Oil wet: Relative permeability value of water at the connate oil saturation in the oil water table (the first row in the table)

Bk_{rgcl}

Relative permeability value of gas at the connate liquid saturation in the gas-liquid table (the first row in the table if SLT is used or the last row in the table if SGT is used).

Bk_{rogcg}

Relative permeability value of oil at the minimum gas saturation in the gas-liquid table (the last row in the table if SLT is used or the first row in the table if SGT is used). Oil wet: Relative permeability value of water at the

minimum gas saturation in the gas-liquid table (the last row in the table if SLT is used or the first row in the table if SGT is used).

Critical Endpoint Values

Bk_{rwro}

Relative permeability value of water at the residual oil saturation (Sorw) in the oil-water table.

Bk_{rocrw}

Relative permeability value of oil at the critical water saturation (Swcrit) in the oil-water table.

Bk_{rgrl}

Relative permeability value of gas at the residual liquid saturation (S_{lrg}) in the gas-liquid table (See *BSWCON Keyword for definition of S_{lrg}).

Bk_{rogcrg}

Relative permeability value of oil at the critical gas saturation (Sgcrit) in the gas-liquid table.

*PCWMAX

Oil-water capillary pressure value (kPa | psi) at the connate water saturation in the water -oil table (the first row in the table).

Oil wet: water-oil capillary pressure value at connate oil saturation in the oil-water table (first row in the table – For oil wet Pcow = Pw-Po). Do not use if the Leverett J Function option (*JFUNC *WATER / *BOTH) is specified.

*PCGMAX

Oil-gas capillary pressure value (kPa | psi) at the connate liquid saturation in the gas-liquid table. Do not use if the Leverett J Function option (*JFUNC *GAS / *BOTH) is specified.

*JFWMAX

Oil-water J Function value (dimensionless) at the connate water saturation in the water-oil table.

For oil wet water-oil J Function value at the connate oil saturation in the oil-water table. Only use if the J Function option (*JFUNC *WATER / *BOTH) is specified.

*JFGMAX

Oil-gas J Function value (dimensionless) at the connate liquid saturation in the gas-liquid table. Only use if the J Function option (*JFUNC *GAS / *BOTH) is specified.

DEFAULTS:

Optional keyword. The default is to use the values from the relative permeability tables corresponding to the *RTYPE region to which the grid block belongs. Any or all of the above keywords may be omitted.

The default relative permeability value scaling method is two-point scaling. If a critical Kr endpoint is assigned to a block, the relative permeability value will be scaled using three-point scaling; provided that the critical endpoint in the table differs from the connate value in the table or the critical grid block end point saturation is different from the connate grid block saturation value. Otherwise two-point scaling will be performed.

See Tutorial 'Two and Three Point Scaling of Relative Permeability Values' for definitions of two-point and three-point relative permeability values.

CONDITIONS:

These keywords can appear in the Rock-Fluid Data section and/or in the Well and Recurrent Data section as all of the end points can be altered during the run.

Values of a keyword can be assigned to a portion of the reservoir blocks. For maximum endpoint values, the unassigned blocks will be filled with default values from the table end points or from tables, in that order.

Note: Prior to GEM 2016.10, in place of keywords *BKRWIRO, *BKROCW, *BKRGCL, *BKROGCG, *BKRWRO, *BKROCRW, *BKRGRL, *BKROGCRG, keywords *KRWIRO, *KROCW, *KRGCL, *KROGCG, *KRWRO, *KROCRW, *KRGRL, *KROGCRG respectively were used as block-based saturation end-points. The latter are still accepted but are deprecated and may not be supported in future releases.

EXPLANATION:

The end point keywords indicate the input of an array of grid values. All array reading option sub-keywords are valid. See the section on Array Reading Options for more details.

For the critical endpoint value scaling to be enabled it is crucial that either the connate/irreducible endpoint saturations <u>in the table</u> are different from the critical/residual endpoint saturations or the grid block connate/irreducible endpoint saturations are different from the grid block critical/residual saturations.

That is: $S_{orw} > S_{oirw}$, $S_{wcrit} > S_{wcon}$, $S_{lrg} > S_{lcon}$, and $S_{gcrit} > S_{gcon}$ in the entered table (via keyword *SWT/*SGT/*SLT) for a particular grid block or the grid block end point values are different (entered via keywords *SWCON, *SWCRIT etc.).

If the above is not true for a particular curve, three-point Kr value scaling will not be performed for that curve.

If the *3PTSCALING keyword is entered then 3 point horizontal scaling is done for all four relative permeability curves: k_{rw}, k_{row}, k_{rg} and k_{rog}. If the *3PTSCALING keyword is NOT specified then the 3 point horizontal scaling will be done for a given curve if either the keyword to turn on 3 point horizontal scaling is specified (*3PTSCAL_KRW etc.) or the corresponding 3 point vertical scaling keyword is specified. For example if *KRWRO is specified then 3 point horizontal scaling is done for the water phase or the k_{rw} curve.

These values override the values of relative permeability and capillary pressures at the endpoints. Once defined these values remain in effect until explicitly overridden by another instance of these keywords. Changing a block's *RTYPE will not change a blocks endpoint if it has been overridden by these keywords (see *RTYPE Keyword). Without the use of these keywords, if a blocks *RTYPE is changed, its endpoints will be changed as well.

Water-Oil Capillary Pressure (J Function) Shift for Each Grid Block (Optional) *PCOW_SHIFT, *JFW_SHIFT

PURPOSE:

These keywords specify the amount that the water-oil capillary pressure or water-oil J Function curve is shifted up or down. These keywords can be used to account for the effects of a tilted water-oil contact.

ARRAY:

*PCOW_SHIFT | *JFW _SHIFT

DEFINITIONS:

*PCOW_SHIFT

*PCOW_SHIFT specifies the amount that a grid block oil -water capillary pressure for water wet (= Po-Pw) or water – oil capillary pressure (= Pw – Pc) for oil wet, curve is shifted up or down from the table value (kPa | psi). Do not use if the J Function option is specified.

*JFW_SHIFT

*JFW_SHIFT specifies the amount that a grid block oil- water J Function curve is shifted up or down from the table value (dimensionless). Only use if the J Function option is specified. For oil wet specifies the amount that a grid block water-oil J function curve is shifted up or down from the table value.

DEFAULTS:

Optional keyword. The default is zero which means no shifting.

CONDITIONS:

This keyword must be in the Rock-Fluid Data section. Any or all of the above keywords may be omitted. If any of the above keywords are used, values for all grid blocks must be specified.

EXPLANATION:

The *PCOW_SHIFT and *JFW_SHIFT keywords indicate the input of an array of grid values. All array reading option subkeywords are valid. See the section on Array Reading Options for more details. These values are added to the value of Pcow or J Functions. The shift is applied after the maximum capillary pressure or J Function of a block has been set. For example, if a maximum Pcow has been defined using *PCWMAX and a Pcow shift has been defined using *PCOW_SHIFT, the actual block maximum is the sum of the two. *PCOW_SHIFT or *JFW_SHIFT can be used to model tilting water-oil contact.

Cation Exchange Capacity for Each Grid Block (Optional)

***CEC-IEX**

PURPOSE:

This keyword signals input of cation-exchange-capacity (CEC) for each grid-block.

ARRAY:

*CEC-IEX

DEFINITIONS:

*CEC-IEX

*CEC-IEX specifies the amount of ion that can be adsorbed on the exchanger surface. It is measured in eq/m^3, where $eq/m^3 = (mol/m^3) \times$ (charge of ion), irrespective of system of units.

DEFAULTS:

Optional keyword. The default is zero which means no cation-exchange.

CONDITIONS:

This keyword must be in the Rock-Fluid Data section. If the above keywords are used, values for all grid blocks must be specified.

EXPLANATION:

The *CEC-IEX keyword indicates the input of an array of grid values. All array reading option sub-keywords are valid. See the section on Array Reading Options for more details.

Either *CEC-IEX or *CEC-CALC should be present in ion-exchange data sets. A *CEC-IEX value of zero is considered to be a valid input which will mean that no ion-exchange would take place.

Cation Exchange Capacity through Correlation (Optional)

***CEC-CALC**

PURPOSE:

This keyword signals use of a porosity based correlation to calculate cation exchange capacity (CEC) for each grid-block.

FORMAT:

*CEC-CALC (cec_par0 cec_par1)

DEFINITIONS:

*CEC-CALC

*CEC-CALC indicates that an internal correlation based on the grid-block porosity will be used to calculate CEC for each grid block.

cec_par0, cec_par1

Optional matching parameters to adjust the calculated value of CEC.

DEFAULTS:

Optional keyword. The default is not to use the internal correlation for calculation of CEC. If *CEC-CALC is present without the two numbers following it, it is assumed that cec_par0 = 0.0 and cec_par1 = 1.0.

CONDITIONS:

This keyword must be in the Rock-Fluid Data section. If keywords *CEC-IEX and *CEC-CALC both are present in the data set, *CEC-CALC gets precedence; keyword *CEC-IEX and related input will be ignored and a warning message will be issued to that effect.

EXPLANATION:

If *CEC-CALC keyword is present following relationship is used to compute CEC for each grid block.

$$\mathrm{CEC}_0\left(\frac{\mathrm{eq}}{\mathrm{m}^3}\right) = \frac{10^{-3}\rho_r\left(1-\phi_e\right)}{\phi_e}\left[625.8(\mathrm{fraction_of_clay}) + 48.863\right] \qquad (1)$$

where, ρ_r is rock density in kg/cm^2, and ϕ_e is initial effective porosity of the grid block. The fraction_of_clay is assumed to be equal to ϕ_{max} - ϕ_e where ϕ_{max} is the maximum of porosity values entered for all the grid-blocks. Normally CEC is correlated with fraction of clay and fraction of organic content, as in Breeuwsma et al. (1986). Correlation (1) was obtained by matching Breeuwsma et al. CEC for 15 samples where experimental values of organic content and fraction of clay were available. The match from correlation (1) was within 8 % of the Breeuwsma et al. calculated values.

Finally CEC is calculated by applying adjusting parameters cec_par0 and cec_par1,

$$\mathrm{CEC}\left(\frac{\mathrm{eq}}{\mathrm{m}^3}\right) = \mathrm{cec_par\,0} + (\,\mathrm{cec_par\,1} \times \mathrm{CEC}_0) \qquad (2)$$

Either *CEC-IEX or *CEC-CALC must be present is an ion-exchange data set.

Reference:

Breeuwsma, A., Wosten, J. H. M., Vleeshouwer, J. J., Van slobbe, A. M., and Bouma, J. (1986). Derivation of land qualities to assess environmental problems from soil surveys. Soil Sci Soc. Am. J., 50, 186-190.

Multicomponent Adsorption from the Gas Phase – Langmuir Adsorption Input (Optional) *ADGCSTC, *ADGMAXC

PURPOSE:

These array keywords control the modelling of the adsorption of component from the gas phase onto the reservoir rock.

ARRAYS:

*ADGCSTC	'comp_name'	Array qualifier	B_i
*ADGMAXC	'comp_name'	Array qualifier	$\omega_{i,max}$

DEFINITIONS:

*ADGCSTC

Introduces the inverse-pressure parameter for the Langmuir isotherm model (1/kPa | 1/psi).

*ADGMAXC

Specifies the maximum moles of adsorbed component per unit mass of rock (gmole of component/kg of rock/lb of rock).

DEFAULTS:

Optional keyword. The default is to have no adsorption.

CONDITIONS:

These keywords must be in the ROCK-FLUID DATA group. If any of the two keywords appear, then they must both appear. Adsorption of components from gas phase is not allowed in presence of adsorption of components from aqueous phase.

EXPLANATION:

These keywords accept any of the array reading options detailed earlier. However, *ADGCSTC and *ADGMAXC must also have a component name, in single quotes, as an additional array reading qualifier. If one of these keywords appears with a certain component name, then both must appear with that name. Non-adsorbing components do not need to be mentioned.

The extended Langmuir isotherm for multicomponent adsorption is (Arri et al., 1992; Hall et al., 1994):

$$\omega_i = \frac{\omega_{i,max} B_i y_{ig} p}{1 + p \sum_j B_j y_{jg}}$$

where

B_i	=	parameter for Langmuir isotherm relation
ω_i	=	moles of adsorbed component per unit mass or rock
$\omega_{i,max}$	=	maximum moles of adsorbed component i per unit mass of rock

p	=	pressure
y_{ig}	=	molar fraction of adsorbed component i in the gas phase

where the above sum runs over the adsorbed components. Normally, only the adsorption of light components (e.g. C_1, C_2, N_2, CO_2) is significant.

Both B_i and ω_i are parameters of the Langmuir isotherm for Component i.

The product y_{ig}*p represents the partial pressure of component i in the gas phase.

See also related keyword *ROCKDEN.

Multicomponent Adsorption from the Gas Phase – Tabular Adsorption Data Entry (Optional) *ADSORBTMAX, *ADSTAB

PURPOSE:

These keywords are provided to enable adsorption data to be entered as a table for each adsorbing component. This data is typically obtained via laboratory measurements with core samples. In GEM this data is used to predict multicomponent adsorption using a mixing rule. The mixing rule is based on the extended multicomponent Langmuir model. *ROCKDEN must appear as described above for *ADGCSTC and *ADGMAXC.

FORMAT:

*ADSORBTMAX comp_name ω_{max}
*ADSTAB 'comp_name'

DEFINITIONS:

*ADSORBTMAX

Identifies whether a given component adsorbs or not. If a component adsorbs, enter a line using *ADSORBTMAX followed by its component name in single quotes (as entered earlier using the *COMPNAME keyword). Follow this by a number ω_{max} indicating the maximum moles of adsorbed component per unit mass of rock (gmole of component/kg of rock/lb of rock). A three component model in which methane and carbon dioxide adsorb might look as follows:

* NC 3

* COMPNAME 'CH4' 'N2' 'CO2'

...

* ADSORBTMAX 'CH4' $\omega_{ch4,max}$

* ADSORBTMAX 'CO2' $\omega_{co2,max}$

where $\omega_{ch4,max}$ is the maximum adsorption value for methane and $\omega_{co2,max}$ is the maximum adsorption value for carbon dioxide.

These maximums will be used to normalize the adsorption data specified with the *ADSTAB keyword. The maximum adsorption values are also used for validating data entered with the *ADSTAB keyword i.e. all adsorption data for a given component are required to be smaller than the corresponding maximums entered using ADSORBTMAX.

*ADSTAB

This keyword should be followed by a component name (in single quotes) and then a two column table showing the amount of component ‘component_name’ adsorbing (gmole of component/[kg | lb] of rock) as a function of partial pressure of that component. The first column is the partial pressure (kPa | psia) and the second column the corresponding adsorption. The adsorption values must all be less than the maximum value for that

component entered with the *ADSORBTMAX keyword. Returning to the above example, the *ADSTAB keyword could appear as:

* ADSTAB 'CH4'

**(partial pressure CH4) (adsorption of CH4)

$\bar{p}_{1,1}$ $\omega_{1,1}$

$\bar{p}_{1,2}$ $\omega_{1,2}$

$\vdots$ $\vdots$

$\bar{p}_{1,m}$ $\omega_{1,m}$

* ADSTAB 'CO2'

**(partial pressure CO2) (adsorption of CO2)

$\bar{p}_{s,1}$ $\omega_{s,1}$

$\bar{p}_{s,2}$ $\omega_{s,2}$

$\vdots$ $\vdots$

$\bar{p}_{s,m}$ $\omega_{s,m}$

*ADSTAB *BINARYCOMP ‘comp_name’ *MOLE_FRACTION xmf

This format applies specifically to the case where there are exactly two adsorbing components and adsorption data is available as a function of total pressure and composition of the gas phase for these components. First enter two column tables showing the amount of the indicated component that is adsorbed (gmole of component/(kg | lb) of rock), as a function of total pressure, when that component has the indicated mole fraction (xmf) in the gas phase. Thus, the first column in the tables represents the total pressure (kPa | psia) and the second column records the corresponding adsorption. These adsorption values must all be less than the maximum value for that component entered with the *ADSORBTMAX keyword. Then enter a table for the next largest mole fraction value, and continue until a table has been entered for the highest mole fraction. When all the tables for the first adsorbing component have been entered, enter all the tables for the second component in the (binary) pair. The first component should appear before the second in the list entered earlier using the *COMPNAME keyword.

Returning to our example the *ADSTAB keyword formats for binary adsorption with adsorption as a function of total pressure and composition would be:

$*\,ADSTAB\ *\,BINARYCOMP\ \ 'CH4'\ \ *\,MOLE_FRACTION\ x_{1,1}$

$**(Component\ **CH4,\ mole\ fraction\ of\ CH4 = x_{1,1})$

$**Total\ Pressure\quad **\,Adsorption$

$p_{1,1}\qquad \omega_{1,1}$

$p_{1,2}\qquad \omega_{1,2}$

$\vdots\qquad\quad \vdots$

$p_{1,m}\qquad \omega_{1,m}$

$*\,ADSTAB\ *\,BINARYCOMP\ \ 'CH4'\ \ *\,MOLE_FRACTION\ x_{1,2}$

$**(Component\ **CH4,\ mole\ fraction\ of\ CH4 = x_{1,2})$

$**Total\ Pressure\quad **\,Adsorption$

$p_{1,1}\qquad \omega_{1,1}$

$p_{1,2}\qquad \omega_{1,2}$

$\vdots\qquad\quad \vdots$

$p_{1,m}\qquad \omega_{1,m}$

$*\,ADSTAB\ *\,BINARYCOMP\ \ 'C02'\ \ *\,MOLE_FRACTION\ x_{2,1}$

$**(Component\ **C02,\ mole\ fraction\ of\ C02 = x_{2,1})$

$**Total\ Pressure\quad **\,Adsorption$

$p_{2,1}\qquad \omega_{2,1}$

$p_{2,2}\qquad \omega_{2,2}$

$\vdots\qquad\quad \vdots$

$p_{2,m}\qquad \omega_{2,m}$

$*\,ADSTAB\ *\,BINARYCOMP\ \ 'C02'\ \ *\,MOLE_FRACTION\ x_{2,2}$

$**(Component\ **C02,\ mole\ fraction\ of\ C02 = x_{2,2})$

$**Total\ Pressure\quad **\,Adsorption$

$p_{2,1}\qquad \omega_{2,1}$

$p_{2,2}\qquad \omega_{2,2}$

$\vdots\qquad\quad \vdots$

$p_{2,m}\qquad \omega_{2,m}$

DEFAULTS:

Optional keywords. The default is to have no adsorption.

CONDITIONS:

These keywords must be in the ROCK-FLUID DATA group and ABSOBTMAX should appear before ADSTAB.

The analytical Langmuir parameter entry and the tabular data entry options cannot be used together in a single data set. For example it is not permitted to enter analytical parameters for adsorbing component 1 and tabular data for adsorbing component 2. Likewise do not enter analytical parameters for one rock type and tabular data for another rock type. It is also required that the list of adsorbing components not change from one rock type to another.

EXPLANATION:

The extended Langmuir isotherm for multicomponent adsorption is (Arri et al., 1992; Hall et al., 1994):

$$\omega_i = \frac{\omega_{i,max} B_i y_{ig} p}{1 + p \sum_j B_j y_{jg}} \qquad (1)$$

where

B_i	=	parameter for Langmuir isotherm relation
ω_i	=	moles of adsorbed component per unit mass or rock
$\omega_{i,max}$	=	maximum moles of adsorbed component i per unit mass of rock
p	=	pressure
y_{ig}	=	molar fraction of adsorbed component i in the gas phase

where the above sum runs over the adsorbed components. Normally, only the adsorption of light components (e.g. C_1, C_2, N_2, CO_2) is significant. However, the component name could be any of the component names defined via *COMPNAME.

Both B_i and ω_i are parameters of the Langmuir isotherm for Component i.

The product y_{ig}*p represents the partial pressure of component i in the gas phase. Given pure component adsorption data, a mixing rule is required to predict adsorption rates for each component in a multicomponent mixture. The form of the multicomponent Langmuir model (Equation 1) suggests a potential mixing rule. Consider the following expression for a mixture where the summation is over all adsorbing components:

$$\omega_i / \omega_{i,max} = \chi_i = \frac{\Gamma_i(\bar{p}_i)}{1 + \sum_j \Gamma_j(\bar{p}_j)} \qquad (2)$$

where $\bar{p}_i$ is the partial pressure of component i, i.e.

$$\bar{p}_i = y_{ig} \cdot p \qquad (3)$$

For a single adsorbing species, Equation (2) can be rearranged to give:

$$\Gamma_i(p) = \frac{\chi_i}{1 - \chi_i} \qquad (4)$$

Equation (2) is the mixing rule used in this project to obtain adsorption values for adsorbing components in a mixture where only pure component adsorption data are available. Equation (2) meets some important criteria: (i) it is a derived based on a fairly acceptable physical model, i.e. the Langmuir model; (ii) it reproduces the actual entered laboratory data in the case where there is only one absorbing component. Equation (2) is expected to perform especially well for cases where a single component dominates the adsorption.

In terms of data entry requirements, the user enters tables of adsorption values, ω_i, versus partial pressure, $y_{ig}p$, for each component as well as a value for the maximum adsorption for each component, $\omega_{i,max}$. The adsorption, ω_i is used to calculate Γ_i at each partial pressure from Equation (4). The converted table is stored internally as Γ_i versus partial pressure for each adsorbing component.

To estimate adsorption in a multicomponent mixture of composition y_{ig}, the partial pressure for each component, $y_{ip}p$ is first calculated. A table look up is then performed for each component to determine Γ_i. Equation (2) is subsequently applied to calculate ω_i from Γ_i and $\omega_{i,max}$. Linear interpolation is used for partial pressures between table entries. For pressures outside the table range, the first or last entry in the table is selected.

Multicomponent Adsorption from the Aqueous Phase – Langmuir Adsorption Input (Optional) *ADACSTC, *ADAMAXC

PURPOSE:

These array keywords control the modeling of the adsorption of component from the aqueous phase onto the reservoir rock.

ARRAYS:

*ADACSTC	'comp_name'	Array qualifier	B_i
*ADAMAXC	'comp_name'	Array qualifier	$\omega_{i,max}$

DEFINITIONS:

*ADACSTC

Introduces the inverse-mole fraction parameter, B_i for the Langmuir isotherm model, where B_i should be expressed in units of (1/mole-fraction).

*ADAMAXC

Specifies $\omega_{i,max}$ the maximum moles of adsorbed component per unit mass of rock (gmole of component / (kg | lb) of rock).

DEFAULTS:

Optional keywords. The default is to have no adsorption of components from the aqueous phase.

CONDITIONS:

These keywords must be in the ROCK-FLUID DATA group. If any of the two keywords appear, then they must both appear.

EXPLANATION:

These keywords accept any of the array reading options detailed earlier. However, *ADACSTC (B_i) and *ADAMAXC ($\omega_{i,max}$) must also have a component name, in single quotes, as an additional array reading qualifier. The component name is any of the component names specified via *COMPNAME or *COMPNAME-AQUEOUS that could potentially be present in the aqueous phase. If one of these keywords appears with a certain component name, then both must appear with that name. Non-adsorbing components do not need to be mentioned.

The extended Langmuir isotherm for multicomponent adsorption is (Arri et al., 1992; Hall et al., 1994):

$$\omega_i = \frac{\omega_{i,max} B_i y_{i_aqu}}{1 + \sum_j B_j y_{j_aqu}}$$

where

B_i	=	parameter for Langmuir isotherm relation
ω_i	=	moles of adsorbed component per unit mass or rock

$\omega_{i,max}$ = maximum moles of adsorbed component i per unit mass of rock

y_{i_aqu} = molar fraction of adsorbed component i in the aqueous phase

and the above sum runs over the adsorbed components.

Both B_i and $\omega_{i,max}$ are parameters of the Langmuir isotherm for Component i.

In case the model has aqueous components involving chemical equilibrium reactions, the adsorption/desorption is allowed only for the primary components, i.e. for (number of aqueous components – number of chemical equilibrium reactions).

Multicomponent Adsorption from the Aqueous Phase – Tabular Adsorption Data Entry (Optional)

***ADSORBTMAXA, *ADSTABA**

PURPOSE:

These keywords are provided to enable adsorption data to be entered as a table for each adsorbing component from the aqueous phase. This data is typically obtained via laboratory measurements with core samples. In GEM this data is used to predict multicomponent adsorption using a mixing rule. The mixing rule is based on the extended multicomponent Langmuir model. Keyword *ROCKDEN must appear as described separately.

FORMAT:

*ADSORBTMAXA comp_name $\omega_{i,max}$
*ADSTABA 'com_name'

DEFINITIONS:

*ADSORBTMAXA

If a component adsorbs, enter a line using *ADSORBTMAXA followed by its component name in single quotes (as entered earlier using the *COMPNAME or *COMPNAME-AQUEOUS keyword). Follow this by a number ω_{max} indicating the maximum moles of adsorbed component per unit mass of rock (gmole of component / (kg | lb) of rock). A two aqueous component model in which 'Polymer' adsorbs might look as follows:

*NC– AQUEOUS 2

*COMPNAME – AQUEOUS 'Polymer' 'NaCl'

...

*ADSORBTMAX 'Polymer' $\omega_{Polymer,max}$

where $\omega_{Polymer,max}$ is the maximum adsorption value for component 'Polymer'.

This maximum value will be used to normalize the adsorption data specified with the *ADSTABA keyword. The maximum adsorption values are also used for validating data entered with the *ADSTABA keyword i.e., all adsorption data for a given component are required to be smaller than the corresponding maximums entered using ADSORBTMAXA.

*ADSTABA

This keyword should be followed by a component name (in single quotes) and then on the next line onwards a two column table showing the amount of component 'comp_name' adsorbing (gmole of component / (kg | lb) of rock) as a function of the mole fraction of that component. The first column is the mole fraction in aqueous phase and the second column the corresponding adsorption. The adsorption values must all be less than the maximum value

for that component entered with the *ADSORBTMAXA keyword.
Returning to the above example, the *ADSTABA keyword could appear as:

*ADSTABA 'Polymer'

**(mole–fraction Polymer) (Adsorption of Polymer)

$y_{1,1}$	$\omega_{1,1}$
$y_{1,2}$	$\omega_{1,2}$
$\vdots$	$\vdots$
$y_{1,m}$	$\omega_{1,m}$

DEFAULTS:

Optional keywords. The default is to have no adsorption.

CONDITIONS:

These keywords must be in the ROCK-FLUID DATA group and ABSORBTMAXA should appear before ADSTABA.

The analytical Langmuir parameter entry and the tabular data entry options cannot be used together in a single data set. For example it is not permitted to enter analytical parameters for adsorbing component 1 and tabular data for adsorbing component 2. Likewise do not enter analytical parameters for one rock type and tabular data for another rock type. It is also required that the list of adsorbing components not change from one rock type to another.

EXPLANATION:

The extended Langmuir isotherm for multicomponent adsorption is (Arri et al., 1992; Hall et al., 1994):

$$\omega_i = \frac{\omega_{i,max} B_i y_{i_aqu}}{1 + \sum_j B_j y_{j_aqu}} \qquad (1)$$

where

B_i	=	parameter for Langmuir isotherm relation
ω_i	=	moles of adsorbed component per unit mass or rock
$\omega_{i,max}$	=	maximum moles of adsorbed component i per unit mass of rock
$y_{i\ aqu}$	=	mole fraction of adsorbed component i in the aqueous phase

where the above sum runs over the adsorbed components. Both B_i and ω_i are parameters of the Langmuir isotherm for component i. The component name is any of the component names defined via *COMPNAME or *COMPNAME-AQUEOUS that could potentially be present aqueous phase.

The y_{i_aqu} represents the mole fraction of component i in the aqueous phase. Given pure component adsorption data, a mixing rule is required to predict adsorption rates for each component in a multicomponent mixture. The form of the multicomponent Langmuir model (Equation 1) suggests a potential mixing rule. Consider the following expression for a mixture where the summation is over all adsorbing components:

$$\omega_i/\omega_{i,max} = \chi_i = \frac{\Gamma_i\left(y_{i_aqu}\right)}{1 + \sum_j \Gamma_j\left(y_{j_aqu}\right)} \qquad (2)$$

where y_{i_aqu} is the mole fraction of component i.

For a single adsorbing species, Equation (2) can be rearranged to give:

$$\Gamma_i\left(y_{i_aqu}\right) = \frac{\chi_i}{1-\chi_i} \qquad (3)$$

Equation (2) is the mixing rule used to obtain adsorption values for adsorbing components in a mixture where only pure component adsorption data are available. Equation (2) meets some important criteria: (i) it is a derived based on a fairly acceptable physical model, i.e. the Langmuir model; (ii) it reproduces the actual entered laboratory data in the case where there is only one absorbing component. Equation (2) is expected to perform especially well for cases where a single component dominates the adsorption.

In terms of data entry requirements, the user enters tables of adsorption values, ω_i, versus aqueous component mole fraction, y_{i_aqu}, for each component as well as a value for the maximum adsorption for each component, $\omega_{i,max}$. The adsorption, ω_i is used to calculate Γ_i at each aqueous mole fraction from Equation (3). The converted table is stored internally as Γ_i versus mole fraction for each adsorbing component.

To estimate adsorption in a multicomponent mixture of composition y_{i_aqu}, a table look up is performed for each component to determine Γ_i. Equation (2) is subsequently applied to calculate ω_i from Γ_i and $\omega_{i,max}$. Linear interpolation is used for mole fractions between table entries. For mole fractions outside the table range, the first or last entry in the table is selected respectively.

In case the model has aqueous components involving chemical equilibrium reactions, the adsorption/desorption is allowed only for the primary components, i.e. for first (number of aqueous components minus number of chemical equilibrium reactions) components.

Aqueous Phase Adsorption Controlling Data (Optional)

***ADRT, *PORFT,*RRFT**

PURPOSE:

These array keyword control residual adsorption level, accessible pore volume, and residual resistance factor for an adsorbing component from the aqueous phase.

ARRAY:

*ADRT	'comp_name'	Array qualifier	adrt
*PORFT	'comp_name'	Array qualifier	porft
*RRFT	'comp_name'	Array qualifier	rrft

DEFINITIONS:

*ADRT

Specifies 'adrt' the residual adsorption level in moles of adsorbed component per unit mass of rock (gmole of component / (kg | lb) of rock). The allowed range is from 0 to $\omega_{i,max}$. Note that $\omega_{i,max}$ is specified either via Langmuir (*ADAMAXC) or tabular (*ADSORBTMAXA) form of adsorption input.

A zero value implies that adsorption is completely reversible, while adrt = $\omega_{i,max}$ denotes completely irreversible adsorption.

*PORFT

Specifies 'porft' the accessible pore volume or fraction of available pore volume. The allowed range is from 0 to 1.

It can be viewed also as one minus the fraction of pore volume that is inaccessible to the adsorbing component.

*RRFT

Specifies 'rrft' the residual resistance factor for the adsorbing component. It must be greater than or equal to 1

A rrft value of 1 implies no resistance factor while a value larger than 1 implies resistance factor due to adsorption. For *RRFT to be applicable it is necessary to specify *RFCALC *ADS-RRFT in the Component Properties section.

'comp_name'

Adsorbing component name within quotes for which *ADRT, *PORFT, or *RRFT values are being specified. A maximum of 8 characters, excluding the quotes, is expected.

DEFAULTS:

Optional keywords. Following defaults apply for all the adsorbing components in aqueous phase:

ADRT defaults to zero, i.e., by default, the adsorption in aqueous phase is completely reversible.

PORFT defaults to 1, i.e., by default, no inaccessible pore volume.

RRFT defaults to 1, i.e., by default there is no resistance effect.

CONDITIONS:

These keywords, if present, must be in the ROCK-FLUID DATA group.

EXPLANATION:

Keyword *ADRT is applicable only to the component 'comp_name' if it is an adsorbing component in the aqueous phase. If in an on-going process desorption takes place, *ADRT will control the extent to which desorption occurs.

The effect of *PORFT for the adsorbing component is to effectively reduce porosity for adsorbing component in the accumulation term.

Adsorption or mechanical entrapment can cause blockage which amounts to a reduction in the effective permeability, which is accounted for by a resistance factor calculated for all of the phases based on the value of *RRFT. For *RRFT to be applicable it is necessary to specify *RFCALC *ADS-RRFT in the Component Properties section.

For a total of m adsorbing components in aqueous phase, the resistance factor is calculated as under, where k and n represent current; and previous timestep values of porosity and adsorption quantities.

$$\mathrm{rf} = \prod_{i}^{m} \left(\frac{A}{B} \right)$$

where,

$$A = 1 + \frac{(\mathrm{rrft(i)} - 1)\omega_i^k / \omega_{i_max}}{\phi^k}$$

and,

$$B = 1 + \frac{(\mathrm{rrft(i)} - 1)\omega_i^n / \omega_{i_max}}{\phi^n}$$

ω_i = moles of adsorbed component per unit mass or rock

$\omega_{i,max}$ = maximum moles of adsorbed component i per unit mass of rock

Note that the porosity or permeability values themselves do not get altered but the effect is taken into account in accumulation and flow terms.

Rock Mass Density (Optional) *ROCKDEN

PURPOSE:

Specify value of rock-density for each gridblock used in calculation of adsorption of component from the gas, aqueous, or solid asphaltene phase onto the reservoir rock.

ARRAY:

*ROCKDEN	Array qualifier	rockden

DEFINITIONS:

rockden

Rock mass density in units of [kg/m3 | lb/ft3]. This is the actual rock density, not including its pore space.

DEFAULTS:

Optional keyword. *ROCKDEN defaults to 2650 kg/m^3 (165.4341 lb/ft^3).

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA group.

EXPLANATION:

See description related to multi-component adsorption form gas or aqueous phase about use of rock density in adsorption calculations in Rock-Fluid Data section, and solid asphaltene phase adsorption in Component Properties section.

Velocity-Dependent Dispersion (Optional) *DISPERARRAY-LNG, *DISPERARRAY-TRN, *DISPERARRAY-ISO

PURPOSE:

These keywords specify the dispersivities for the calculation of longitudinal and transverse velocity-dependent dispersion effects for all fluid phases.

ARRAY:

*DISPERARRAY-LNG
*DISPERARRAY-TRN
*DISPERARRAY-ISO

DEFINITIONS:

*DISPERARRAY-LNG
Dispersivities in the longitudinal flow direction (parallel to the flow) in (m | ft).

*DISPERARRAY-TRN
Dispersivities in the transverse flow direction (perpendicular to the flow) in (m | ft).

*DISPERARRAY-ISO
Dispersivities assuming that longitudinal and transverse dispersivities are identical in (m | ft).

DEFAULTS:

Optional keyword. The default is to have no dispersion.

CONDITIONS:

These keywords must be in the ROCK-FLUID DATA group. Either DISPERARRAY-LNG and DISPERARRAY-TRN are both entered, or DISPERARRAY-ISO is entered alone, or none of the keywords are used.

EXPLANATION:

*DISPERARRAY-LNG, *DISPERARRAY-TRN and *DISPERARRAY-ISO indicate the input of an array of grid values. All array reading option subkeywords are valid. See the section on ARRAY READING OPTIONS for more details.

Dispersion is the mixing of fluids caused by diffusion, local velocity gradients, locally heterogeneous streamline lengths, and mechanical mixing (Lake, 1989). The dispersive flux J_i of Component i represented by a concentration y is given by:

$$J_i = -\sum_k \phi \rho_k S_k D_{ik} \nabla y_{ik}, \quad i = 1,..., n_c; \; k = o, g, w$$

If (x, y, z) represents the coordinate system, the dispersion tensor, $\mathbf{D}_{ik}$, may be expressed as:

$$D_{ik} = \begin{bmatrix} D_{ik,11} & D_{ik,12} & D_{ik,13} \\ D_{ik,21} & D_{ik,22} & D_{ik,23} \\ D_{ik,31} & D_{ik,32} & D_{ik,33} \end{bmatrix}$$

where the tensor components are given by expressions of the form:

$$D_{ik,mm} = \frac{D^*_{ik}}{F_k \, \phi \, S_k} + \frac{(\alpha_{kl} - \alpha_{kt})}{\phi \, S_k \, |u_k|} u^2_{km} + \alpha_{kt} \frac{|u_k|}{\phi \, S_k}$$

$$D_{ik,mn} = \frac{(\alpha_{kl} - \alpha_{kt})}{\phi \, S_k \, |u_k|} u_{km} \, u_{kn}$$

where:

D^*_{ik}	=	molecular diffusion coefficient of Component i in Phase k
F_k	=	formation resistivity factor for Phase k
α_{kl}	=	longitudinal dispersivity for Phase k
α_{kt}	=	transverse dispersivity for Phase k
μ_k	=	phase velocity for Phase k

Note that molecular diffusion is discussed elsewhere (see *DIFFC_OIL, *DIFFC_GAS and *DIFFC-AQU in the Component Properties section).

The dispersivities are currently assumed to be phase independent.

For an isotropic system, $\alpha_{kl} = \alpha_{kt} = \alpha_k$. When this occurs, only one array needs to be input (*DISPERARRAY-ISO), and the other two arrays are set equal to it, internal to the simulator.

Dispersion often needs a fully implicit formulation (use *AIMSET *CON 3; see *AIMSET).

Coal Bed Methane Features (Optional) *COAL-DIF-COMP, *COAL-DIF-TIME

PURPOSE:

Either of these keywords activates the multicomponent coal bed features in GEM, and also specifies the matrix (coal) to fracture (cleat) diffusion process in a coal bed. These keywords should be used for Coal Bed Methane (CBM) modelling and Enhanced Coal Bed Methane (ECBM) modelling.

ARRAY:

*COAL-DIF-COMP
*COAL-DIF-TIME

DEFINITIONS:

COAL-DIF-COMP

Sets positive real numbers giving gas phase diffusion values for a specified component (in cm^2/sec; all unit systems) that describes matrix (coal) to fracture (cleat) mass transfer. True molecular diffusivities may be entered, but it is more common to regard these coefficients as adjustable parameters which are used to match field observations. Note that the overall mass transfer rate will be a product of these diffusion coefficients with an internally-determined shape factor that depends on fracture (cleat) spacing and that has units of 1/area. See below for further details.

COAL-DIF-TIME

Sets positive real numbers that imply certain gas phase diffusion values As noted above, the overall mass transfer rate from matrix (coal) to fracture (cleats) will be a product of a diffusion coefficient with an internally-determined shape factor that depends on fracture (cleat) spacing and that has units of 1/area. This product has units of inverse time, and can be set directly by specifying a time (not inverse time) using this keyword. Units are always days. See below for further details.

DEFAULTS:

Optional keyword. Default is no inclusion of special Coal Bed modelling functionality. No default values.

CONDITIONS:

These keywords must appear in the ROCK-FLUID DATA group. Either COAL-DIF-COMP or COAL-DIF-TIME should be entered, but not both. The DUALPOR keyword must have been used in the RESERVOIR DESCRIPTION SECTION and a proper description of a dual porosity reservoir must have been given.

EXPLANATION:

This keyword activates Coal Bed modelling functionality for modelling coal beds in GEM. These features work within a dual porosity framework and include the modelling of gas phase diffusion between matrix cells (coal) and fracture cells (cleat system).

The preferred procedure for preparing the model is to first define the grid structure and sizing as usual, and then to specify the DUALPOR keyword (using the default SHAPE GK) and a set of cleat spacings using the DIFRAC/DJFRAC/DKFRAC keywords (recalling that zero values can be used to indicate the lack of fracture planes perpendicular to some direction). Fracture porosity is required to indicate the reservoir volume that the cleats occupy, while matrix porosities should be set to small positive values to allow for the modelling of a small amount of free gas within the matrix porosity. Similarly, fracture permeabilities are required to describe the permeability of the cleat system. Values must also be entered for matrix permeabilities as well, but since the simulator disables ALL matrix-to-fracture Darcy flow when coal bed modelling is being used, a positive value is only used to indicate that there is a pathway for diffusion to occur between matrix (coal) and fracture (cleats), while a zero value indicates that no connection of any kind is to be made between the matrix and fracture for that cell.

Fluid properties are usually defined using WINPROP and usually only require the definition of a single or multicomponent gas using library properties. (Do not use the DIFFUSION keyword as it is not used for Coal Bed modelling.) Relative permeability tables for gas and water are required for the Rock-Fluid section, and gas adsorption data is required for the matrix rock (coal), through use of a Langmuir model or by table. Adsorption coefficients for the fracture need to be set to zero. The simulator will calculate a matrix pressure that keeps the desorbing matrix gases in equilibrium with a free gas phase in the matrix porosity, and this matrix gas will move between the fractures (cleats) and matrix by a diffusion process described by the COAL-DIF-COMP and COAL-DIF-TIME keywords.

When initializing the model, pressures and compositions can be set as usual. It is possible to define an initialization region that consists only of matrix cells and another consisting of fracture cells, and to set the initial pressures lower for the matrix cells than for the fracture cells (cleats). The model will then initialize an adsorbed gas in place in the matrix (coal) using the matrix pressure, which will result in a lower gas content than would be expected by looking at the fracture (cleat) system pressures, and hence models an undersaturated coal bed system. Note that the model will <u>always</u> initialize the matrix (coal) with no water ($S_{w,matrix} = 0$) when Coal Bed modelling is being used. (Since Coal Bed modelling inhibits matrix-to-fracture Darcy flow for both gas and water, there seems little point in modelling water saturations within the matrix (coal) as such water would remain completely bound throughout the entire simulation).

Wells should be completed and operated as usual, noting that the wells will automatically complete themselves in the fractures (cleats) of specified perforation cells.

As regards gas diffusion between the matrix (coal) and the fracture (cleat) system, the gas phase mass transfer rate for component "k" from the small elemental portion of a matrix cell "m" that is bounded by a set of fracture cleats associated with a fracture cell "f" can be written as follows:

$$Rate_{elem\text{-}basis} = \Sigma\ 2*(A/\Delta) * Diffus(k) * Sg^{A\text{-}mod} * (C(k,gas,m) - C(k,gas,f))$$

Here "A" is the contact area between the elemental matrix portion, given by the product of fracture spacings, and "Δ" is the distance between the centre of the element and the fracture, computed as half the fracture spacing, in a given direction, summed over the three spatial directions. C(k,gas,m) is the concentration of component "k" in the gas phase of matrix cell "m" and C(k,gas,f) is the same for fracture cell "f". Diffus(k) is the diffusion value given by COAL-DIF-COMP for component "k" (input in units of cm^2/sec, but converted internally by the simulator for use in the formulae here) Diffus(k) must include any tortuosity effects, and the fact that no matrix porosity term is included in the above means that the tortuosity is being assumed to be proportional to matrix porosity, thereby canceling out the effects of that term. (Compare the description of the DIFFUSION keyword in the Component Properties section). Finally, the presence of water in the fracture can affect diffusive flows. When net diffusion is from the matrix to the fracture, $Sg^{A\text{-}mod}$ is taken to be 1 (which is the assumed gas saturation in the matrix) , and fracture water has no effect. But, when net diffusion is from the fracture to the matrix, water in the fracture is expected to reduce the contact area for diffusive flow. Thus, $Sg^{A\text{-}mod}$ is taken to be the gas saturation in the fracture (that is, 1-Sw) when the gas (molar) density in the fracture exceeds that of the co-located matrix.

The above expression can be scaled up to the volume of the entire block by multiplying the above by the ratio of the bulk block volume "Vol" to the volume of the small elemental portion:

$$\text{Rate}_{\text{block-basis}} = \text{Vol} * \text{Shape} * \text{Diffus(k)} * Sg^{A\text{-}mod} * (\text{C(k,gas,m)} - \text{C(k,gas,f)})$$

where $\text{Shape} = 4 * \sum (1/\text{FracSpacing})^2$.

If COAL-DIF-COMP is not used to set "Diffus(k)", then COAL-DIF-TIME can be used to define 1/(Shape*Diffus(k) in units of days.

Example: The following fragment uses time-based input. Note that it suffices to set these values for one porosity system only (as they are actually a property shared between matrix and fracture porosities for a single physical block), so stating them once without use of the *MATRIX or *FRACTURE qualifiers suffices. A component name is required however. The data fragment follows:

```
...
*COAL-DIF-TIME 'CO2' *CON 100.0
*COAL-DIF-TIME  'N2' *CON 100.0
*COAL-DIF-TIME 'CH4' *CON 100.0
...
```

Flag to Use Alternate Relative Permeability Table Definition for Coal Bed Methane (Optional) *KRCBM

PURPOSE:

*KRCBM indicates that the input relative permeability tables (*SWT, *SGT, *SLT) do not contain columns for relative permeability to oil.

FORMAT:

*KRCBM

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword must be in the ROCK-FLUID DATA keyword group. This keyword should only be used with coal bed methane (CBM) modeling. If *KRCBM is entered in a non-CBM model, an error message is issued.

EXPLANATION:

The values of relative permeability to oil in water (kro column in *SWT table) and also in gas and connate water (krg column in *SGT and *SLT tables) are ignored since the CBM reservoirs contain only gas and water phases. These columns are required data entry and therefore mock values have to be input. With the *KRCBM keyword, these two columns are no longer needed in the relative permeability tables.

Subdomain Reinfiltration (Optional) *SD_REINF

PURPOSE:

*SD_REINF works with TRANSD to modify the Inter-Subdomain connection (defined by TRANSD > 0.0) to represent reinfiltration.

ARRAY:

*SD_REINF

DEFAULTS:

Optional keyword. Defaults: 0.0

CONDITIONS:

This keyword may be in the Rock-Fluid Data keyword group only and is only valid when the *SUBDOMAIN dual porosity option is in effect and TRANSD is defined for at least one Inter-Subdomain connection. Use of this keyword is enhanced if the *FRACVOL Reservoir Description Keyword is used to reduce the relative size of the top and bottom blocks in the subdomain.

EXPLANATION:

The use of the TRANSD array allows the matrix stacks in each subdomain to be connected vertically together. This allows gravity drainage to occur directly from subdomain to subdomain.

There is normally capillary continuity between subdomains (see *TRANSD).

The *SD_REINF array introduces capillary discontinuity between the bottom block of a matrix subdomain and other subdomain blocks it is connected to (including the connection created using *TRANSD

The lowest subdomain block stands in for the fracture when modeling flow from the matrix into a vertical fracture then reinfiltration from the vertical fracture back into the matrix.

In order to be valid, it is important that *FRACVOL be used to reduce the thickness of the last subdomain block, as it is this subdomain block which is used to represent the vertical fracture between matrix blocks. See keywords *FRACVOL and *TRANSD.

Normally values of *SD_REINF = 1.0 or 0.0 are used. This turns the reinfiltration model on (1.0) or off (0.0). However in certain instance it may be desirable to use this model to approximately account for limited reinfiltration which occurs alongside gravity drainage. In this case, fractional values of *SD_REINF may be entered.

*SD_REINF introduces a capillary discontinuity by modifying the bottom-most subdomain block's Pcog and Pcow using the following formula (for each subdomains lowest block).

A scale factor (PCSCALE) is calculated for both oil-water and oil-gas capillary pressure in the lowest subdomain block.

PCSCALE = 1.0 – *SD_REINF

The capillary pressures of that single subdomain block are adjusted by PCSCALE.

PCOW = PCOW*PCSCALE

PCOG = PCOG*PCSCALE

Examples:

The following provides an example of TRANSD and *SD_REINF usage:

```
**In Reservoir Description
** Turn on direct gravity drainage between subdomains
** for all blocks

*TRANSD *CON 1.0
    :
 ** In Rock-Fluid Data
** Introduce capillary discontinuity in inter
** Subdomain flow.

*SD_REINF CON 1.0
```

The suggested range of values for SD_REINF is:

	SI	Field
min	0.0	0.0
max	1.0	1.0

Single-Phase Relative Permeability Interpolation (Optional)

***TRLIMS**

PURPOSE:

Use of *TRLIMS indicates that reduced temperature limits will be used together with the pseudo-critical temperature of a single hydrocarbon phase to generate a relative permeability interpolating function. This function is used to calculate a water-hydrocarbon relative permeability by interpolating between the water-oil and gas-liquid relative permeability curves.

FORMAT:

*TRLIMS (troil trgas)

DEFINITIONS:

TRLIMS
: Indicates that single phase relative permeability interpolation will be used.

troil trgas
: Optional input of the reduced temperature limits used to define the region for interpolation.

DEFAULTS:

Optional keyword. Default is to not use relative permeability interpolation. If TRLIMS is entered by itself, with no values specified for troil and trgas, the default limits of troil = 0.75 and trgas = 1.25 will be used.

CONDITIONS:

If it appears, this keyword must be in the ROCK-FLUID DATA keyword group. It can appear anywhere within this group.

EXPLANATION:

When there is only a single hydrocarbon phase in a grid block, by default the relative permeability of the phase will be determined from the krow curve in the SWT table if the phase is identified as an oil, and from the krg curve in the SLT or SGT table if the phase is identified as a gas.

In gas injection processes, or in reservoirs with a critical transition from oil to gas (undersaturated GOC), there may be a continuous variation of phase properties from an oil-like to a gas-like region. To allow a continuous variation of relative permeability consistent with the variation of other phase properties, interpolation between the krow and krg curves can be done. The interpolated relative permeability tends to krow for oil-like phases, and to krg for gas-like phases.

The pseudo-critical temperature for a single hydrocarbon phase can be calculated using the Li mixing rule:

$$T_{c,mix} = \frac{\sum_i z_i v_{c,i} T_{c,i}}{\sum_i z_i v_{c,i}}$$

Use of this mixing rule for single phase identification is described under *PHASEID *TCMIX, and tuning of the mixing rule to achieve the desired phase identification is described under *TCMULT.

The reduced temperature at a grid block is defined as tred = $T_{block}/T_{c,mix}$. The interpolating function is calculated as f = (tred – troil)/(trgas – troil), for troil < tred < trgas.

If tred < troil: f=0 and the relative permeability krow is used directly.

If tred > trgas: f = 1 and the relative permeability krg is used directly.

The hydrocarbon-water relative permeability is then calculated from:

krhw = (1-f)*krow + f*krg

where krow and krg have their endpoints scaled so that the endpoints of krhw vary continuously as f changes.

Examples:

To use single-phase relative permeability interpolation with the default reduced temperature limits:

```
*TRLIMS
```

To use single-phase relative permeability interpolation with the user-input reduced temperature limits:

```
*TRLIMS  0.7  1.3
```

Non-Darcy Flow in Reservoir (Optional)

***NONDARCY, *KLINKENBERG**

PURPOSE:

*NONDARCY signals the use of the non-Darcy reservoir flow option. *KLINKENBERG indicates input of a reference pressure for a low-pressure correction to Darcy's law.

FORMAT:

```
*NONDARCY          (*GEERTSMA)
                   (*FG1)
                   (*FG2)
-or-
*NONDARCY   *GENERAL    wt_ndarcy
Coefficient Table
*KLINKENBERG            pkkbrg
```

DEFINITIONS:

*GEERTSMA

*GEERTSMA signals the use of Geertsma's (1974) correlation for gas phase non-Darcy flow.

*FGI

*FG1 signals the use of Frederick and Graves (1994) first correlation for gas phase non-Darcy flow.

*FG2

*FG2 signals the use of Frederick and Graves (1994) second correlation for gas phase non-Darcy flow.

*GENERAL

*GENERAL signals that a table of coefficients follows which will define how non-Darcy flow is a function of saturation, porosity and permeability for each phase.

wt_ndarcy

A weighting factor which controls how much the non-Darcy Forchheimer number is allowed to change over a timestep. The default is 1.00. Excellent results are obtained when a value of 0.50 is used. Values above 0.50 are not recommended.

Coefficient Table

A table of coefficients for β (beta) the non-Darcy flow coefficient for each phase (defined below). β (beta) always has units of 1/ft. regardless of the unit system used in the simulator. It is always assumed that we are using Permeability in mD and a α_g so that we obtain a β factor with units of 1/ft.

pkkbrg

Reference pressure for the Klinkenberg effect [kpa | psi].

DEFAULTS:

Optional keywords. The default is not to include these effects.

CONDITIONS:

These keywords must be in the ROCK-FLUID keyword group.

EXPLANATION:

For any phase the Forchheimer equation can be written as

$$-\nabla \phi_p = \frac{\mu_p}{K K_{rp}} + \beta_p \rho_p \left| \vec{U}_p \right| \vec{U}_p$$

where ϕ_p is the potential of phase P and β_p is the non-Darcy coefficient for that phase.
The above equation can be rewritten as:

$$-\nabla \phi_p = \frac{\mu_p}{K K_{rp}} \left(1 + Fo_p\right) \vec{U}_p$$

with

$$Fo_p = \beta_p K K_{rp} \rho_p \left| \vec{U}_p \right| / \mu_p$$

Fo is referred to as the Forchheimer Number.

The correlations for β_p are all of the form (K in md, and βp in 1/ft.)

$$\beta_p = \frac{\alpha_p}{\left(K K_{rp}\right)^{N1_p} \left(\phi S_p\right)^{N2_p}}$$

When using the *NONDARCY *GEERTSMA, *FG1 or *FG2 option β for all phases except gas is assumed to be zero. For gas the following parameters for α_g, $N1_g$ and $N2_g$ are used.

Correlation	α_g	$N1_g$	$N2_g$
*Geertsma	48511.34	0.5	5.5
*FG1	7.89E10	1.6	0.404
*FG2	2.11E10	1.55	1.0

When using the *NONDARCY *GENERAL option the user can enter a table of α, N1, N2 and $Forch_{max}$ (defined below) for each phase in the reservoir.

It is always assumed that we are using Permeability in mD and a α_g so that we obtain a β factor with units of 1/ft.

The first three columns of the table represent α, N1 and N2 respectively, while the rows indicate which phase the coefficients correspond to. The last column (4th) is $Forch_{max}$, which is normally defaulted to 1.00e+04, can be used to input the maximum Forchheimer no. for

each phase. A Forchheimer no. calculated to be greater than $Forch_{max}$ is limited to $Forch_{max}$. It is recommended that $Forch_{max}$ is allowed to default to 1.00e+04.

The Coefficient Tables are of the following form, the form differs slightly for different *MODEL (model) types

For models with gas, water and oil phases present (no solvent) the form of the coefficient tables is:

***NONDARCY *GENERAL**		**wt_ndarcy**	
α_g	$N1_g$	$N2_g$	$Forch_{max}g$
α_w	$N1_w$	$N2_w$	$Forch_{max}w$
α_o	$N1_o$	$N2_o$	$Forch_{max}o$

Note: A value of $N2_p = 0.0$ will eliminate the dependence of β_p on (ϕ x S_p) and allow a correlation for β_p of the form:

$$\beta_p = \frac{\alpha_p}{\left(K K_{rp}\right)^{N1_p}}$$

This form is common when modelling non-Darcy flow in fractures.

At low pressures, Darcy's law requires a correction for the mean flowing pressure. Thus, the effective absolute permeability for gas is assumed to be related to permeability measured for liquids as follows:

$$k_{g,\,eff} = k\ (1 + (pkkbrg / p)$$

where pkkbrg is a reference pressure for a given gas and a given porous medium, p is the reservoir pressure and the permeability k is assumed to have been measured for liquids (permeability at infinite pressure). This dependence of permeability on pressure, known as the Klinkenberg effect, is attributed to gas slippage along the pore wall of the porous medium.

Examples:

1. To use the coefficients associated with the Frederick and Graves model II for the gas phase only , and the default maximum Forcheimer number, but with a weight factor of 0.5:

```
*NONDARCY  *GENERAL  0.5
2.11E+10  1.55  1.0
0.0       0.0   0.0
0.0       0.0   0.0
```

2. To specify a model with no dependence on porosity, to be applied to all phases, and with an increased maximum Forcheimer number:

```
*NONDARCY  *GENERAL  0.5
27.3E+09  1.1045  0.0  5.0E+04
27.3E+09  1.1045  0.0  5.0E+04
27.3E+09  1.1045  0.0  5.0E+04
```

3. To use the coefficients associated with the Geertsma model for the gas phase:

```
*NONDARCY  *GEERTSMA
```

Non-Darcy Coefficient Correction Factor (Conditional)

***NDARCYCOR**

PURPOSE:

Allows the input of block by block correction factors for the non-Darcy β factor calculated as described in the background section for the *NONDARCY keyword.

ARRAY:

*NDARCYCOR

DEFAULTS:

Conditional keyword. By default all values are 1.0.

CONDITIONS:

Used only when non-Darcy flow is needed. This keyword may appear in the Rock-Fluid or Recurrent Data sections.

DEFINITION:

This array allows the user to modify the Forchheimer Number for any block. The correction is only applied to the gas phase unless the *NONDARCY *GENERAL option is used. If *NONDARCY *GENERAL is used, the correction is applied to all phases.

Setting *NDARCYCOR to zero in a region of the reservoir will force that region to exhibit Darcy flow and significantly reduce the non-Darcy model's calculations. The factor is dimensionless.

*NDARCYCOR can also be used to effectively model non-Darcy flow in thin grid blocks which are set up to represent fractures. The proper definition of *NDARCYOR, which is a function of the fracture pseudo permeability and of $N1_p$ (in the correlation for non-Darcy β factor), will allow the user to model non-Darcy flow in 0.10 inch thick fractures using grid blocks as wide as 1 foot. See the tutorial section on modelling non-Darcy flow in fractures for a discussion of how this is done.

STARS Scaling Option (Optional) *SCALING-STARS

PURPOSE:

*SCALING-STARS implies the selection of end point scaling approach first implemented in STARS.

FORMAT:

*SCALING-STARS

DEFAULTS:

Optional keyword. No default.

CONDITIONS:

This keyword must be the in the ROCK- FLUID DATA keyword group.

EXPLANATION:

One of the end point saturations for the oil phase relative permeability curve (in the context of two phase oil-gas system; SGT or SLT table) is the irreducible oil saturation, *soirg*. By default the value assigned to a grid block is based on the value entered with the SGT or SLT table. For the SGT table the last row in the table specifies the maximum gas saturation which in turn is equal to 1- *soirg – swcon*. *Swcon* is the connate water saturation. For the SLT table the saturation specified in the first row is the connate liquid saturation which is equal to *soirg + swcon*. The user can change this end point saturation for a particular grid block(s) by entering value(s) with either the keyword SOIRG or the keyword SLCON. If SCALING-STARS option is chosen then use the corresponding keyword SOIRG. The connate liquid saturation for a grid block will then be calculated as the sum of the entered values of *soirg* and *swcon*. If SLCON keyword is used then *soirg* is calculated as the difference between the connate liquid and connate water, *soirg = slcon –swcon*.

For the case where equal values of connate and critical saturations are specified via the SWT/SGT/SLT table and SCALING-STARS is chosen, then changing a grid block value with one of the end point modification keywords such as SWCON/SWCRIT will also modify the corresponding critical or connate value. For example if SCALING-STARS appears in the rock fluid section of the data and connate water and critical water values specified via SWT table are the same and equal, to say 0.2. Now changing the connate water value for a specific grid block via the keyword SWCON to 0.3 will also change the critical water saturation for that grid block to 0.3.

Langmuir Maximum Adsorption Parameter Multiplier (Optional) *LANGMULT

PURPOSE:

*LANGMULT indicates input of multiplier fractions used to adjust the entered maximum Langmuir adsorption parameter values (see keyword *ADGMAXC and *ADSORBTMAX; *ADAMAX and *ADSORBTMAXA) on a grid block by grid block basis.

ARRAY:

*LANGMULT

DEFAULTS:

Optional keyword. The default is a multiplier value of 1.0

CONDITIONS:

This keyword must be in the ROCK and FLUID properties section of the data file. Any of the array reading options can be used for this keyword.

EXPLANATION:

The *LANGMULT array assigns selected blocks the specified multiplier. The multiplier entered is applied to all adsorbing components in gas phase. For the multiplier to have any effect adsorption must be turned on. The multiplier essentially scales the maximum Langmuir adsorption constant for each component and grid block by the value of the multiplier. Therefore for each component

adgmax (m,ips) = adgmax(m,ips) * xlangmult(ips)

Here “m” stands for the adsorbing component and “ips” for the grid block index.

Note: The keyword *ADGMAX uses grid block array reading options. Therefore instead of the keyword *LANGMULT it is possible to modify the array values using the functionality associated with the *MOD keyword. However if adsorption tabular input is used to enter adsorption versus pressure and composition, then the maximum adsorption values are read in for each rock type and not on a grid block by grid block basis. To modify values on a grid block by grid block basis, therefore, the keyword LANGMULT is required.

Example syntax:

Keyword block 1 below

```
*ADGMAXC 'C1'  *MATRIX   *CON 0.6666667
*ADGMAXC 'CO2' *MATRIX   *CON 1.3333333
 *LANGMULT *MATRIX  *IJK ** applies to all components
  1:10  1:10  1:1 0.75  ** multiplier applied to ADGMAXC,
                        ** ‘C1’ and ‘CO2’
```

and keyword block 2 below

```
*ADGMAXC 'C1'  *MATRIX   *CON 0.5
*ADGMAXC 'CO2' *MATRIX   *CON 1.0
```

are equivalent ways of specifying the same values of the maximum adsorption Langmuir parameter for matrix blocks 1:10 1:10 1 since 0.667*0.75 = 0.5 and 1.333*0.75=1.0

Three-point Endpoint Scaling (Optional) *3PTSCALING, *3PTSCAL_KRW, *3PTSCAL_KROW, *3PTSCAL_KRG, *3PTSCAL_KROG

PURPOSE:

Enable multi-point scaling for saturation endpoints of rock-fluid curves.

FORMAT:

*3PTSCALING	(*ON | *OFF)
*3PTSCAL_KRW	(*ON | *OFF)
*3PTSCAL_KROW	(*ON | *OFF)
*3PTSCAL_KRG	(*ON | *OFF)
*3PTSCAL_KROG	(*ON | *OFF)

DEFINITIONS:

*3PTSCALING (*ON | *OFF)
: Enable (*ON) or disable (*OFF) multi-point scaling for saturation endpoints of all rock-fluid curves. This is equivalent to specifying the same option for all the relative permeability endpoints.

*3PTSCAL_KRW (*ON | *OFF)
: Enable (*ON) or disable (*OFF) three-point scaling for saturation endpoints of the water relative permeability curve specified by *SWT (k_{rw} for *WATWET, k_{ro} for *OILWET).

*3PTSCAL_KROW (*ON | *OFF)
: Enable (*ON) or disable (*OFF) three-point scaling for saturation endpoints of the oil relative permeability to water curve specified by *SWT (k_{row} for *WATWET, k_{ro} for *OILWET).

*3PTSCAL_KRG (*ON | *OFF)
: Enable (*ON) or disable (*OFF) three-point scaling for saturation endpoints of the gas relative permeability curve specified by *SLT (k_{rg}).

*3PTSCAL_KROG (*ON | *OFF)
: Enable (*ON) or disable (*OFF) three-point scaling for saturation endpoints of the oil relative permeability to gas curve specified by *SLT (k_{rog} for *WATWET, k_{rwg} for *OILWET).

DEFAULTS:

If any of the three-point endpoint scaling keywords appear without an *ON or *OFF then *ON is assumed.

CONDITIONS:

Each keyword applies to all rock types, that is, all rock types will use the same endpoint scaling option.

If *3PTSCALING *ON is in effect but the multi-point saturation scaling keyword for any particular rock-fluid curve (e.g. *3PTSCAL_KRW) is entered, the selection based on the individual keyword such as *3PTSCAL_KRW overrides the choice made with *3PTSCALING.

If *3PTSCALING appears together with any of the saturation three-point scaling keywords listed above, the keyword located later in the data set over-rides the others. For example, if *3PTSCAL_KRW *OFF appears after *3PTSCALING *ON, the k_{rw} curve uses saturation two-point scaling. Similarly if *3PTSCALING *ON appears after *3PTSCAL_KRW *OFF, then three-point scaling will be used in saturation (horizontal) scaling of the water k_{rw} curve.

EXPLANATION:

When the three point saturation scaling option is used, the scaling option of relative permeability endpoints used to calculate k_{rw}, k_{row}, k_{rog} and k_{rg}, as described under the keyword explanation for "Saturation Endpoints for each Grid Block" is altered.

In the default "two point" scaling, relative permeability curves are scaled between two saturation endpoints, i.e.:

k_{rw}	Scaled between S_{wcrit} and $(1-S_{oirw})$
k_{row}	Scaled between S_{wcon} and $(1-S_{orw})$
k_{rg}	Scaled between S_{lcon} and $(1-S_{gcrit})$
k_{rog}	Scaled between S_{lrg} and $(1-S_{gcon})$

When any of *3PTSCALING, *3PTSCAL_KRW, *3PTSCAL_KROW, *3PTSCAL_KRG or 3PTSCAL_KROG is enabled an interior scaling point is added to the scaling procedure.

For each of k_{rw}, k_{row}, k_{rg}, and k_{rog}, a different interior scaling point is used depending on the value of block water saturation (oil-water table) and block liquid saturation (gas-liquid table).

$$k_{rw} \quad \text{Scaled between} \begin{cases} S_{wcrit} \text{ and } (1-S_{orw}) & \text{if block } S_w \le (1-S_{orw}) \\ (1-S_{orw}) \text{ and } (1-S_{oirw}) & \text{if block } S_w > (1-S_{orw}) \end{cases}$$

$$k_{row} \quad \text{Scaled between} \begin{cases} S_{wcon} \text{ and } S_{wcrit} & \text{if block } S_w \le S_{wcrit} \\ S_{wcrit} \text{ and } (1-S_{orw}) & \text{if block } S_w > S_{wcrit} \end{cases}$$

$$k_{rg} \quad \text{Scaled between} \begin{cases} S_{lcon} \text{ and } S_{lrg} & \text{if block } S_l \le S_{lrg} \\ S_{lrg} \text{ and } (1-S_{gcrit}) & \text{if block } S_l > S_{lrg} \end{cases}$$

$$k_{rog} \quad \text{Scaled between} \begin{cases} S_{lrg} \text{ and } (1-S_{gcrit}) & \text{if block } S_l \le (1-S_{gcrit}) \\ (1-S_{gcrit} \text{ and } (1-S_{gcon}) & \text{if block } S_w > (1-S_{gcrit}) \end{cases}$$

It is important to ensure that when using three point scaling, block endpoint scaling arrays for (S_{wcrit} and S_{wcon}), (S_{orw} and S_{oirw}), (S_{lrg} and S_{lcon}), and (S_{gcrit} and S_{gcon}) are significantly different (> 1%). If not, the three point saturation scaling calculation will revert to two-point scaling, to avoid using a near infinite scaling factor. This is only done in the affected blocks and it does not disable *3PTSCALING, *3PTSCAL_KRW, *3PTSCAL_KROW, 3PTSCAL_KRG or *3PTSCAL_KROG globally.

More generally the scaling from block saturations (BS) to table saturations (TS) has the form (using nomenclature from "Saturation Endpoint for each Grid Block" scaling explanation):

$Sat_{ts} = Slow_{ts} + (Sat_{bs} - Slow_{bs}) \times (Supp_{ts} - Slow_{ts}) / (Supp_{bs} - Slow_{bs})$

For k_{rw} at (Sat_{ts})

$Sat = S_w$

Two point scaling:

$Slow_{ts} = S'_{wcrit}$, $Slow_{bs} = S_{wcrit}$, $Supp_{ts} = 1 - S'_{oirw}$, $Supp_{bs} = 1 - S_{oirw}$

Three point scaling:

When $Sat_{bs} < 1 - S_{orw}$: $Slow_{ts} = S'_{wcrit}$, $Slow_{bs} = S_{wcrit}$, $Supp_{ts} = 1 - S'_{orw}$, $Supp_{bs} = 1 - S_{orw}$

When $Sat_{bs} > 1 - S_{orw}$: $Slow_{ts} = 1 - S'_{orw}$, $Slow_{bs} = 1 - S_{orw}$, $Supp_{ts} = 1 - S'_{oirw}$, $Supp_{bs} = 1 - S_{oirw}$

For k_{row} at (Sat_{ts})

$Sat = S_w$

Two point scaling:

$Slow_{ts} = S'_{wcon}$, $Slow_{bs} = S_{wcon}$, $Supp_{ts} = 1 - S'_{orw}$, $Supp_{bs} = 1 - S_{orw}$

Three point scaling:

When $Sat_{bs} < S_{wcrit}$: $Slow_{ts} = S'_{wcon}$, $Slow_{bs} = S_{wcon}$, Suppts = S'_{wcrit}, $Supp_{bs} = S_{wcrit}$

When $Sat_{bs} > S_{wcrit}$: $Slow_{ts} = S'_{wcrit}$, $Slow_{bs} = S_{wcrit}$, $Supp_{ts} = 1 - S'_{orw}$, $Supp_{bs} = 1 - S_{orw}$

For k_{rog} at (Sat_{ts})

$Sat = S_l$

Two point scaling:

$Slow_{ts} = S'_{lrg}$, $Slow_{bs} = S_{lrg}$, $Supp_{ts} = 1 - S'_{gcon}$, $Supp_{bs} = 1 - S_{gcon}$

Three point scaling:

When $Sat_{bs} < 1 - S_{gcrit}$: $Slow_{ts} = S'_{lrg}$, $Slow_{bs} = S_{lrg}$, $Supp_{ts} = 1 - S'_{gcrit}$, $Supp_{bs} = 1 - S_{gcrit}$

When $Sat_{bs} > 1 - S_{gcrit}$: $Slow_{ts} = 1 - S'_{gcrit}$, $Slow_{bs} = 1 - S_{gcrit}$, $Supp_{ts} = 1 - S'_{gcon}$, $Supp_{bs} = 1 - S_{gcon}$

For k_{rg} at (Sat_{ts})

$Sat = S_l$

Two point scaling:

$Slow_{ts} = S'_{lcon}$, $Slow_{bs} = S_{lcon}$, $Supp_{ts} = 1 - S'_{gcrit}$, $Supp_{bs} = 1 - S_{gcrit}$

Three point scaling:

When $Sat_{bs} < S_{lrg}$: $Slow_{ts} = S'_{lcon}$, $Slow_{bs} = S_{lcon}$, $Supp_{ts} = S'_{lrg}$, $Supp_{bs} = S_{lrg}$

When $Sat_{bs} > S_{lrg:}$ $Slow_{ts} = S'_{lrg}$, $Slow_{bs} = S_{lrg}$, $Supp_{ts} = 1 - S'_{gcrit}$, $Supp_{bs} = 1 - S_{gcrit}$

Table Based Scaling: Saturation Endpoints for Relative Permeability (Optional) *TSWCON, *TSWCRIT, *TSOIRW, *TSORW, *TSGCON, *TSGCRIT, *TSLCON, *TSORG, TSOIRG

PURPOSE:

These keywords specify the end points of the relative permeability curve of each relative permeability table. The entered relative permeability curves (via keywords *SWT/*SGT) are scaled (overridden) based on these end points.

FORMAT:

*TSWCON	TS_{wcon}
*TSWCRIT	TS_{wcrit}
*TSOIRW	TS_{oirw}
*TSORW	TS_{orw}
*TSGCON	TS_{gcon}
*TSGCRIT	TS_{gcrit}
*TSLCON	TS_{lcon}
*TSORG	TS_{org}
*TSOIRG	TS_{oirg}

DEFINITIONS:

TS_{wcon}

Used to specify the connate water saturation, S_{wcon} , of a relative permeability table. This value overrides the value specified with the *SWT table, the first saturation entry in the *SWT table. 1-TS_{wcon} is the maximum oil saturation. Oil wet: Connate oil saturation. 1- TS_{wcon} is the maximum water saturation.

TS_{wcrit}

Used to specify the critical water saturation, S_{wcrit} , of a relative permeability table. This value overrides the value specified with the *SWT table. TS_{wcrit} is the saturation at which water first becomes mobile in the two phase oil-water system. It is always equal to or greater than TS_{wcon}.

Oil wet: Critical oil saturation. It is the saturation at which oil first becomes mobile in the two phase oil-water system. It is always equal to or greater than TS_{wcon}.

TS_{oirw}

Used to specify the irreducible oil saturation, S_{oirw} , of a relative permeability table. This value overrides the value specified with the *SWT table, the last water saturation entry in the *SWT table. TS_{oirw} is the minimum oil saturation. 1- TS_{oirw} is the maximum water saturation.

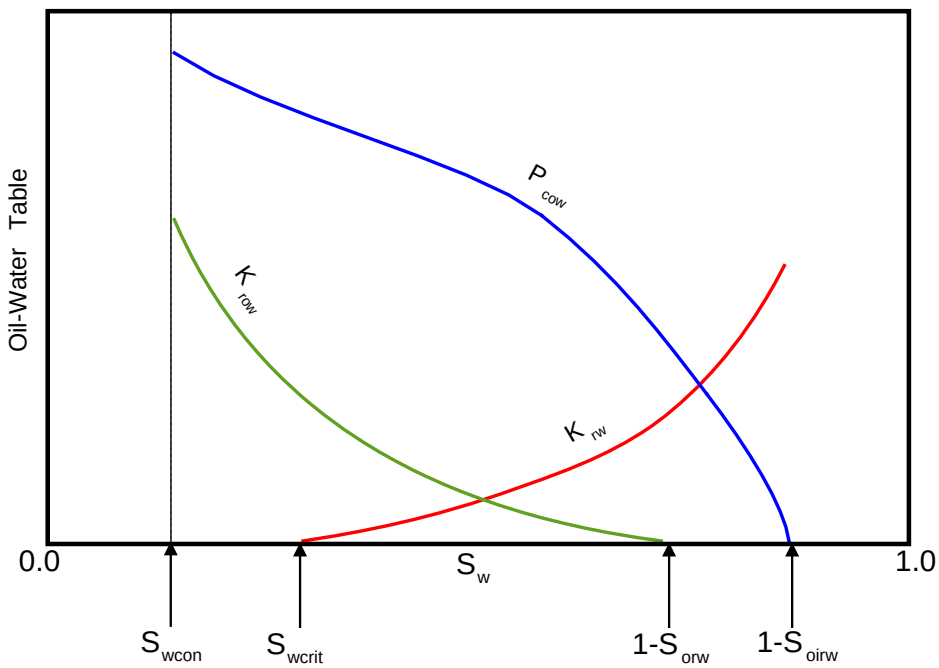
Oil wet: Irreducible (or connate) water saturation of a table. It is the last oil saturation entry in the *SWT table. TS_{oirw} is the minimum water saturation.

TS_{orw}

Used to specify the residual oil saturation, S_{orw} , of a relative permeability table. This value overrides the value specified with the *SWT table. TS_{orw} is the oil saturation at which oil first becomes mobile in the two phase oil-water table. It is always equal to or greater than TS_{oirw}.

Oil wet: Residual water saturation of a grid block. TS_{orw} is the water saturation at which water first becomes mobile in the two phase oil-water table. It is always equal to or greater than $TS_{oirw.}$

Note: Graph shows the water wet case



TS_{gcon}

Used to specify the connate gas saturation, S_{gcon} , of a relative permeability table. This value overrides the value specified with the *SGT/*SLT table. TS_{gcon} is the minimum gas saturation in the two phase gas-liquid system. It corresponds to the first entry in a *SGT table.

TS_{gcrit}

Used to specify the critical gas saturation , S_{gcrit} , of a relative permeability table. This value overrides the value specified with the *SGT/*SLT table. TS_{gcrit} is the gas saturation at which gas first becomes mobile. It is always equal to or greater than TS_{gcon}.

TS_{lcon}

Used to specify the connate liquid oil saturation, S_{lcon} , of a relative permeability table. This value overrides the value specified with the

*SGT/*SLT table. TS_{lcon} includes connate water (TS_{wcon}). It is the minimum liquid saturation in the two phase gas-liquid system and corresponds to the first entry in the *SLT table and the last entry in the *SGT table.

Oil wet: Connate liquid saturation of a grid block in the gas liquid table. TS_{lcon} is the sum of connate oil saturation and irreducible water saturation in the two phase gas-liquid (connate oil + water) system. It corresponds to the first entry in the SLT table and the last entry in the SGT table.

TS_{org}

Used to specify the residual oil saturation, S_{org} , of a relative permeability table. This value overrides the value specified with the *SGT/*SLT table. It does not include connate water, thus TS_{org} + TS_{wcon} = residual liquid saturation.

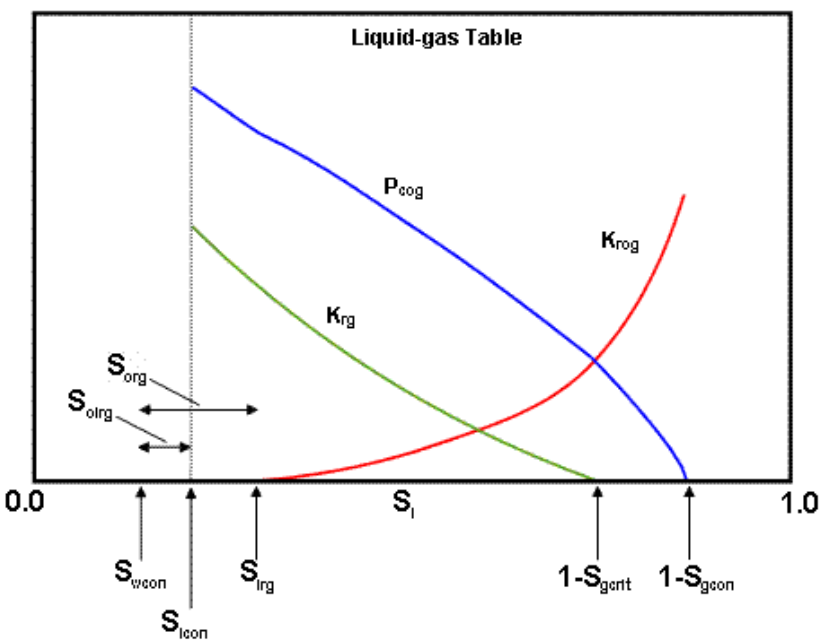
Oil wet: Residual water saturation of a table in the two phase gas-liquid (connate oil + water) system.

TS_{oirg}

Used to specify the irreducible oil saturation in a two phase gas-oil system, S_{oirg} , of a relative permeability table. This value overrides the value specified with the *SGT/*SLT table. Specify either the irreducible oil saturation (with TS_{oirg}) or connate liquid with keyword TS_{lcon}. If TS_{oirg} is specified then connate liquid is calculated as the sum of TS_{oirg} + TS_{wcon}. If TS_{lcon} is specified then TS_{oirg} is calculated as TS_{lcon} - TS_{wcon}.

Oil wet: Irreducible water saturation of a table in the two phase water-gas system.

Note: Graph shows the water wet case



where S_{lrg} is the residual liquid in the gas-liquid table.

S_{oirg} is the irreducible oil in the gas liquid table.

$$S_{lrg} = S_{org} + S_{wcon}$$

$$S_{oirg} = S_{lcon} - S_{wcon}$$

DEFAULTS:

Optional keyword. The default is to use the end points from the entered relative permeability tables via keywords *SWT/*SGT. Any or all of the above keywords may be omitted.

CONDITIONS:

These keywords, if present, must appear in the Rock-Fluid Data section.

For a data set with no *RPT keyword specified these may appear after *SWT, *SLT or *SGT relative permeability tables have been entered.

For a data set with *RPT specified but without interpolation sets these keywords may appear after *SWT, *SLT or *SGT and associated relative permeability tables have been entered for each *RPT.

For a data set with *RPT and interpolation sets specified via *KRINTRP, these keywords may appear after occurrence of *SWT, *SLT or *SGT and associated relative permeability tables for each *KRINTRP.

If for a particular grid-block both block-based (say, *BSROW) and table-based (say, *TSROW) end points are assigned, the value specified with block-based (say, *BSROW) end point will be used.

EXPLANATION:

The end point keywords indicate the input of a single real value for a particular relative permeability table. The relative permeabilities and capillary pressures are looked up on their respective tables after the lookup saturation (either S_w or S_l) have been scaled appropriately.

For the oil-water table and for the water wet case:

$$K_{rw}, S_w^{'} = S_{wcrit}^{'} + (S_w - S_{wcrit}) * \left(\frac{1 - S_{oirw}^{'} - S_{wcrit}^{'}}{1 - S_{oirw} - S_{wcrit}} \right)$$

$$K_{row}, S_w^{''} = S_{wcon}^{'} + (S_w - S_{wcon}) * \left(\frac{1 - S_{orw}^{'} - S_{wcon}^{'}}{1 - S_{orw} - S_{wcon}} \right)$$

$$P_{cow}, S_w^{'''} = S_{wcon}^{'} + (S_w - S_{wcon}) \left(\frac{1 - S_{oirw}^{'} - S_{wcon}^{'}}{1 - S_{oirw} - S_{wcon}} \right)$$

Sw is the blocks water saturation

$S_w^{'}, S_w^{''}, S_w^{'''}$ are the saturations used to interpolate the original user entered tables with respect to K_{rw}, K_{row} and P_{cow} respectively.

S_{wcrit}, S_{wcon}, S_{oirw} and S_{orw} are the user entered table end point values (from the keywords of the same name)

S'_{wcrit}, S'_{wcon}, S'_{oirw} and S'_{orw} are the original table end point values (calculated directly from the relative permeability tables)

For the gas-liquid table and for the water wet case:

$$K_{rog},\ S'_l = S'_{lrg} + (S_l - S_{lrg}) * \left(\frac{1 - S'_{lrg} - S'_{gcon}}{1 - S_{lrg} - S_{gcon}} \right)$$

$$K_{rg},\ S''_l = S'_{lcon} + (S_l - S_{lcon}) * \left(\frac{1 - S'_{lcon} - S'_{gcrit}}{1 - S_{lcon} - S_{gcrit}} \right)$$

$$P_{cog},\ S'''_l = S'_{lcon} + (S_l - S_{lcon}) \left(\frac{1 - S'_{lcon} - S'_{gcon}}{1 - S_{lcon} - S_{gcon}} \right)$$

$$S_l = S_o + S_w$$

$$S_{lrg} = S_{wcon} + S_{org}$$

S'_l, S''_l, S'''_l are the saturations used to interpolate the original user entered tables with respect to K_{rog}, K_{rg} and P_{cog} respectively.

S_{org}, S_{lcow}, S_{gcon} and S_{gcrit} are the user entered table end point values (from the keywords of the same name)

S'_{org}, S'_{lcon}, S'_{gcon} and S'_{gcrit} are the original table end point values (calculated directly from the relative permeability tables)

Note that each relative permeability curve is now scaled with respect to different end points which define the range of each curve. The table below applies to the water wet case.

K_{rw}	$S_{wcrit} \le S_w \le 1 - S_{oirw}$,	thus K_{rw} is scaled by S_{wcrit} and S_{oirw}
K_{row}	$S_{wcon} \le S_w \le 1 - S_{orw}$,	thus K_{row} is scaled by S_{wcon} and S_{orw}
P_{cow}	$S_{wcon} \le S_w \le 1 - S_{oirw}$,	thus P_{cow} is scaled by S_{wcon} and S_{oirw}
K_{rog}	$S_{lrg} \le S_l \le 1 - S_{gcon}$,	thus K_{rog} is scaled by S_{wcon}, S_{org} and S_{gcon}.
K_{rg}	$S_{lcon} \le S_l \le 1 - S_{gcrit}$,	thus K_{rg} is scaled by S_{lcon} and S_{gcrit}
P_{cog}	$S_{lcon} \le S_l \le 1 - S_{gcon}$,	thus P_{cog} is scaled by S_{lcon} and S_{gcon}

Table Based Scaling: Maximum Relative Permeability Values (Optional) *TKRWIRO, *TKROCW, *TKRGCL, *TKROGCG, *TKRWRO, *TKROCRW, *TKRGRL, *TKROGCRG

PURPOSE:

These keywords specify the vertical end point values of the relative permeability curves for each relative permeability table. The entered curves (via keywords *SWT/*SGT) curves are scaled based on these end points. Use of any of these keywords forces GEM to use the new endpoint scaling; see "Notes on Rock Fluid".

FORMAT:

*TKRWIRO	Tk_{rwiro}
*TKROCW	Tk_{rocw}
*TKRGCL	Tk_{rgcl}
*TKROGCG	Tk_{rogcg}
*TKRWRO	Tk_{rwro}
*TKROCRW	Tk_{rocrw}
*TKRGRL	Tk_{rgrl}
*TKROGCRG	Tk_{rogcrg}

DEFINITIONS:

Tk_{rwiro}

Relative permeability value of water at the irreducible oil saturation in the oil-water table (the last row in the table).

Oil wet: Relative permeability value of oil at the irreducible water saturation in the oil water table (last row in the SWT table)

Tk_{rocw}

Relative permeability value of oil at the connate water saturation in the oil-water table (the first row in the table).

Oil wet: Relative permeability value of water at the connate oil saturation in the oil water table (the first row in the table)

Tk_{rgcl}

Relative permeability value of gas at the connate liquid saturation in the gas-liquid table (the first row in the table if *SLT is used or the last row in the table if *SGT is used).

Tk_{rogcg}

Relative permeability value of oil at the minimum gas saturation in the gas-liquid table (the last row in the table if *SLT is used or the first row in the table if *SGT is used).

Oil wet: Relative permeability value of water at the minimum gas saturation in the gas-liquid table (the last row in the table if *SLT is used or the first row in the table if *SGT is used).

Critical Endpoint Values

Tk_{rwro}

Relative permeability value of water at the residual oil saturation (S_{orw}) in the oil-water table.

Oil wet: Relative permeability value of oil at the residual water saturation in the oil water table.

Tk_{rocrw}

Relative permeability value of oil at the critical water saturation (S_{wcrit}) in the oil-water table.

Oil wet: Relative permeability value of water at the residual (critical) oil saturation in the oil water table.

Tk_{rgrl}

Relative permeability value of gas at the residual liquid saturation (S_{lrg}) in the gas-liquid table (See *SWCON Keyword for definition of S_{lrg}).

Tk_{rgcrg}

Relative permeability value of oil at the critical gas saturation (S_{gcrit}) in the gas-liquid table.

Oil wet: Relative permeability value of water at the critical gas saturation in the gas-liquid table.

DEFAULTS:

Optional keyword. The default is to use the values from the relative permeability tables entered via the *SWT and *SLT or *SGT keywords. Any or all of the above keywords may be omitted.

The default relative permeability value scaling method is two-point scaling. If a critical Kr endpoint is assigned to a block, the relative permeability value will be scaled using three-point scaling; provided that the critical endpoint in the table differs from the connate value in the table or the entered critical table end point saturation is different from the entered connate table end point saturation value. Otherwise two-point scaling will be performed.

See Tutorial ‘Two and Three Point Scaling of Relative Permeability Values’ for definitions of two-point and three-point relative permeability values.

CONDITIONS:

These keywords, if present, must appear in the Rock-Fluid Data section.

For a data set with no *RPT keyword specified these may appear after *SWT, *SLT or *SGT relative permeability tables have been entered.

For a data set with *RPT specified but without interpolation sets these keywords may appear after *SWT, *SLT or *SGT and associated relative permeability tables have been entered for each *RPT.

For a data set with *RPT and interpolation sets specified via *KRINTRP, these keywords may appear after occurrence of *SWT, *SLT or *SGT and associated relative permeability tables for each *KRINTRP.

EXPLANATION:

The end point keywords indicate the input of a single real value corresponding to a particular relative permeability table. For the critical endpoint value scaling (vertical scaling) to be enabled it is crucial that either the connate/irreducible endpoint saturations in the table are different from the critical/residual endpoint saturations or the table based entered connate/irreducible endpoint saturations are different from the table entered critical/residual saturations.

That is: $S_{orw} > S_{oirw}$, $S_{wcrit} > S_{wcon}$, $S_{lrg} > S_{lcon}$, and $S_{gcrit} > S_{gcon}$ in the entered table (via keyword *SWT/*SGT/*SLT) for a particular table or the table end point values are different (entered via keywords *TSWCON, *TSWCRIT etc).

If the above is not true for a particular curve, three-point Kr value scaling will not be performed for that curve.

If the *3PTSCALING keyword is entered then 3 point horizontal scaling is done for all four relative permeability curves: k_{rw}, k_{row}, k_{rg} and k_{rog}. If the *3PTSCALING keyword is NOT specified then the 3 point horizontal scaling will be done for a given curve if either the keyword to turn on 3 point horizontal scaling is specified (*3PTSCAL_KRW etc) or the corresponding 3 point vertical scaling keyword is specified. For example if *TKRWRO is specified then 3 point horizontal scaling is done for the water phase or the k_{rw} curve.

If a grid block, ips, is assigned a rock type via keyword *RTYPE with only one interpolation table, and both a block based end point (say, *BSORW) and table-based end point (say, *TSORW) are specified, then value specified with block-based end point will be used in scaling, $S_{orw2} = BS_{orw}(ips)$.

If a grid block, ips, is assigned a rock type via keyword *RTYPE with two or more interpolation tables, and both a block based end point (say, *BSORW) and table-based end point (say, *TSORW) are specified, then value used in scaling for a particular grid block, ips, and particular table j, will be calculated based on the following equation:

$$S_{orw_2} = BS_{orw}(ips)\frac{TS_{orw}(j)}{T_{sorw}(1)}$$

Examples below clarify the concept.

Examples:

Case 1: Single relative permeability table is entered for a given rock type

In this scenario all grid blocks are assigned to rock type 1 via keyword *RTYPE. Rock type 1, keyword *RPT 1 has only one relative permeability table specified (via keywords *SWT/*SGT). This also implies that keyword *KRINTRP is NOT specified, as this keyword is used in entering cases where multiple interpolation tables are present. In this case the value specified for the grid block overrides the value specified for the table.

```
*RPT 1
*SWT
..
..
*SGT
..
..
*TSORW 0.2
*BSORW *CON 0.3
*RTYPE *CON 1
```

The end point $Sorw_2$ value will be 0.3 as specified by *BSORW and the value of 0.2 specified via *TSORW will be ignored.

Case 2: Multiple relative permeability tables are entered for a given rock type

In this scenario all grid blocks are assigned to rock type 1 via keyword *RTYPE. Rock type 1, keyword *RPT 1 has three relative permeability table specified (via keyword *KRINTRP).

```
*RPT 1 **(rock type #1)
*INTCOMP *EQVFRIEX 'Na-X'
*KRINTRP 1  ** (table #1)
*INTCOMP_VAL 0.35
*SWT
..
..
*SGT
..
..
*TSORW 0.1
*KRINTRP 2  ** (table #2)
*INTCOMP_VAL 0.25
*SWT
..
..
*SGT
..
..
*TSORW 0.2
*KRINTRP 3  ** (table #3)
*INTCOMP_VAL 0.05
*SWT
..
..
*SGT
..
..
*TSORW 0.25

*BSORW *CON 0.3
*RTYPE *CON 1
```

This is a case with three relative permeability interpolation sets specified. Table #1 has a critical (or residual) oil saturation (*TSORW keyword) of 0.1 specified, table #2 has a critical oil saturation value of 0.2 specified and table #3 has a critical oil saturation of 0.25 specified. The grid block end point of 0.3 (*BSORW keyword) is specified for the critical oil saturation.

Suppose for a particular grid block (say block # 5) the value of the interpolating parameter (equivalent fraction of Na-X) is equal to 0.2. This implies that tables two and three are selected for relative permeability interpolation for grid block #5.

The end point for table #2 is calculated as follows:

$$S_{orw2} = BS_{orw}(5)\frac{TS_{orw}(2)}{TS_{orw}(1)}$$

Since $BS_{orw}(5)$ is 0.3, for j equal to 2, $TS_{orw}(2)$ is 0.2 and $TS_{orw}(1)$ is 0.1, giving a value of 0.3*0.2/0.1 or 0.6 for S_{orw2} .

The end point for table #3 is calculated as follows:

$$S_{orw2} = BS_{orw}(5)\frac{TS_{orw}(3)}{TS_{orw}(1)}$$

Since $BS_{orw}(5)$ is 0.3, for j equal to 3, $TS_{orw}(3)$ is 0.25 and $TS_{orw}(1)$ is 0.1, giving a value of 0.3*0.25/0.1 or 0.75 for S_{orw2} .

Residual Oil Saturation Calculation for STONE 1 Relative Permeability Model (Optional) *SOMWAT, *SOMGAS, *SOMWATI, *SOMGASI

PURPOSE:

Keywords *SOMWAT, *SOMGAS, *SOMGASI, *SOMWATI allow the user to enter a table of residual oil saturation as a function of water/gas saturation to be used in the STONE 1 three phase oil relative permeability model.

FORMAT:

```
*SOMWAT | *SOMWATI
** Sw          Som
Sw1            Som1
Sw2            Som2
.              .
.              .
*SOMGAS | *SOMGASI
**Sg           Som
Sg1            Som1
Sg2            Som2
.              .
.              .
```

DEFINITIONS:

*SOMWAT

This keyword is used to enter a two column table of residual oil saturation (Som) as a function of the water saturation (Sw) for the drainage case. The residual oil saturation is used in calculating the oil phase relative permeability when the STONE 1 three phase relative permeability model is in effect (see *KROIL *STONE1).

If hysteresis is turned on for the oil curve for either the two phase oil-water system (k_{row}) or two phase oil-gas system (k_{rog}) then *SOMWAT applies to the drainage case, that is, where the oil saturation is increasing.

Following this keyword a two column table of water saturation (Sw) versus oil residual saturation (Som) is required. The table must start on a new line. The first row of this table should have the connate water saturation in the first column and the residual oil saturation (Som) as specified in the oil-gas table (*SGT or *SLT) or *SORG keyword in the second column. The last row of this table should have the water saturation equal to $1 - S_{orw}$ and the residual oil saturation equal to S_{orw}. S_{orw} is the residual oil saturation specified via the *SWT table or the *SORW keyword. The table must contain at least two rows.

Use keyword *SOMWATI to enter a table of water saturation versus residual oil saturation for the imbibition case, that is, where the oil saturation is decreasing.

*SOMWATI

This keyword is used to enter a two column table of residual oil saturation (Som) as a function of the water saturation (Sw) for the imbibition case. The residual oil saturation is used in calculating the oil phase relative permeability when the STONE 1 three phase relative permeability model is in effect (see *KROIL *STONE1).

If hysteresis is turned on for the oil curve for either the two phase oil/water system (k_{row}) or two phase oil/gas system (k_{rog}) then values specified via the *SOMWATI may be used in the calculation of the residual oil saturation. Use the keyword *SOMWATI to enter a table of water saturation versus residual oil saturation for the imbibition case, that is, where the oil saturation is decreasing.

If a grid block is on a scanning curve and hysteresis is turned on (by specifying *HYSKROW and/or *HYSKROG) then values read off from the *SOMWATI table as well as the *SOMWAT table are used to calculate a final value of the residual oil saturation as shown below.

Following this keyword a two column table of water saturation (Sw) versus oil residual saturation (Som) is required. The table must start on a new line. The first row of this table should have the connate water saturation in the first column and the residual oil saturation specified via the *HYSKROG keyword curve in the second column.

If k_{rog} hysteresis is not turned on, then the S_{org} value should be equal to the value specified via the oil-gas table (*SGT or *SLT) in the second column. The last row of this table should have the water saturation equal to $1 - S_{orw}$ and the residual oil saturation equal to S_{orw}.

If k_{row} hysteresis is turned on then S_{orw} is the residual oil saturation specified via the *HYSKROW keyword. If hysteresis is not turned on then the S_{orw} value is specified via the *SWT table. The table must contain at least two rows.

*SOMGAS

This keyword is used to enter a two column table of residual oil saturation (Som) as a function of the gas saturation (Sg) for the imbibition case. The residual oil saturation is used in calculating the oil phase relative permeability when the STONE 1 three phase relative permeability model is in effect (see *KROIL *STONE1).

If hysteresis is turned on for the oil curve for either the two phase oil/water system (k_{row}) or two phase oil-gas system (k_{rog}) then *SOMGAS applies to the drainage case, that is, where the oil saturation is increasing.

Following this keyword a two column table of gas saturation (Sg) versus oil residual saturation (Som) is required. The table must start on a new line. The first row of this table should have the connate gas saturation in the first column and the residual oil saturation specified in the oil-water table (*SWT) in the second column. The last row of this table should have the gas saturation equal to $1 - S_{org} - S_{wcot}$ and the residual oil saturation equal to S_{org}. S_{org} is the residual oil saturation specified via the *SGT or *SLT table. The table must contain at least two rows.

Use the keyword *SOMGASI to enter a table of gas saturation versus residual oil saturation for the imbibition case, that is, where the oil saturation is decreasing.

*SOMGASI

This keyword is used to enter a two column table of residual oil saturation (Som) as a function of the gas saturation (Sg) for the imbibition scenario, where hysteresis is specified and the oil saturation is decreasing. The residual oil saturation is used in calculating the oil phase relative permeability when the STONE 1 three phase relative permeability model is in effect (see *KROIL *STONE1).

If hysteresis is turned on for the oil curve for either the two phase oil-water system (k_{row}) or two phase oil-gas system (k_{rog}) then values specified via the *SOMGASI may be used in the calculation of the residual oil saturation.

If a grid block is on a scanning curve and hysteresis is turned on (by specifying *HYSKROW and/or HYSKROG) then values read off from the SOMGASI table as well as the SOMGAS table are used to calculate a final value of the residual oil saturation as shown below.

Following this keyword a two column table of gas saturation (Sg) versus oil residual saturation (Som) is required. The table must start on a new line. The first row of this table should have the connate gas saturation in the first column and the residual oil saturation specified via the HYSKRO curve in the second column.

If k_{row} hysteresis is not turned on, then the S_{orw} value should be equal to the value specified via the oil/water table (*SWT) in the second column. The last row of this table should have the gas saturation equal to $1 - S_{org} - S_{wcot}$ and the residual oil saturation equal to S_{org}.

If k_{rog} hysteresis is turned on then S_{org} is the residual oil saturation specified via the *HYSKROG keyword. If hysteresis is not turned on then the S_{org} value is specified via the *SGT or *SLT table. The table must contain at least two rows.

DEFAULTS:

These keywords are optional. If neither *SOMWAT, *SOMGAS appears in the data set, then the default method of calculating the residual oil saturation is by linear average of S_{orw} and S_{org} based on the grid block gas saturation. See inheritance criteria below on how SOMWAT,

SOMGAS tables are applied to rock-types (interpolation sets) that do not have these keywords specified.

CONDITIONS:

Only one of *SOMWAT or *SOMGAS can appear in the data set.

For *SOMWATI to be specified, *SOMWAT must also be specified.

For *SOMGASI to be specified, *SOMGAS must also be specified.

Either k_{row} hysteresis (keyword *HYSKROW) and/or k_{rog} hysteresis (keyword *HYSKROG) must be turned on for the *SOMWATI or *SOMGASI table to be utilized.

INHERITENCE:

If the table corresponding to keywords *SOMWAT, *SOMGAS, *SOMWATI or *SOMGASI is missing for a particular rock type/set number, then that rock type/set number will inherit the previous table specified in the sequence.

Example 1: single set per rock type

```
*RPT 1
*RPT 2
     *SOMWAT **(table (a))
*RPT 3
*RPT 4
     *SOMWAT **(table (b))
*RPT 5
```

In this case grid blocks assigned to rock type 1 (no *SOMWAT table specified) will use the default correlation to calculate S_{om}. Grid blocks assigned to rock type 2 will use the entered *SOMWAT table (a) to calculate S_{om}. Rock type 3 does not have a *SOMWAT table specified, so it will inherit the table specified for rock type 2, table (a). Therefore grid blocks assigned to rock type 2 and 3 will use the same *SOMWAT table (a). Grid blocks assigned to rock type 4 will use the assigned *SOMWAT table (b). Rock type 5 does not have a *SOMWAT table specified, so it will inherit the table specified for rock type 4, table (b). Therefore grid blocks assigned to rock type 4 and 5 will use the same *SOMWAT table (b).

Example 2: Multiple sets per rock type

```
*RPT 1
    *KRINTRP 1
    *KRINTRP 2
*RPT 2
     *KRINTRP 1
     *SOMWAT **(table (a))
     *KRINTRP 2
     *SOMWAT **(table (b))
*RPT 3
     *KRINTRP 1
     *KRINTRP 2
```

In this case there are three rock types (RPT 1-3) each with two sets (KRINTRP 1 and 2). In this case grid blocks assigned to rock type 1, sets 1 and 2 (no SOMWAT table specified) will use the default correlation to calculate S_{om}. Grid blocks assigned to rock type 2 will use the entered SOMWAT tables (a) and (b) to calculate S_{om}. Rock type 3, sets 1 and 2 do not have a *SOMWAT table specified, so will inherit the last table specified, that is table (b), for both

sets 1 and 2. Therefore grid blocks assigned to rock type 3 will use the *SOMWAT table (b) for both interpolation sets 1 and 2.

EXPLANATION:

In Stone's first model (STONE1), as modified by Aziz and Settari (1979), relative permeability to oil, k_{ro} is computed as

$$k_{ro} = \frac{S_o^* k_{row} k_{rog}}{k_{rocw}\left(1 - S_w^*\right)\left(1 - S_g^*\right)}$$

where

$$S_o^* = \left(S_o - S_{om}\right)/\left(1 - S_{wcrit} - S_{om}\right)$$
$$S_w^* = \left(S_w - S_{wcrit}\right)/\left(1 - S_{wcrit} - S_{om}\right)$$
$$S_g^* = S_g/\left(1 - S_{wcrit} - S_{om}\right)$$

for the "minimal" value S_{om} of the oil saturation, GEM uses the linear function of S_g proposed by Fayers and Matthews (SPEJ April 1984, 224-232) as the default method:

$$S_{om}(S_g) = (1 - a_{S_g})S_{orw} + a_{S_g} S_{org}$$

where

$$a_{S_g} = S_g/\left(1 - S_{wcrit} - S_{org}\right)$$

As an alternative the user can choose the functional form of the dependence of S_{om} on either the water saturation (S_w) or gas saturation (S_g). This can be done by entering a table of S_{om} versus S_w (*SOMWAT keyword) or a table of S_{om} versus S_g (*SOMGAS keyword). If either *SOMWAT or *SOMGAS is entered then table with linear interpolation is used to look up S_{om} as a function of the water or gas saturation instead of the Fayers and Matthews equation shown above.

If hysteresis is turned on for either the k_{row} curve or the k_{rog} curve or both then the table entered with *SOMWAT or *SOMGAS becomes associated with drainage, that is where the oil saturation is increasing. In this case another table corresponding to imbibition can also be entered with the keywords *SOMWATI or *SOMGASI. Imbibition implies the oil saturation is decreasing.

Case I – Both k_{row} and k_{rog} hysteresis are turned on and the grid block is on a scanning curve for both k_{row} and k_{rog} cases. *SOMWATI table is entered. Calculate S_{om} as follows, shown for *SOMWAT or *SOMWATI table entries.

$$S_{om} = S_{om}^d + F\left(S_{om}^i - S_{om}^d\right)$$

$$F = F_{sw} F_w + \left(1 - F_{sw}\right) F_g$$

$$F_{sw} = \frac{S_w - S_{wcon}}{1 - S_{wcon} - S_{orw}^h}$$

$$F_w = \frac{S^i_{orw} - S^h_{orw}}{S^i_{orw} - S^d_{orw}}$$

$$F_g = \frac{S^i_{org} - S^h_{org}}{S^i_{org} - S^d_{org}}$$

$$S^h_{orw} = S^d_{orw} + \frac{(S^m_{orw} - S^d_{orw})}{1 + C_{sorw}(S^m_{orw} - S^d_{orw})}$$

$$S^h_{org} = S^d_{org} + \frac{(S^m_{org} - S^d_{org})}{1 + C_{sorg}(S^m_{org} - S^d_{org})}$$

where

S_{om}	Residual oil saturation (ROS), used in STONE 1 model described above
S^d_{om}	ROS drainage, looked up from the entered *SOMWAT table as a function of grid block water saturation S_w
S^i_{om}	ROS imbibition, looked up from the entered *SOMWATI table as a function of grid block water saturation S_w
S^h_{orw}	ROS k_{row} curve, calculated from the entered Land's constant and the drainage to imbibition reversal saturation, S^m_{orw}
S^h_{org}	ROS k_{rog} curve, calculated from the entered Land's constant and the drainage to imbibition reversal saturation, S^m_{org}
S^d_{orw}	ROS k_{row} drainage curve, inferred from the entered *SWT table
S^i_{orw}	ROS k_{row} imbibition curve, entered via the *HYSKROW keyword
S^d_{org}	ROS k_{rog} drainage curve, inferred from the entered *SGT or *SLT table
S^i_{org}	ROS k_{rog} imbibition curve, entered via the *HYSKROG keyword
S^m_{org}	ROS k_{rog} imbibition curve, entered via the *HYSKROG keyword
S^m_{orw}	Oil saturation corresponding to drainage to imbibition reversal saturation, k_{rog} curve
C_{sorw}	Land's constant k_{row} curve, calculated internally based on S_{ormax} (or S_{wrmax}) entered via the *HYSKROW keyword
C_{sorg}	Land's constant k_{rog} curve, calculated internally based on S_{orgmax} entered via the *HYSKROG keyword
S_{wcon}	Connate water saturation

Case II – Only k_{row} is hysteresis is turned on and the grid block is on a scanning curve or both k_{row} and k_{rog} hysteresis are turned on but the block is on a scanning curve for k_{row} and drainage for k_{rog}.

$$S_{om} = S_{om}^{d} + F\left(S_{om}^{i} - S_{om}^{d}\right)$$

$$F = F_{w} = \frac{S_{orw}^{i} - S_{orw}^{h}}{S_{orw}^{i} - S_{orw}^{d}}$$

Case III – Only k_{rog} is hysteresis is turned on and the grid block is on a scanning curve or both k_{row} and k_{rog} hysteresis are turned on but the block is on a scanning curve for k_{rog} and drainage for k_{row}.

$$S_{om} = S_{om}^{d} + F\left(S_{om}^{i} - S_{om}^{d}\right)$$

$$F = F_{g} = \frac{S_{org}^{i} - S_{org}^{h}}{S_{org}^{i} - S_{org}^{d}}$$

Saturation Endpoint Interpolation (Optional) *INTERP_ENDS

PURPOSE:

Specify the method for treating saturation endpoints when interpolating between curves as directed by keywords *KRINTRP.

FORMAT:

*INTERP_ENDS (*ON | *OFF)

DEFINITIONS:

*ON

Use a two stage method for interpolation between two sets of rock-fluid curves. First, the saturation (horizontal axis) endpoints are interpolated separately and applied to the two normalized curves. Then the curve values (vertical axes) are obtained at the required saturation.

*OFF

Use a single-stage method. The curve values are obtained at the required saturation, for each set of curves separately, which are then interpolated to give the final result.

DEFAULTS:

If *INTER_ENDS is absent then *OFF is assumed.

If *INTERP_ENDS appears without *ON or *OFF, *ON is assumed.

CONDITIONS:

The interpolation method specified by *INTERP_ENDS or its default applies to all rock types, that is all rock types use the same method.

EXPLANATION:

In general, the pair of relative permeability curves involved in the interpolation possess different shapes and saturation endpoints (S_{wcrit}, S_{orw}, etc.). If the model is operated in the vicinity of those saturation endpoints, it will be important to account for the change of saturation endpoints in order to correctly estimate the phase mobility in the region.

The following will discuss how the two-stage interpolation is done for curves with different endpoints and how the three-phase relative permeabilities are computed. The discussion will focus on the oil relative permeability k_{ro} for a water-wet system, which equals k_{row} in a water/oil two-phase system. The interpolation of other relative permeabilities is treated similarly.

Interpolative Values of Saturation Endpoints

Denoting the two sets of curves as A and B, the residual oil saturation of set A, S^A_{orw}, and S^B_{orw} of set B, S^B_{orw}, the interpolation required is based on the phase molar fraction x of a component. At a given molar fraction x^i, the saturation endpoint S^i_{orw} is computed by a linear interpolation of Eq. (E1) below.

$$S^i_{orw} = (1-\omega_i)S^A_{orw} + \omega_i S^B_{orw} \qquad \text{(E1)}$$

where ω_i is the weighting factor computed from eq. (E2),

$$\omega_i = \frac{x_i - x^A}{x^B - x^A} \qquad \text{(E2)}$$

and x^A and x^B are the value of mole fraction x specified for sets A and B (dtrapn in this case, see *DTRAPN for keyword details). Interpolation parameter ω_i is constrained to the range [0,1], so S^i_{orw} will lie between S^A_{orw} and S^B_{orw} inclusive.

Interpolation of Relative Permeability

If relative permeabilities obtained from both sets A and B are nonzero, that is, the grid oil saturation S_o (=1-S_w) is larger than both S^A_{orw} and S^B_{orw} the interpolative oil relative permeability is computed by a formula similar to the saturation endpoint (eq. (E1)):

$$k^i_{row} = (1-\omega_i)k^A_{row} + \omega_i k^B_{row} \qquad \text{(E3)}$$

or, if the curvature change parameter 'ocrv' is entered, by the formula given in the overview section "Interpolation of Relative Permeability and Capillary Pressure"

$$k^i_{row} = (1-\omega_i^{ocrv})k^A_{row} + \omega_i^{ocrv} k^B_{row}$$

and the following condition is imposed: $k^i_{row} = k^A_{row}$ if $\omega_i < 0$ and $k^i_{row} = k^B_{row}$ if $\omega_i > 1$.

If oil saturation S_o is less than both S^A_{orw} and S^B_{orw}, it is obvious that $k^i_{row} = 0$.

For the situation where oil saturation S_o lies in between S^A_{orw} and S^B_{orw}, interpolation parameter X^C given by eq. (E4) corresponds to an intermediate curve which has an endpoint S_o.

$$x^c = x^A + \left(\frac{S_o - S^A_{orw}}{S^B_{orw} - S^A_{orw}}\right)\left(x^B - x^A\right) \qquad \text{(E4)}$$

For the case of $S^B_{orw} < S^A_{orw}$ and $x^B > x^A$, we have $k^i_{row} = 0$ when $x^i \le x^c$. For $x^i > x^c$ the interpolation will take the form

$$k^i_{row} = \left(\frac{x^i - x^c}{x^B - x^c}\right)k^B_{row} \qquad \text{(E5)}$$

For the opposite case of $S^B_{orw} > S^A_{orw}$ and $x^B < x^A$, there is an explanation and expression similar to eq. (E5) above.

Two-Stage Interpolation and Middle Phase Relative Permeability

In a three-phase flow situation, the relative permeability of middle phase (k_{ro} if water-wet and k_{rw} if oil-wet) will be calculated using one of the four-phase models (Stone model 1, Stone model 2 or Baker's isoperm or Segregated).

All three-phase models can be used with the two-stage interpolation method. In the case of Stone model 2 (*STONE2) will conduct the interpolation on each pair of two-phase curves and the middle-phase relative permeability is computed using interpolative two-phase values using the STONE2 correlation. In the other cases the final oil phase relative permeability (for

water wet as an example) is calculated by end point interpolation using the residual oil saturation as calculated for STONE1 the bounding values of the interpolating parameter and the oil saturation as the interpolating parameter:

For the water wet case:

$$S_{om}^{A} = S_{orw}^{A} + (S_{lr}^{A} - S_{wc}^{A} - S_{orw}^{A})\frac{S_g}{(1 - S_{lr}^{A})}$$

$$S_{om}^{B} = S_{orw}^{B} + (S_{lr}^{B} - S_{wc}^{B} - S_{orw}^{B})\frac{S_g}{(1 - S_{lr}^{B})}$$

Where S_{lr}^{A} = the residual liquid saturation, $S_{org}^{A} + S_{wc}^{A}$

S_{wc}^{A} = the critical water saturation

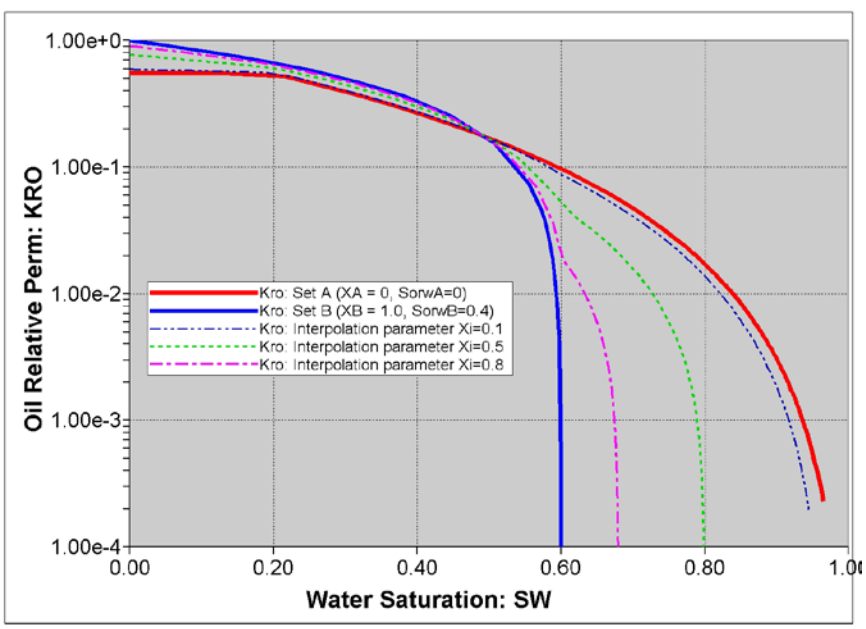
S_{orw}^{A} = the residual (critical) oil saturation oil/water table

S_g = the gas saturation in the block

S_{org}^{A} = the residual (critical) oil saturation oil/gas table

Example:

The graph below illustrates the behavior of oil relative permeability at different interpolation parameter x^i. The two sets of curves of user input have residual oil saturation $S_{orw}^{A} = 0$ and $S_{orw}^{B} = 0.4$. The interpolation parameters at the sets are $x^A = 0$ and $x^B = 0$. As interpolation parameter x^i increases, the interpolative curves approach to set B from set A but all curves are bounded by set A and B. The endpoint variations at different x^i are also shown clearly.



Initial Conditions

Notes on Initial Conditions

GEM currently provides ten different ways to initialize the reservoir.

1. *VERTICAL *BLOCK_CENTER *WATER_OIL, formerly *VERTICAL *ON, which is still accepted by GEM. Gravity initialization for under-saturated reservoirs. In this approach the water-oil contact depth, together with a reference pressure at a reference depth are specified. Global composition for each grid block must be assigned by the user. Saturations for a whole grid block are set equal to their values at the grid block center.

   Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and water saturations. So and Sg are determined by flash calculations. If the flash determines that a grid block contains both oil and gas, no special treatment is applied. In this case, drastic fluid rearrangements may occur at the first time-step when gases move upward and oil downward. This option should be used only when one is certain that the pressures and compositions provided will result in only water and oil phases being present in the reservoir.

   For this approach use the following keywords:

   *DWOC, *REFDEPTH, *REFPRES, *ZGLOBALC (or *ZGLOBAL) and *SWOC.

2. *VERTICAL *BLOCK_CENTER *WATER_GAS, formerly *VERTICAL *GASCAP, which is still accepted by GEM, with water-oil and gas-oil contact depths set equal. Gravity initialization for reservoirs containing only water and gas phases. In this approach the water-gas contact depth, together with a reference pressure at a reference depth are specified. A gas phase composition must also be specified. Saturations for a whole grid block are set equal to their values at the grid block center. If the reference depth does not lie in the gas zone, *ZOIL may be required.

   Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and water saturations. Sg is then determined by subtraction.

   For this approach use the following keywords:

   *DWGC, *REFDEPTH, *REFPRES, *ZGAS, (*ZOIL) and *SWOC.

3. *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS, formerly *VERTICAL *GASCAP, which is still accepted by GEM. Gravity initialization for saturated reservoirs. In this approach, a gas cap in equilibrium with an oil zone is assumed. The water-oil contact and gas-oil contact depth, together with a reference pressure at a reference depth must be specified. Two sets of compositions, one representing the averaged oil zone composition and the other representing the averaged gas

zone composition, are required. (Note that the simulator will attempt to determine a missing *ZGAS, or *ZOIL, internally from a saturation pressure calculation if the reference depth is in the oil, or gas, zone, respectively.) Saturations for a whole grid block are set equal to the value at the block center.

Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and all oil, gas and water saturations. Flash calculations are performed to determine the oil phase and gas phase compositions. Global composition is calculated by mixing the oil and gas phase by the gravity- capillary-equilibrium determined saturations. Thus the global composition in the reservoir may vary with depth.

For this approach use the following keywords:

*DWOC, *DGOC, *REFDEPTH, *REFPRES, *ZOIL, *ZGAS and *SWOC.

To use this option to initialize an under-saturated reservoir, set the gas-oil-contact depth to be less than the depth of the top of the reservoir. The result of this water-oil initialization will be somewhat different from that obtained under the *VERTICAL *BLOCK_CENTER *WATER_OIL option listed above if the composition is such that a flash predicts a gas phase somewhere in the reservoir; in such a case the *BLOCK_CENTER *WATER_OIL option initializes with non-zero gas saturations, whereas the present option with DGOC above the reservoir top forces zero gas saturations throughout the reservoir.

To use this option to initialize a water-gas reservoir, set the water-oil contact depth equal to the gas-oil contact depth. The result in this case will be the same as that obtained under the *BLOCK_CENTER *WATER_GAS option listed above.

To ensure depth-varying composition for all grid blocks, enter the saturated oil composition at the bottom of the reservoir as the oil zone composition; and enter the saturated gas composition at the gas-oil contact as the gas-zone composition. See the explanations of *ZOIL and *ZGAS for more information.

4. *VERTICAL *BLOCK_CENTER *COMP, formerly *VERTICAL *COMP, which is still accepted by GEM. Gravity initialization for either saturated or under-saturated reservoirs with user-specified variations of composition with depth. In this approach the water-oil contact depth, together with a reference pressure at a reference depth are specified. A table of global composition vs. depth must be entered by the user. The user must also enter a critical depth; at depths below the critical depth, single hydrocarbon phase blocks are identified as containing oil, while at depths above the critical depth, single hydrocarbon phase blocks are identified as containing gas. The composition vector for a whole grid block is set equal to its value at the grid block center.

   Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and water saturations. Global composition for each grid block is determined from the blocks depth using the composition vs. depth table. So and Sg are then calculated from flash calculations. If a grid block contains both oil and gas, no special treatment is applied. Thus, in this case, drastic fluid rearrangements may occur at the first time-step when gases moves to the top of the reservoir and oil sinks to the bottom.

   For this approach use the following keywords:

   *DWOC, *REFDEPTH, *REFPRES, *SWOC, *CDEPTH and *ZDEPTH.

5. *VERTICAL *DEPTH_AVE *WATER_OIL . Gravity initialization for an under-saturated reservoir with saturations averaged over the depth interval covered by a grid block. In this approach, for example, if a grid block has its block center slightly below the water-oil contact depth, the water saturation assigned to the block is the average over the block volume of the local saturations, and not simply the water saturation value below the water-oil contact.

   For this option, the water-oil contact depth, together with a reference pressure and a reference depth must be specified. Note that these are the same data required when using option (3) above (*VERTICAL *BLOCK_CENTER *GASCAP) for an undersaturated reservoir, except that for the present option it is not necessary to specify a gas-oil contact depth above the reservoir top.

   The composition can be specified by either of the three options, i.e., via

   (a) A composition vector representing the averaged oil zone composition (*ZOIL). or

   (b) A composition variation with depth (*ZDEPTH) or

   (c) Global composition of each grid block (*ZGLOBAL or *ZGLOBALC)

   When this initialization is performed, a table is constructed giving oil phase pressure, water-oil capillary pressure, and several other properties as functions of depth. This table is constructed by Runge-Kutta integration of the equation determining the hydrostatic pressure. Water saturation is then determined as a function of depth by inverting the table which gives water-oil capillary pressure as a function of saturation. Block water saturations are then determined by numerical quadrature of the local water saturation.

   For *DEPTH_AVE initialization use of *SWINIT is allowed.

   In the Gas Cap, *SWINIT can vary between zero and the critical (not connate) water saturation. If *SWINIT greater than critical water saturation is found, Sw in that block will be set equal to the critical water saturation.

   In the oil zone, *SWINIT must be greater than the connate water saturation. If *SWINIT less than connate is found, Sw in the block will be set equal to connate water saturation, Pcow will be scaled but the block will not be in equilibrium.

   For this approach use the following associated keywords:

   *DWOC, *REFDEPTH, *REFPRES, *ZOIL or *ZDEPTH or *ZGLOBAL(C), and *SWOC. If required, *SWINIT may be used.

6. *VERTICAL *DEPTH_AVE *WATER_OIL_GAS . Gravity initialization for a saturated reservoir with a gas cap, with saturations averaged over the depth interval covered by a grid block. In this approach, for example, if a grid block has its block center slightly above the gas-oil contact depth, the gas saturation assigned to the block is the average over the block volume of the local gas saturation, and not simply the gas saturation value in the gas cap.

   For this option, the water-oil and gas-oil contact depths, together with a reference pressure and a reference depth must be specified. Note that these are the same data required when using option (3) above (*VERTICAL *BLOCK_CENTER *GASCAP).

The composition can be specified by either of the three options, i.e., via

(a) Composition vectors representing the averaged oil zone composition and the averaged gas zone composition (*ZOIL and *ZGAS). or

(b) A composition variation with depth (*ZDEPTH) or

(c) Global composition of each grid block (*ZGLOBAL or *ZGLOBALC)

When this initialization is performed, a table is constructed giving oil phase pressure, water-oil capillary pressure, oil- gas capillary pressure, and several other properties as functions of depth. This table is constructed by Runge-Kutta integration of the equation determining the hydrostatic pressure. Water and gas saturations are then determined as functions of depth by inverting the tables which give water-oil capillary pressure and oil-gas capillary pressure as functions of saturation. Block water and gas saturations are then determined by numerical quadrature of the local water saturation.

For *DEPTH_AVE initialization use of *SWINIT is allowed.

In the Gas Cap, *SWINIT can vary between zero and the critical (not connate) water saturation. If *SWINIT greater than critical water saturation is found, Sw in that block will be set equal to the critical water saturation.

In the oil zone, *SWINIT must be greater than the connate water saturation. If *SWINIT less than connate is found, Sw in the block will be set equal to connate water saturation, Pcow will be scaled but the block will not be in equilibrium.

For this approach use the following associated keywords:

*DWOC, *DGOC, *REFDEPTH, *REFPRES, *ZOIL & *ZGAS, or *ZDEPTH or *ZGOBAL(C), and *SWOC. If required, *SWINIT may be used.

7. *VERTICAL *DEPTH_AVE *WATER_GAS . Gravity initialization for a reservoir in which all of the hydrocarbon fluid is initially in the gas phase, with saturations averaged over the depth interval covered by a grid block. In this approach, for example, if a grid block has its block center slightly below the water-gas contact depth, the water saturation assigned to the block is the average over the block volume of the local water saturation, and not simply the water saturation value in the water zone.

   For this option, the water-gas contact depth, together with a reference pressure and a reference depth must be specified.

   The composition can be specified by either of the three options, i.e., via

   (a) A composition vector representing the averaged gas zone composition (*ZGAS). or

   (b) A composition variation with depth (*ZDEPTH) or

   (c) Global composition of each grid block (*ZGLOBAL or *ZGLOBALC)

   For *WATER_GAS, alone among the depth-averaged options, the default is for a sharp transition between water and gas, before the averaging over grid block thicknesses. The *TRANZONE subkeyword allows the user to specify that a water-gas transition zone should exist as determined by the WATER-OIL capillary pressure curve entered by the user.

For *DEPTH_AVE initialization use of *SWINIT is allowed.

In the gas-zone, *SWINIT can vary between zero and the critical (not connate) water saturation. If *SWINIT greater than critical water saturation is found, Sw in that block will be set equal to the critical water saturation.

In the oil zone, *SWINIT must be greater than the connate water saturation. If *SWINIT less than connate is found, Sw in the block will be set equal to connate water saturation, Pcow will be scaled but the block will not be in equilibrium.

When this initialization is performed, a table is constructed giving water or gas phase pressure and several other properties as functions of depth. This table is constructed by Runge-Kutta integration of the differential equation determining the hydrostatic pressure. Water and gas saturations are then determined as functions of depth. Block water and gas saturations are then determined by numerical quadrature of the local saturations.

For this approach use the following associated keywords:

*DWGC, *REFDEPTH, *REFPRES, *ZGAS or *ZDEPTH or *ZGLOBAL(C), and *SWOC. If required, *SWINIT may be used.

GEM also allows the specification of multiple initialization regions, each with distinct reference pressure, reference depth, depths of contacts (gas- oil, water-oil, and water-gas as appropriate) and fluid compositions. For more information, see the entries in this chapter for the keywords *REFDEPTH, *REFPRES, *DWGC, *DWOC, *DGOC, *GOC_PC, *WOC_PC, *WGC_PC, *SWOC, *ZGAS, *ZOIL, *CDEPTH, *ZDEPTH, and *ITYPE.

Multiple initialization regions can be used with all of GEM's initialization methods EXCEPT for *USER_INPUT. The user must apply the same initialization method to all of the regions; it is not currently possible to specify different initialization methods for different initialization regions. The regions are specified by listing the grid blocks belonging to each region.

8. Oil Wet Initialization (DEPTH_AVE and BLOCK_CENTER COMP initializations):

   If the reservoir is identified as oil wet (keyword RPT OILWET) then the user can initialize the model, that is specify the initial pressure and fluid global compositions / saturations in the reservoir in one of three ways: Using DEPTH_AVE, all sub options are available, using BLOCK_CENTER COMP (only sub option supported), or using USER_INPUT.

   DEPTH_AVE and BLOCK_CENTER COMP initializations (when used with the oil wet option) assume that the reservoir is initially water wet and has had its wettability altered (to oil wet). This type of initialization therefore produces results which are very similar to a water wet reservoir initialization which at a later date undergoes a wettability change to oil wet, and is then allowed to re-equilibrate.

   The capillary pressure between non wetting and wetting phases still govern the transition zones, and similar to the water wet case the gas zone contains gas and water, the oil zone contains oil and water (both mobile and immobile) and the water zone contains water.

The position and the extent of the transition zone (relative to the water-oil contact DWOC) is governed by the entered capillary pressure curve (via keywords *SWT and SGT) and the by the value of the capillary pressure specified at the WOC. With the DEPTH_AVE initialization the user has the choice of entering a value of the capillary pressure at the water oil contact with the keyword WOC_PC. If no value is entered by the user then the default for oil wet is to assume that the DWOC entered corresponds to the free water level and therefore the capillary pressure corresponding to the minimum value in the SWT table will be assigned. For the oil wet option please note that the capillary pressure values in the SWT table as well as the value entered with WOC_PC keyword are interpreted to be the non wetting phase (water) pressure minus the wetting phase pressure (oil) or

$P_{cwo} = P_w - P_o$

In the case of BLOCK_CENTER COMP initialization, the user is not provided with the option of entering a capillary pressure corresponding to the oil water contact depth. GEM internally assigns a capillary pressure of zero at the WOC. The user should then enter a DWOC that corresponds to zero capillary pressure.

Recall that for the water wet case typically the DWOC corresponds to the minimum depth at which a water saturation of 100% is reached. For the water wet case transition zone extends upwards from the DWOC. Below the WOC is the water zone. For the oil wet case the maximum wetting phase (oil) saturation occurs at the WOC corresponding to the minimum value of Pcwo in the SWT table. The transition region extends downward towards the 100% water saturation depth.

A different WOC_PC can be entered for each initialization region for DEPTH_AVE initialization.

Within an initialization region, all rock types should either be oil wet or water wet. If both oil wet and water wet rock types are found, unphysical results may occur. The DEPTH_AVE initialization option will warn the user if this occurs and stop the simulation. For a more detailed discussion of the oil wet option please turn to the tutorial section of this manual.

9. *USER_INPUT, formerly *VERTICAL *OFF, which is still accepted by GEM. Pressure, water saturation and global composition are specified for each grid block. Oil and gas saturations are then determined from flash calculations. Use the following keywords for this option:

   *PRES, *SW and *ZGLOBALC (or *ZGLOBAL).

10. A combination of two different methods, *USER_INPUT and one of the gravity capillary initialization methods such as *DEPTH_AVE. Please note that this method is currently available for GEM coal bed methane option only. In this scenario typically one section of a reservoir is assigned initial pressures, saturations etc based on user input values and for another part of the reservoir, the pressures, saturations are calculated based on the specified gravity-capillary initialization method such as DEPTH_AVE. Typically matrix blocks will be initialized using user specified values of pressure, saturations, compositions and fracture blocks using a gravity capillary equilibrium method. All the necessary keywords for each method are required, as if each method was being specified

singly. The keywords necessary for initialization with USER_INPUT method must be specified as a block and the keywords necessary for specifying a gravity-capillary type method must be specified as a block.

Note that when the second method in such initializations is *DEPTH_AVE, composition vector for the same must not be specified by *ZGLOABAL(C). Doing so may lead to unexpected results without any warning or error message.

For a gravity capillary initialization method such as DEPTH_AVE in the specific context of two different initialization methods, region 1 should not be assigned to a grid block via ITYPE. This is because region 1 is “associated” with user input grid blocks. This allows or enables certain parameter values to be entered for user input grid blocks. For example the first value entered with the keyword *SWOC will be assigned to grid blocks using the USER_INPUT method even though the keyword SWOC needs to be specified as part of the gravity capillary initialization method such as DEPTH_AVE . If a particular optional keyword (for example TCMULT) is allowable under both user input and the gravity capillary method, then enter that keyword ONLY under the gravity-capillary method. The value entered for region 1 will then be assigned to user input grid blocks. The example below illustrate the data entry requirements for multiple initialization regions:

Example: The complete set of keywords used to initialize the reservoir for coal bed methane model is shown below. In this case the fracture system is initialized using *VERTICAL *COMP and the matrix system using *USER_INPUT. Please note the following:

i) All the keywords required for gravity-capillary initialization are specified as a block following the keyword VERTICAL COMP etc and all keywords required for user input follow the USER_INPUT keyword as a block that is contiguous lines of keywords. If a keyword is allowed for both USER_INPUT and gravity capillary initialization then enter that keyword with the gravity capillary initialization block alone (for example keyword TCMULT)

ii) Since region 1 is reserved, a minimum of two regions are required for VERTICAL COMP or any other gravity capillary initialization method. Region 2 is assigned to fracture blocks with keyword *ITYPE *FRACTURE 2. Fracture blocks will be assigned a value of 0.9999 as a maximum water saturation in a grid block (keyword SWOC) and a critical temperature multiplier value of 1.0 (keyword TCMULT)

```
*INITIAL
*USER_INPUT
*PRES MATRIX CON 5005.0
*SW MATRIX CON      0.1
*ZGLOBALC 'C1'  MATRIX CON         1.0
*ZGLOBALC 'CO2' MATRIX CON         0.0

*VERTICAL *COMP
*NREGIONS 2
*REFPRES  5000.0 7650.0    ** first value (region 1) not used
*REFDEPTH 1000.0  1000.0   ** first value (region 1) not used
```

```
*ZDEPTH 1 1000.0 1.0 0.0   ** region 1 values not used
        2 1000.0 1.0 0.0
*DWOC 100 100              ** first value not used
*SWOC 0.592 0.9999         ** first value is assigned to user input
blocks
*CDEPTH 9999. 9999.        ** first value not used
*TCMULT 1.2 1.0            ** first value is assigned to user
                              input blocks

*ITYPE *FRACTURE *CON 2    ** reserve region 1 for user input
```

Flexible Separators for Initialization

GEM allows surface separators to be specified in a wide variety of ways. Please see the WELL AND RECURRENT DATA section of this manual for a full explanation. This variety applies not only to well separators but also to the initialization separator which is used to estimate the volumes of surface streams initially present in the reservoir. The keywords *SEPARATOR, *DENSTR-KEY, *DENSTR-MIX, *INJECTOR-P, *INJECTOR-T, *INJECTOR-EOS, and *PLANT-TABLE have been modified or added to the INITIAL CONDITIONS section to allow use and control of the flexible separator format for the initialization separator.

The *SEPARATOR keyword has been kept backward compatible; old GEM data sets run as they did with older versions without alteration.

Initial Conditions Identifier (Required) *INITIAL

PURPOSE:

*INITIAL indicates the beginning of initial condition values.

FORMAT:

*INITIAL

DEFAULTS:

Required keyword. No default.

CONDITIONS:

This keyword must be the first keyword in the INITIAL CONDITIONS keyword group, which must come immediately after the ROCK-FLUID DATA keyword group.

Vertical Equilibrium or User Input Selection (Required)

***VERTICAL, *USER_INPUT**

PURPOSE:

*VERTICAL and *USER_INPUT control the type of reservoir initialization calculation.

FORMAT:

```
*VERTICAL        ( *BLOCK_CENTER  phase | *DEPTH_AVE  phase  option )
    phase =      ( *WATER_OIL_GAS |
                   *WATER_OIL |
                   *WATER_GAS  (*NOTRANZONE | *TRANZONE) )
    option =     ( *EQUIL | *NOEQUIL )
                 ( *FINE_INTG )
*USER_INPUT
```

DEFINITIONS:

*VERTICAL

Indicates that pressures are determined from the hydrostatic equation and saturations from the capillary pressure tables.

*BLOCK_CENTER

Indicates that block saturations are to be assigned as the values prevailing at the block center.

*DEPTH_AVE

Indicates that block saturations are to be assigned as averages of the corresponding saturations over the depth interval spanned by the grid block.

*WATER_OIL

Perform gravity-capillary equilibrium initialization of a reservoir initially containing no gas.

*WATER_GAS

Perform gravity-capillary equilibrium initialization for reservoirs with only water and gas phases initially present.

*WATER_OIL_GAS

Perform gravity-capillary equilibrium initialization of a reservoir initially containing water, oil, and gas phases.

*COMP

Perform gravity-capillary equilibrium initialization with user specified composition as a function of grid block depth. This is the only BLOCK_CENTER option available for initializing oil wet reservoirs.

*EQUIL

With *DEPTH_AVE options, add a pressure correction to each phase (during the simulation) in order that the reservoir initially be in gravitational equilibrium. The averaging of saturations over depth prevents gravitational

equilibrium from being established simply by setting saturations from capillary pressure tables. The calculated pressure corrections can be printed by including a *OUTPRN *RES *ALL line in the I/O CONTROL data section.

*NOEQUIL

Do not add a phase pressure correction when a *DEPTH_AVE option is used. *EQUIL is the default.

*TRANZONE

A subkeyword for the *WATER_GAS option only, which specifies that the water-oil capillary pressure curves entered by the user should be used to determine a water-gas transition zone in which the gas saturation is above its critical value and the water saturation is between 1-Sorw and Swcrit. The presentation of this subkeyword in the format line above was constrained by page width and may be misleading -- *TRANZONE should be used with the *WATER_GAS option only. *NOTRANZONE is the default.

*NOTRANZONE

A subkeyword for the *WATER_GAS option only, which specifies that the transition from the water to the gas zone should be sharp, before the averaging over grid block thicknesses. *NOTRANZONE is the default -- if neither *TRANZONE nor *NOTRANZONE is encountered after *WATER_GAS, *NOTRANZONE is assumed.

*FINE_INTG

Subkeyword to do fine integration for *DEPTH_AVE initialization when a corner point grid is used. During the evaluation of block phase saturations, fine integration divides a corner point block into a number of horizontal intervals along its depth. These intervals include all block corners, and may also include important depth levels, such as oil-water-contact (assuming the block intersects one of these important depth levels).

Within each depth interval, error controlled Romberg integration is applied, which integrates the product of phase saturation and the area of a horizontal interval. The average block saturation is given by the summation of these integrations over the whole block volume.

This option provides better in-place calculation precision when a corner point block is highly skewed (e.g. top and bottom surfaces highly tilted from horizontal). Applying the option to grid systems other than corner point grids would have no effect and so the option is internally switched off for grid systems other than corner point grids.

*USER_INPUT

Specify pressure, composition and saturation of each grid block. Use this subkeyword only when the pressure and composition at each grid block need to be defined individually. This keyword replaces, and is exactly equivalent to, the *VERTICAL *OFF option of earlier versions, which is still accepted by GEM.

DEFAULTS:

One of *VERTICAL or *USER_INPUT is required. No default.

If *VERTICAL is entered without any subkeyword, *VERTICAL *BLOCK_CENTER *WATER_OIL is assumed, which is also the case if *VERTICAL *BLOCK_CENTER is entered. If *VERTICAL *DEPTH_AVE is entered, then *VERTICAL *DEPTH_AVE *WATER_OIL_GAS *EQUIL is assumed. *NOTRANZONE is assumed for *WATER_GAS.

If *FINE_INTG is not found, the default is no fine integration.

CONDITIONS:

This keyword must be the second keyword in the INITIAL CONDITIONS keyword group.

EXPLANATION:

These keywords indicate how the initial reservoir conditions are to be determined.

Under *VERTICAL *BLOCK_CENTER *WATER_OIL (formerly *VERTICAL *ON, which is still accepted by GEM), the water-oil contact depth (*DWOC) together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. One composition vector (entered under *ZGLOBALC or *ZGLOBAL), representing the averaged reservoir fluid composition, is required. This composition is assigned to all grid blocks.

Under *VERTICAL *BLOCK_CENTER *WATER_GAS (formerly *VERTICAL *GASCAP, which is still accepted by GEM, with the water-oil and gas-oil contact depths set equal), the water-gas contact depth (*DWGC) together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. A composition vector (entered under *ZGAS) representing the average hydrocarbon composition in the gas zone is required. If the reference depth does not lie in the gas zone, *ZOIL may be required.

Under *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS (formerly *VERTICAL *GASCAP, which is still accepted by GEM), the water-oil contact (*DWOC) and gas-oil contact (*DGOC) depths, together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. Two sets of compositions, one representing the averaged oil zone composition (*ZOIL) and the other representing the averaged gas zone composition (*ZGAS), are required. (Note that the simulator will attempt to determine a missing *ZGAS, or *ZOIL, internally from a saturation pressure calculation if the reference depth is in the oil, or gas, zone, respectively.)

To use this option to initialize a gas reservoir, set *DGOC = *DWOC and *ZOIL = *ZGAS, or use *VERTICAL *BLOCK_CENTER *WATER_GAS as outlined above.

To use this option to initialize an under-saturated oil reservoir, set *DGOC to be above the top of the reservoir. When this is done the results may be slightly different from those obtained using *VERTICAL *BLOCK_CENTER *WATER_OIL because the latter option retains gas anywhere in the reservoir that the equation of state finds a gas phase, whereas the current option does not allow gas below the maximum depth at which the oil-gas capillary pressure allows gas to exist.

With *COMP, the water-oil contact (*DWOC) location, together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. A table representing the variation of global composition with depth must be entered using *ZDEPTH. The user must enter a critical depth, which determines the identity, oil or gas, of a single phase hydrocarbon

fluid, using the keyword *CDEPTH. Since the capillary pressure value of zero at the WOC is assigned internally by GEM for this option, it is incumbent on the user to enter a value of water-oil contact depth (DWOC) that corresponds to zero capillary pressure. See also the tutorial section for a more comprehensive discussion of oil wet initialization.

Under *VERTICAL *DEPTH_AVE *WATER_OIL, the water-oil contact depth (*DWOC) together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. There are three options for specifying the composition vector. Either one of such options should be used.

(a) One composition vector (entered under *ZOIL), representing the averaged reservoir fluid composition. This composition is assigned to all grid blocks, or

(b) A table representing the variation of global composition with depth entered using *ZDEPTH, or

(c) Global composition for each grid block entered via *ZGLOBAL(C).

This type of initialization is used for undersaturated oil reservoirs in which the added accuracy in initial amounts in place made possible by the depth averaging is wanted.

Under *VERTICAL *DEPTH_AVE *WATER_GAS, the water-gas contact depth (*DWGC) together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. There are three options for specifying the composition vector. Either one of such options should be used.

(a) One composition vector (entered under *ZGAS), representing the averaged reservoir composition in the gas zone. This composition is assigned to all grid blocks, or

(b) A table representing the variation of global composition with depth entered using *ZDEPTH, or

(c) Global composition for each grid block entered via *ZGLOBAL(C).

This type of initialization is used for water-gas reservoirs in which the added accuracy in initial amounts in place made possible by depth averaging of saturations is wanted. If *TRANZONE is specified, a water-gas transition zone is determined using the entered water-oil capillary pressure curves.

Under *VERTICAL *DEPTH_AVE *WATER_OIL_GAS, the water-oil contact (*DWOC) and gas-oil contact (*DGOC) depths, together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. Two sets of compositions, one representing the averaged oil zone composition (*ZOIL) and the other representing the averaged gas zone composition (*ZGAS), are required. There are three options for specifying the composition vector. Either one of such options should be used.

(a) Two sets of compositions, one representing the averaged oil zone composition (*ZOIL) and the other representing the averaged gas zone composition (*ZGAS), or

(b) A table representing the variation of global composition with depth entered using *ZDEPTH, or

(c) Global composition for each grid block entered via *ZGLOBAL(C).

This option is used for water-oil-gas reservoirs in which the added accuracy in initial amounts in place made possible by depth averaging of saturations is wanted.

If the capillary pressure at the WOC is not entered by the user using the keyword WOC_PC then the default assigned by GEM is the value of zero for the water wet case and the minimum capillary pressure entered with the corresponding to SWT table for the oil wet case. See also the tutorial section for a more comprehensive discussion of oil wet initialization.

See the notes in the beginning of this section for more information.

Under *USER_INPUT (formerly *VERTICAL *OFF, which is still accepted by GEM), pressure, water saturation and global composition at each grid block must be specified by the user under the *PRES, *SW, and *ZGLOBALC (or *ZGLOBAL) keywords. Oil and gas saturations are then determined from flash calculations.

Example:

For a water-gas reservoir with a transition-zone and gravity equilibrium, enter

```
*VERTICAL *DEPTH_AVE *WATER_GAS *TRANZONE *EQUIL
```

For a water-oil-gas reservoir with a transition zone, gravity equilibrium and fine integration, enter:

```
*VERTICAL *DEPTH_AVE *FINE_INTG
```

Number of Initialization Regions (Optional) *NREGIONS

PURPOSE:

*NREGIONS introduces the integer nreg which specifies the number of initialization regions into which the reservoir has been divided.

FORMAT:

*NREGIONS nreg

DEFINITIONS:

nreg

Integer specifying the number of initialization regions into which the reservoir has been divided. Different depths of contact, reference pressures, etc. may be specified in each region.

DEFAULTS:

Optional keyword. If *NREGIONS is not encountered in the initialization data, then nreg is defaulted to 1.

CONDITIONS:

This keyword must be in the INITIAL CONDITIONS keyword group, and must precede all keywords which refer to nreg, namely the following:

```
*DWOC, *DGOC, *DWGC, *GOC_PC, *WOC_PC, *WGC_PC *ITYPE,*ZOIL,
*ZGAS, *SWOC, *ZDEPTH, *CDEPTH, *REFDEPTH,*REFPRES
```

Specification of Initialization Regions (Optional) *ITYPE

PURPOSE:

*ITYPE introduces an array integers taking the values 1, ..., nreg which specify to which initialization region a given grid block or set of grid blocks belong. Any of the array-reading options may be used.

ARRAY:

*ITYPE init_reg_array

DEFINITIONS:

init_reg_array

An array of integers specifying the initialization region for each grid block. The array consists of integers in the range 1, ..., nreg; the integer nreg was entered under the *NREGIONS keyword. The format for entering the array is explained in the Keyword Data Entry System section of this manual, under Input of Grid Property Arrays.

DEFAULTS:

Optional keyword. If *ITYPE is not encountered it is assumed that all grid blocks are in initialization region 1.

CONDITIONS:

If it appears, this keyword must be in the INITIAL CONDITIONS keyword group, and must follow the *NREGIONS keyword.

Example:

For a 5 X 5 X 4 Cartesian reservoir with the first two layers in the first initialization region and the last two layers in the second:

```
*NREGIONS 2
*ITYPE *IJK
   1:5  1:5  1:2  1
   1:5  1:5  3:4  2
```

Oil Zone and Gas Cap Compositions (Conditional)

***ZOIL, *ZGAS**

PURPOSE:

*ZOIL introduces input of oil zone compositions (one composition vector for each initialization region in the reservoir).

*ZGAS introduces input of gas cap compositions.

FORMAT:

```
*ZOIL    yok(1,1:nc)
         yok(2,1:nc)
         :
         yok(nreg,1:nc)
*ZGAS    ygk(1,1:nc)
         ygk(2,1:nc)
         :
         ygk(nreg,1:nc)
```

DEFINITIONS:

yok(1,1:nc)

nc real numbers representing the mole fractions of the nc components in the oil zone fluid in the first initialization region. The entered numbers will be normalized internally to sum to unity. nc is the number of hydrocarbon components characterizing the fluid; see the Fluid Component Data section.

ygk(1,1:nc)

nc real numbers representing the mole fractions of the nc components in the gas cap fluid in the first initialization region. The entered numbers will be normalized internally to sum to unity.

DEFAULTS:

Conditional keyword.

In a given initialization region the default for yok is the equilibrium oil composition calculated from flash calculations at the gas-oil contact pressure using ygk for the same region.

In a given initialization region the default for ygk is the equilibrium gas composition obtained by flashing yok for the same region at the gas-oil contact pressure.

At least one of yok or ygk must be entered for each initialization region if either *ZOIL or *ZGAS is required for the initialization method specified. If a required phase composition is not entered and the entered composition is single-phase at the gas-oil contact pressure, then GEM is unable to generate the default values of yok or ygk and an error message is printed. In this case, review the data for consistency or enter both yok and ygk directly.

CONDITIONS:

If they appear, these keywords must be in the INITIAL CONDITIONS keyword group. If they appear, and if *NREGIONS also appears, then these keywords must follow *NREGIONS.

They are required with *VERTICAL *GASCAP, *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS or *WATER_GAS, and may be also used for *VERTICAL *DEPTH_AVE *WATER_OIL_GAS, *WATER_OIL, *WATER_GAS as one of the composition specification methods.

If *ZOIL appears in the data set, then one yok vector must be entered for EACH of the nreg initialization regions in the reservoir. It is not possible to enter yok for some regions but not for others. If the number of yok vectors read is greater than zero but less than nreg, an error message is printed and the simulation will not proceed.

If *ZGAS appears in the data set, then one ygk vector must be entered for EACH of the nreg initialization regions in the reservoir. It is not possible to enter ygk for some regions but not for others. If the number of ygk vectors read is greater than zero but less than nreg, an error message is printed and the simulation will not proceed.

EXPLANATION:

With the keywords *VERTICAL *GASCAP, *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS or *WATER_GAS, and *VERTICAL *DEPTH_AVE *WATER_OIL_GAS, grid block global compositions are calculated from yok and ygk. For grid blocks in the oil zone, yok is flashed to determine an equilibrium oil and gas phase. Global composition is then calculated by mixing the oil and gas phase by the predetermined saturations obtained from gravity-capillary-equilibrium considerations. For example, if a block is far from the gas-oil transition zone and no gas phase exists in the block, any gas phase obtained by flashing yok at the grid block pressure is discarded and the oil phase composition from the flash is taken as the grid block composition. Since grid block pressure normally varies with depth, the global composition may also vary with depth.

If the entered yok yields a single (oil) phase when flashed at all grid block pressures, the oil phase composition is equal to yok in all grid blocks. In this case the grid block composition is not depth dependent.

To get different compositions for all grid blocks in the oil zone, enter the saturated oil composition at the bottom of the reservoir as yok. This ensures a two-phase condition for all grid blocks as pressure decreases with decreasing depth.

If *ZGAS is missing, ygk is assumed to be the gas phase composition obtained by flashing the yok at the gas-oil contact pressure. If flashing yok at the gas-oil contact pressure fails to yield two phases, yok cannot be determined and the program will be stopped. Thus it is advisable to enter both yok and ygs. They can be determined from appropriate flash calculations using WINPROP.

Similar procedures are used for grid blocks in the gas zone and when *ZOIL is missing. To get different compositions for all grid blocks in the gas zone, enter the saturated gas composition at the gas-oil contact as ygk.

In order to avoid large saturation changes (and large grid block pressure changes) during the first timestep of the simulation, it is recommended that yok and ygk be in equilibrium with each other at the reservoir temperature and the pressure which prevails at the gas-oil contact. One way to ensure this is to use the GOC depth as the reference depth and the pressure at the GOC as the reference pressure. Then a flash or saturation calculation done with a phase behavior package such as CMG's WINPROP at reservoir temperature and the reference pressure will yield gas and oil compositions which are in equilibrium.

Examples:

For nc = 5 and one initialization region:

```
*ZOIL  .40  .40  .10  .07  .03
*ZGAS  .80  .13  .06  .01  .00
```

For nc = 4 and 3 initialization region:

```
*ZOIL  .40  .40  .10   .10
       .40  .50  .05   .05

       .40  .40  .10   .10

*ZGAS  .80  .13  .07  0.
       .85  .10  .05  0.
       .85  .10  .05  0.
```

Initial Reservoir Fluid Properties (Conditional) *PRES, *SW, *ZGLOBALC, *ZGLOBAL

PURPOSE:

All of these keywords are required for the USER_INPUT initialization method.

*PRES indicates the input of reservoir pressures (kPa | psi).

*SW indicates input of initial water saturation (fraction).

*ZGLOBALC indicates input of initial grid block global composition on a component-by-component basis.

*ZGLOBAL indicates input of initial grid block global composition. The entered numbers will be normalized to sum to unity internally. *ZGLOBALC is preferred.

*ZGLOBAL or *ZGLOBALC may also be used with *BLOCK_CENTER as well as *DEPTH_AVE initializations.

ARRAY:

*PRES
*SW
*ZGLOBALC
*ZGLOBAL

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

These keywords must be in the INITIAL CONDITIONS keyword group. *PRES, *SW and one of the ZGLOBAL specifications are REQUIRED with *USER_INPUT and *VERTICAL *OFF. *PRES and *SW CANNOT be used with any of the other initialization options.

EXPLANATION:

The *PRES keyword defines the initial reservoir block pressures (of the oil phase). Any of the array reading options can be used.

The *SW keyword defines the initial water saturation at each block. Any of the array reading options can be used.

The *ZGLOBAL keyword defines the initial global composition at each block. Any of the array reading options can be used. Nc values corresponding to the nc components must be entered for each grid block. Nc is defined in the Fluid Component Data section. Global composition is required for grid blocks even if they are filled with water (Sw=1) if the solubility option (hydrocarbons soluble in water) is used. In this case, enter the oil phase composition immediately above the WOC as ZGLOBAL for these water blocks. The following is now preferred.

The *ZGLOBALC keyword can also be used to define the initial global composition at each block. Any of the array reading options can be used with *ZGLOBALC. However, an additional qualifier consisting of a component name, enclosed in single quotes, is required, and the required array values then consist only of the global molar fractions for the single

named component. This keyword must be used Nc times to define the entire set of values. Global composition is still required for grid blocks even if they are filled with water (Sw=1) if the solubility option (hydrocarbons soluble in water) is used.

For example, consider a 3-component fluid (with components name 'C1', 'C2-5' and 'C6PLUS')in a 200 x 100 x 3 reservoir:

```
*PRES     *KVAR  16100.0 16000.0 15900.0
*SW       *CON   0.18
*ZGLOBAL  *IJK   1:200  1:100  3  0.45  0.32  0.23
                 1:200  1:100  2  0.40  0.30  0.30
                 1:200  1:100  1  0.36  0.29  0.35
```

or the following could be used:

```
*PRES     *KVAR  16100.0 16000.0 15900.0
*SW       *CON   0.18
*ZGLOBALC  'C1'      *IJK   1:200  1:100  3  0.45
                            1:200  1:100  2  0.40
                            1:200  1:100  1  0.36
*ZGLOBALC  'C2-5'    *IJK   1:200  1:100  3  0.32
                            1:200  1:100  2  0.30
                            1:200  1:100  1  0.29
*ZGLOBALC  'C6PLUS'  *IJK   1:200  1:100  3  0.23
                            1:200  1:100  2  0.30
                            1:200  1:100  1  0.35
```

See the notes at the beginning of this section for more information.

Initial Reservoir Temperature (Optional) *TEMPER

PURPOSE:

This keyword is used to specify the initial reservoir temperature at any block.

ARRAY:

*TEMPER

DEFAULTS:

Optional keyword. The *TRES keyword is always required to be entered in the Component Properties section of the data set. The default block temperature will be that specified with TRES.

CONDITIONS:

This keyword must be in the INITIAL CONDITIONS keyword group. It can be used with any of the initialization options.

EXPLANATION:

The *TEMPER keyword defines the initial reservoir temperature at any block. Any of the array reading options can be used.

The *TRES keyword must always be specified in the Fluid model section. This is used to initialize some temperature dependent viscosity properties, as described in the TRES section of the manual.

The *TEMPER keyword can be used to define the initial reservoir temperature for all of the reservoir blocks, or only some portion of the blocks. Any block that does not get defined using *TEMPER will take the temperature defined using *TRES.

For example, temperatures in a 4-layer reservoir could be specified as:

```
*TEMPER   *KVAR  204.0  208.5  213.0  220.5
```

Reference Depth and Reference Pressure (Conditional)

***REFDEPTH**

PURPOSE:

*REFDEPTH indicates input of reference depth.

*REFPRES indicates input of reference pressure.

FORMAT:

*REFDEPTH	depth(i)	i = 1, nreg
*REFPRES	Pressure(i)	i = 1, nreg

DEFINITIONS:

depth(i)

Depth of reference pressure (*REFPRES) (m | ft) for initialization region i . If the value is negative, then *REFDEPTH is interpreted as an altitude above some datum. For accurate initialization, the reference depth should be within the reservoir, but GEM can handle cases in which the reference depth is above or below the reservoir.

pressure(i)

The reference pressure (kPa | psi) for initialization region i at the reference depth (*REFDEPTH). It is used to set reservoir block pressures.

nreg

The number of initialization regions into which the reservoir has been divided.

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

If they appear, these keywords must be in the INITIAL CONDITIONS keyword group. If they appear, and if *NREGIONS also appears, then these keywords must follow the *NREGIONS keyword.

They are required with *VERTICAL *ON, *VERTICAL *GASCAP, and *VERTICAL *COMP (which are equivalent to *VERTICAL *BLOCK_CENTER *WATER_OIL, *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS, and *VERTICAL *BLOCK_CENTER *COMP) and with *VERTICAL *BLOCK_CENTER *WATER_GAS, *VERTICAL *DEPTH_AVE *WATER_OIL, *VERTICAL *DEPTH_AVE *WATER_GAS and *VERTICAL *DEPTH_AVE *WATER_OIL_GAS.

EXPLANATION:

The reference pressure and the reference depth are used in the vertical equilibrium options, along with the capillary pressures, grid block depths, water-oil contact and the gas-oil contact depths, to calculate the block pressures and saturations.

See the notes in the beginning of this section for more information.

Depth to Water-Oil, Gas-Oil, and Water-Gas Contacts (Conditional) *DWOC, *DGOC, *DWGC

PURPOSE:

*DWOC indicates input of water-oil contact depth.

*DGOC indicates input of gas-oil contact depth.

*DWGC indicates input of water-gas contact depth.

FORMAT:

*DWOC	depth(i),	i = 1, nreg
*DGOC	depth(i),	i = 1, nreg
*DWGC	depth(i),	i = 1, nreg

DEFINITIONS:

depth(i)

Depth (m | ft) to the specified contact plane in initialization region i. If the value is negative, then depth is interpreted as an altitude above some datum.

nreg

Number of initialization regions into which the reservoir has been divided.

DEFAULTS:

Conditional keywords. No default value.

CONDITIONS:

If they appear, these keywords must be in the INITIAL CONDITIONS keyword group. If they appear, and if *NREGIONS also appears, then these keywords must follow *NREGIONS.

*DWOC is required with *VERTICAL *ON, *VERTICAL *GASCAP, *VERTICAL *COMP (which are equivalent to *VERTICAL *BLOCK_CENTER *WATER_OIL, *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS, and *VERTICAL *BLOCK_CENTER *COMP) and with *VERTICAL *DEPTH_AVE *WATER_OIL and *VERTICAL *DEPTH_AVE *WATER_OIL_GAS.

*DGOC is required with *VERTICAL *GASCAP (which is equivalent to *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS) and with *VERTICAL *DEPTH_AVE *WATER_OIL_GAS.

*DWGC is required with *VERTICAL *BLOCK_CENTER *WATER_GAS and with *VERTICAL *DEPTH_AVE *WATER_GAS

EXPLANATION:

The *DWOC keyword defines the depth to the water-oil plane. If there is no water-oil contact set depth to a value greater than the depth of the bottom of the reservoir.

The *DGOC keyword defines the depth to the gas-oil contact plane. If there is no gas-oil contact set depth to a value less than the depth of the top of the reservoir.

The *DWGC keyword defines the depth to the water-gas contact in a reservoir in which no oil is present. If there is no water-gas contact set depth to a value greater than the depth of the bottom of the reservoir.

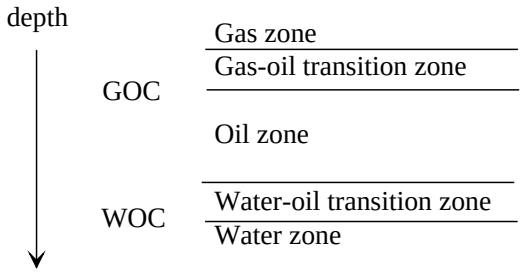
The definition for the term 'contact depth' as used in the *BLOCK_CENTER options differs slightly from the definition used in the *DEPTH_AVE options. This is unfortunate, but was necessary in order that the capillary pressure vs. depth tables could be constructed for the *DEPTH_AVE options. For the *BLOCK_CENTER options, the water-oil contact depth is defined as the smallest depth at which the water saturation assumes its maximum value (unity or nearly unity). For the *DEPTH_AVE options, the water-oil contact depth is defined as the depth at which the water-oil capillary pressure assumes a prescribed value. The default value for this capillary pressure is zero, but the user can set a non-zero value for the water-oil capillary pressure at the water-oil contact in the *DEPTH_AVE options by using the *WOC_PC keyword.

The capillary pressure at the WOC is generally zero. For the BLOCK_CENTER COMP case it cannot be overridden. For the DEPTH_AVE initialization it can be overridden by the user entered value. Depending on the entered capillary pressure table and the value of capillary pressure selected for the WOC, the transition region can be above the WOC or below the WOC or can straddle both sides of the WOC.

For the *BLOCK_CENTER options, the gas-oil contact depth is defined as the smallest depth at which the gas saturation has the value zero. For the *DEPTH_AVE options, the gas-oil contact depth is defined as the depth at which the oil-gas capillary pressure assumes a prescribed value. The default value for this capillary pressure is zero, but the user can set a non-zero value for the oil-gas capillary pressure at the gas-oil contact in the *DEPTH_AVE options by using the *GOC_PC keyword.

For the *BLOCK_CENTER options, the water-gas contact depth is defined as the smallest depth at which the gas saturation has the value zero. For the *DEPTH_AVE options, the water-gas contact depth is defined as the depth at which the gas-water capillary pressure assumes a prescribed value. The default value for this capillary pressure is zero, but the user can set a non-zero value for the gas-water capillary pressure at the water-gas contact in the *DEPTH_AVE options by using the *WGC_PC keyword. Note that the gas-water capillary pressure is entered in the *SWT table in the Pcow and/or Pcowi columns, in the Rock-Fluid data section.

Under the *BLOCK_CENTER initialization options, the gas-oil transition zone is ABOVE the gas-oil contact, and the water-oil transition zone is above the water-oil contact. In the *DEPTH_AVE options, some gas may lie below the gas-oil contact and some oil may lie below the water-oil contact, depending on the capillary pressure values chosen at the contacts. Similar analogy holds for water-gas systems (see the explanations with the *WOC_PC, *GOC_PC and *WGC_PC keywords.).



Oil wet reservoirs

For the water wet case typically the DWOC corresponds to the minimum depth at which a water saturation of 100% is reached. For the water wet case transition zone extends upwards from the DWOC. Below the WOC is the water zone. For the oil wet case the maximum wetting phase (oil) saturation occurs at the WOC corresponding to the minimum value of Pcwo in the SWT table. The transition region extends downward towards the 100% water saturation depth.

Water Saturation Below Water-Oil Contact (Optional) *SWOC

PURPOSE:

*SWOC indicates input of water saturation below the water-oil contact

FORMAT:

*SWOC swbowc(i), i = 1, nreg

DEFINITIONS:

swbowc(i)
: Water saturation below oil-water contact (fraction) in initialization region i.

nreg
: The number of initialization regions into which the reservoir has been divided.

DEFAULTS:

Optional keyword. Default is 0.999999.

CONDITIONS:

If it appears, this keyword must be in the INITIAL CONDITIONS keyword group. If it appears, and if *NREGIONS also appears, then this keyword must follow *NREGIONS.

This keyword is required with *VERTICAL *ON and *VERTICAL *GASCAP (equivalent to *VERTICAL *BLOCK_CENTER *WATER_OIL and *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS) and *VERTICAL *BLOCK_CENTER *WATER_GAS and all of the *VERTICAL *DEPTH_AVE options, although the value may be defaulted for any of the above options, so that a *SWOC line need not appear in the data set.

EXPLANATION:

This keyword sets the water saturation below the water-oil contact during gravity-capillary initializations. The swbowc should not be too different from that calculated from the relative permeabilities of the water zone.

When the water solubility option is used, avoid entering 1.0 for SWBOWC, as this may lead to ill-conditioned problems with poor convergence.

Initial Reservoir Fluid Formation Volume Factors (Optional)

***BO, *BG, *RS**

PURPOSE:

*BO input initial oil formation volume factor.

*BG input initial gas formation volume factor.

*RS input initial solution gas-oil ratio.

FORMAT:

*BO	boinit
*BG	bginit
*RS	rsinit

DEFINITIONS:

boinit
: Initial oil formation volume factor, (res m3/std m3 | res bbl/stb)

bginit
: Initial gas formation volume factor, (res m3/std m3 | res bbl/stb)

rsinit
: Initial solution gas-oil ratio, (m3/m3 | SCF/stb)

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword must be in the INITIAL CONDITIONS keyword group. It is required when separator conditions to calculate the initial fluid-in-place have not been entered, i.e. when the keyword *SEPARATOR does not appear in the data set.

EXPLANATION:

Initial oil and gas in-place values are calculated using one of the following two approaches:

1. When bo, bg and rs are entered, the in-place values are calculated directly from the reservoir fluid volumes.
2. When a set of initial averaged separator conditions are entered via *SEPARATOR, in-place values are calculated by flashing all the reservoir fluids through this separator.

The latter approach is preferred as it produces results consistent with the well production calculations.

Separator for Initial Fluid-In-Place Calculations (Optional)

***SEPARATOR**

PURPOSE:

*SEPARATOR specifies the separator conditions through which the initial fluid-in-place is calculated.

FORMAT:

```
*SEPARATOR          pstage      tstage
                      :           :
-or-
*SEPARATOR stage_identification_lines
  (*WET-GAS stage_number)
  *STREAM-DEN stream_density_method_lines
```

where the stage_identification_lines (one set per stage) have the form:

```
(*EOS eos_set_no pstage tstage)   output_destination_lines
( *PLANT plant_table_no )
```

and the output_destination_lines have the form:

```
(*LIQUID-TO)      (stage_no)
(*MIDDLE-TO)      ('stream_name')
(*VAPOR-TO)
```

and the stream_density_method_lines have the form:

```
'stream_name'     (*EOS eos_set_no pstream tstream (root))
                  (*KEY den_table_no)
                  (*MIX den_set_no)
                  (*GASLAW zfac pstream tstream)
```

DEFINITIONS:

pstage

A set of real numbers specifying the separator pressure (kPa | psi). In the first (shorter) format for the *SEPARATOR keyword shown above, the stock tank pressure and temperature must be the pressure and temperature of the last stage.

tstage

A set of real numbers specifying the separator temperature (deg C | deg F). In the first (shorter) format for the *SEPARATOR keyword shown above, the stock tank pressure and temperature must be the pressure and temperature of the last stage.

*EOS

When appearing in a stage identification line, *EOS indicates that the stage is represented by a two- phase EOS flash at pressure and temperature pstage, tstage. When appearing in a stream density method line, *EOS indicates that the stream density is to be determined by a single-phase EOS density calculation at pressure and temperature pstream, tstream.

eos_set_no

The number of the equation-of-state parameter set to be used for the stage flash or for the stream density calculation. This is the set number entered for the EOS parameter set using the *EOSSET keyword in the COMPONENT PROPERTIES section of the data set (see the COMPONENT PROPERTIES manual section).

*PLANT

This subkeyword indicates that the fluid separation in the stage is to be represented by a table of component distribution fractions entered using the *PLANT-TABLE keyword.

plant_table_no

The number of the distribution fraction table entered using the *PLANT-TABLE keyword which is to be used to represent the fluid separation in the current stage.

*LIQUID-TO

Subkeyword introducing the destination of the liquid output of the current separator stage. This subkeyword is valid for both the *EOS and the *PLANT stage types.

*MIDDLE-TO

Subkeyword introducing the destination of the middle output of the current separator stage. This subkeyword is valid ONLY for the *PLANT stage type and not for the *EOS stage type.

*VAPOR-TO

Subkeyword introducing the destination of the vapor output of the current separator stage. This subkeyword is valid for both the *EOS and the *PLANT stage types.

*WET-GAS

The integer following *WET-GAS is the stage whose inlet stream (composition and rate) will be taken as the wet gas stream. If *WET-GAS is not entered, or if the short (first) format is used, the inlet stream to stage one is taken as the wet gas stream.

stage_no

The stage of the current separator to which the stage output whose destination is being specified is fed. stage_no must be larger than the stage number of the current stage. The stage numbering is the same as the order in which the stages are read in the data lines.

stage_number

Integer specifying a particular stage of the current separator. The stage numbering is the same as the order in which the stages are read in the data lines.

stream_name

As a stage outlet destination, this character string must be one of OIL, INL, or GAS. It specifies the surface stream which is the destination of the particular output of the current stage, or the stream for which the density

calculation is being specified. When used for the density calculation specification, WTG is also accepted as the stream_name. INL stands for the intermediate liquid stream, WTG for the wet gas stream.

pstream

When *EOS is specified as the surface stream mass density calculation method, a single-phase EOS density calculation is done at pressure pstream (kPa | psi).

tstream

When *EOS or *GASLAW is specified as the surface stream mass density calculation method, a single-phase EOS density calculation is done at temperature tstream (deg C | deg F).

root

Optional subkeyword of *STREAM-DEN *EOS specifying which root of the cubic EOS is used to compute the stream density. Root may be *LIQUID, *VAPOR, or *GIBBS. When *LIQUID is specified the smaller compressibility root is used; when *VAPOR is specified the larger; and when *GIBBS is specified the root giving the lower molar Gibbs free energy is chosen. The default is *LIQUID for the OIL and INL streams and *VAPOR for the GAS and WTG streams.

zfac

When the *GASLAW density calculation method is used, the relation PV = ZRT is applied, and mass_density = molecular_weight*P/ZRT. zfac (dimensionless) gives the value of the compressibility factor Z to be used in the density calculations.

*KEY

This subkeyword specifies that the named surface stream will have its mass density interpolated as a function of a key-component mole fraction from a table entered using the *DENSTR-KEY keyword.

*MIX

This subkeyword specifies that the named surface stream will have its mass density computed with a mixing rule for molar volumes using component mass densities entered using the *DENSTR-MIX keyword.

den_table_no

The number of the density table entered using the *DENSTR-KEY keyword which is to be used in the mass density calculation for the current stream.

den_set_no

The number of the component mass density set entered using the *DENSTR-MIX keyword which is to be used in the mass density calculation for the current stream.

DEFAULTS:

Optional keyword. The default initialization separator is a single stage separator using a two-phase EOS flash with the last EOS parameter set at the standard conditions of 101.325 kPa and 15.56 deg C or 14.7 psia and 60.0 deg F. (See the NOTE below however.)

GEM maintains an internal standard pressure pstd, a standard temperature tstd, and a standard EOS set number istd which are used to do an EOS calculation to establish surface conditions for all injection wells. These are first set as pstd = 14.7 psia or 101.325 kPa, tstd = 60.0 deg F or 15.56 deg C, and istd = last (largest) EOS set number, by default. If *SEPARATOR appears in the initial conditions data section and the last stage of this separator is of EOS type (as it always is if the shorter form of input is selected above), then pstd, tstd and istd are set to the pressure, temperature and EOS set number of this last stage. Note that the *INJECTOR-P, *INJECTOR-T and *INJECTOR-EOS keywords are available to override these items in either the INITIAL CONDITIONS section or the WELL AND RECURRENT DATA section.

Note:

If *SEPARATOR does NOT appear, BUT some or all of the *INJECTOR-P, *INJECTOR-T or *INJECTOR-EOS keywords are used to alter pstd, tstd, or istd, then these alterations will also be made for the default initialization separator.

CONDITIONS:

This keyword must be located in the INITIAL CONDITIONS keyword group.

It cannot be used if any of the keywords *BO, *BG, or *RS are used.

EXPLANATION:

When the first (shorter) form is used to enter the *SEPARATOR data, the stages are all of the EOS type. The destination of the liquid output of stage n is the feed of stage n+1 or the OIL stream if stage n is the last stage; the destination of the vapor output of all stages is the GAS stream. The OIL and GAS streams are the only surface streams allowed when the shorter form is used, with a single-phase EOS mass density calculation done for both streams at the pressure and temperature of the last stage using the last (highest index) set of EOS parameters entered.

When the longer format is used to enter the initialization separator, up to three surface hydrocarbon streams are supported: OIL, GAS, and INL (intermediate liquid).

Initial fluids-in-place are calculated using one of the following:

1. When the initial bo, bg and rs are entered via *BO, *BG and *RS, initial surface oil and gas volumes in place are calculated directly from the reservoir fluid volumes.
2. When a set of initial averaged separator conditions are entered via *SEPARATOR, the initial fluid-in-place volumes are calculated by passing all the reservoir fluid through this separator. If the separator has two surface streams, then initial oil and gas volumes in place are reported; if the separator has three output streams, then initial oil, gas, and intermediate liquid (INL) volumes in place are reported. This approach is preferred as it produces results consistent with the well production calculations.

Example 1:

To define a two-stage EOS separator, enter:

```
*SEPARATOR 10000.0  18.0
             101.3  15.6
```

Initial amounts in place of oil, gas, and water will be reported. The inlet stream to stage 1 is by default the wet gas stream.

Example 2:

The following two groups of data specify the same initial conditions separator, where the first set uses the first (shorter) form and the second set uses the longer form:

```
*SEPARATOR 100.    120.
            14.69   60.
```

-or-

```
*SEPARATOR *EOS 2 100. 120. *LIQUID-TO 2
                            *VAPOR-TO 'GAS'
           *EOS 2 14.69 60. *LIQUID-TO 'OIL'
                            *VAPOR-TO 'GAS'
  *STREAM-DEN 'GAS' *EOS 2 14.69 60.
              'OIL' *EOS 2 14.69 60.
```

For the second set, the *EOS after *SEPARATOR specifies that the separation in the first stage is to be done by a two-phase EOS flash using EOS parameter set 2 at pressure 100 psi and temperature 120 deg F., with the liquid from the flash feeding the second stage and the vapor going directly to the GAS surface stream. The *EOS on the third line indicates that the separator's second stage uses an EOS flash with parameter set 2 at pressure 14.69 psi and temperature 60 degrees Fahrenheit, with the liquid from that flash going to the OIL surface stream and the vapor going to the GAS surface stream. The *STREAM-DEN keyword indicates (1) that the separator has only two stages, since no more *EOS nor *PLANT lines are encountered; and (2) that both the GAS and OIL streams are to have their mass densities determined by a single-phase EOS calculation using parameter set 2 at 14.69 psi and 60 deg F. Initial amounts in place of oil, gas, and water will be reported.

Example 3:

Two-stage separator with one EOS stage and one plant stage :

```
*SEPARATOR *EOS 2 150. 120. *LIQUID-TO 'OIL'
                            *VAPOR-TO 2
           *PLANT 1         *LIQUID-TO 'OIL'
                            *MIDDLE-TO 'INL'
                            *VAPOR-TO 'GAS'
  *STREAM-DEN 'OIL' *KEY 1
              'INL' *MIX 1
              'GAS' *MIX 2
```

These lines specify that stage 1 of the initial conditions separator uses EOS parameter set number 2 in a two-phase flash at 150 psi and 120 deg F, that the liquid output from this flash goes directly to the OIL surface stream, and that the vapor output becomes the feed to stage 2. Stage 2 uses distribution fractions in plant table number 1 to determine the fluid split; the liquid output from this split goes to the OIL stream, the middle output to the INL (intermediate liquid) stream, and the vapor output of the split to the GAS surface stream. The mass density of the OIL surface stream is interpolated as a function of a key-component mole fraction from stream density table number 1; the INL stream has its density computed using a mixing rule with component mass density set number 1, and the GAS stream has its mass density computed using a mixing rule with component mass density set number two. Initial amounts in place of oil, intermediate liquid, gas, and water will be reported. The inlet to stage 2 (the plant stage) is interpreted as the wet gas stream. The wet gas mass density is computed using PV = ZRT with Z = 1.0, P = 14.69 psi, and T the Rankine equivalent of 60 degrees F.

Example 4:

A four-stage separator with 3 EOS stages and a single plant stage :

```
*SEPARATOR *EOS 2 150.  120.  *LIQUID-TO 2
                              *VAPOR-TO  3
           *EOS 3 30.    80.  *LIQUID-TO 'OIL'
                              *VAPOR-TO  3
           *EOS 4 14.69  60.  *LIQUID-TO 'INL'
                              *VAPOR-TO  4
           *PLANT 1           *LIQUID-TO 'OIL'
                              *MIDDLE-TO 'INL'
                              *VAPOR-TO 'GAS'
  *STREAM-DEN 'OIL' *EOS 3 14.69 60.
              'INL' *KEY 1
              'GAS' *MIX 1
```

Initial amounts in place of oil, intermediate liquid, gas, and water will be reported.

Specification of Non-EOS Stream Density Calculations (Optional) *DENSTR-KEY, *DENSTR-MIX

PURPOSE:

*DENSTR-KEY and *DENSTR-MIX introduce tables and component density sets respectively which allow the computation of the mass density of surface streams of separators without reference to a set of EOS parameters. These keywords allow entry of the tables and component mass density sets which are referred to under the *STREAM-DEN subkeyword of the *SEPARATOR and the *WELSEP keywords.

FORMAT:

```
*DENSTR-KEY  table_no
        *DENUNITS       (*LIQUID)
                        (*GAS)
        *KEYF  kf  *KEYL kl
        *KEYFRAC  y1 ... yn
        *STRDEN  d1 ... dn
and
*DENSTR-MIX set_no
        *DENUNITS       (*LIQUID)
                        (*GAS)
        *COMP-DEN  d(1) ... d(nc)
```

DEFINITIONS:

table_no

Integer which designates the table being entered as the table_no-th stream density table. This integer will in turn be used in data lines introduced by the *STREAM-DEN subkeyword of the *SEPARATOR and *WELSEP keywords to associate a separator surface stream with a method to compute its mass density. The tables must be read consecutively; e.g. if a table is read after table 4, its table_no must be 5 or a data error will result.

*DENUNITS

Subkeyword which introduces another subkeyword which specifies whether, when field units are used, the entered density values are expressed in lbm/ft3 or in lbm/bbl. If *DENUNITS does not appear or *DENUNITS appears with no following subkeyword, lbm/bbl is assumed. If SI units are used, then this subkeyword has no effect, and may be omitted.

*LIQUID

Subkeyword of *DENUNITS which specifies that, if field units are used, the entered density values are expressed in the unit system which is appropriate for liquid surface streams, namely lbm/bbl. *OIL and *INL are also accepted in place of *LIQUID. Specifying *LIQUID under *DENUNITS does not restrict the use of the table or component density set to liquid surface streams; for a gas stream the appropriate unit conversion is done.

*GAS

Subkeyword of *DENUNITS which specifies that, if field units are used, the entered density values are expressed in the unit system which is appropriate for gas surface streams, namely lbm/ft3. *SOLVENT is also accepted in place of *GAS. Specifying *GAS under *DENUNITS does not restrict the use of the table or component density set to gas surface streams; for a liquid stream the appropriate unit conversion is done.

*KEYF

Subkeyword to introduce the hydrocarbon component number of the first component which is part of the "key component", which is in general actually a set of consecutively numbered components.

kf

Integer index of the first hydrocarbon component which is included in the "key component".

*KEYL

Subkeyword to introduce the hydrocarbon component number of the last component which is included in the "key component".

kl

Integer index of the last hydrocarbon component which is included in the "key component". kl must be greater than or equal to kf and the mole fractions of components kf, ..., kl in the plant feed are summed to obtain the key component mole fraction.

*KEYFRAC

Subkeyword which introduces the set of n values of the key component mole fraction to which the following input line of stream mass densities applies. There is no explicit upper limit to the integer n, since the tables are stored consecutively in a singly-subscripted array.

y1 ... yn

The n values of the key component mole fraction for which stream mass density values will be entered on the next data line. y1 ... yn must be a monotonically increasing list of dimensionless real numbers whose values are between zero and one. GEM determines the value of n by counting the number of real numbers which follow *KEYFRAC before *STRDEN is found. If n = 1 the stream mass density is independent of the key component mole fraction. If no mole fraction values are entered (n = 0) an error is detected and the run is terminated.

*STRDEN

Subkeyword which introduces the set of n values of the stream mass density.

d1 ... dn

The n values of stream mass density in (kg/m3 | lbm/bbl) or (kg/m3 | lbm/ft3) depending on what is entered under or defaulted for the *DENUNITS keyword as explained above. These values correspond in order to the values y1 ... yn entered on the preceding line. These values must be non-negative

real numbers. GEM determines how many values of the stream mass density are entered by counting the real numbers entered after *STRDEN and before the next keyword; if this number is not equal to the n determined in reading y1 ... yn, an error is generated.

set_no

Integer which designates the component mass density set being entered as the set_no-th set. This integer will in turn be used in data lines introduced by the *STREAM-DEN subkeyword of the *SEPARATOR and *WELSEP keywords to associate a separator stream with a method to compute its mass density. The component mass density sets must be read consecutively; e.g. if a set is read after set 4, its set_no must be 5 or a data error will be generated. These component mass density sets are used ONLY for the mass density calculations for the designated streams; they do not affect any EOS calculations or reservoir transport equations of any kind.

*COMP-DEN

Subkeyword which introduces the set of nc component mass densities, where nc is the number of hydrocarbon components entered in the FLUID PROPERTIES keyword section.

d(1) ... d(nc)

The set of nc values of component mass density, expressed in (kg/m3 | lbm/ft3) or (kg/m3 | lbm/ft3) as set or defaulted under the *DENUNITS subkeyword as explained above. These values are used as described in the EXPLANATION section below to compute the separator stream mass density as a function of composition. These must be positive real values. If a zero is entered an error is generated.

DEFAULTS:

If table_no is not entered after *DENSTR-KEY, the current last table index is incremented by one and the incremented index is used. Similar remarks hold for *DENSTR-MIX and the set_no.

CONDITIONS:

This keyword may appear in either the INITIAL CONDITIONS keyword group or the WELL AND RECURRENT DATA keyword group. A density table or component density set cannot be modified once it has been read; a user wanting to change the density table or the component mass density set used by a given separator stream must enter a new table or set with number exceeding the currently last table or set number by one and redefine the stream density calculation to refer to the new table or set.

EXPLANATION:

The *STREAM-DEN subkeyword of the *SEPARATOR and *WELSEP keywords allows three different methods to be specified for computing the mass density of a separator's surface streams : (1) Single-phase EOS calculation, (2) interpolation in a density table as a function of a key-component mole fraction, or (3) from a mixing rule averaging component molar volumes which are computed from user-entered component mass densities.

Method (1) uses the EOS parameters entered as described in the FLUID PROPERTIES section of this manual and is not described further here.

When method (2) is specified, the mole fractions of components kf through kl in the separator stream are summed to give the stream key-component mole fraction ky (the molar rates of all components in the stream are assumed to be known). If there are consecutive tabular key-component mole fractions yi and yi+1 such that yi < ky < yi+1, then the density is interpolated linearly as

$$\text{density} = \frac{(\text{ky} - \text{yi})}{(\text{yi}+1-\text{yi})} * \text{di}+1 + \frac{(\text{yi}+1-\text{ky})}{(\text{yi}+1-\text{yi})} * \text{di}$$

If ky < y1 or ky > dn, no extrapolation is done and the stream mass density is set to d1 or dn, respectively.

When method (3) is specified, the user-entered component mass densities are used to compute component molar volumes v(m) (m3/gmol | ft3/gmol),

v(m) = M(m)/d(m),

where M(m) is the molecular weight of component m (kg/gmol | lbm/gmol) and d(m) is the user-entered component mass density. The stream molar volume is then computed from a linear mixing rule,

V(stream) = Sum(m = 1,nc) x(m)*v(m) ,

where x(m) is the mole fraction of component m in the separator stream, and the mass density D(stream) (kg/m3 | lbm/ft3) is then calculated as

D(stream) = M(stream)/V(stream)

where M(stream) is the average molecular weight of the stream,

M(stream) = Sum(m = 1,nc) x(m)*M(m) .

Unit conversion to barrels as the volume unit are then done if necessary.

Example 1:

```
*DENSTR-KEY 3
   *KEYF 5 *KEYL 6
**            y1      y2      y3
   *KEYFRAC   0.0     0.05    0.1
**            d1      d2      d3
   *STRDEN    320.3   321.1   322.2
```

The mole fractions y1, y2, y3 are to be compared with sums of the mole fractions of the fifth and sixth hydrocarbon components. The densities d1, d2, and d3 are in lbm/bbl ; the table could be used for OIL or INL streams, both of which are measured in liquid units. This table could not be used for a GAS stream, as the units are not correct.

Example 2:

```
*DENSTR-MIX 2
**              C1    C2   C3-C4  C5-C6  C7-C9  C10+
   *COMP-DEN   7.47  14.0  23.8    36.9   53.2  105.5
```

If field units are used, the component mass densities in the above are expressed in lbm/ft3. If the resulting mixing rule is used for a GAS stream, the density values are returned directly with the units lbm/ft3; if the mixing rule is used for an OIL or INL stream, the density value is converted to lbm/bbl.

Standard Pressure, Temperature, and EOS Set (Optional)

***INJECTOR-P, *INJECTOR-T, *INJECTOR-EOS**

PURPOSE:

*INJECTOR-P and *INJECTOR-T specify the standard pressure and temperature used internally in GEM for converting surface rates to reservoir rates for injection wells. These keywords also affect the default initialization separator if *SEPARATOR does not appear.

FORMAT:

```
*INJECTOR-P      pstd
*INJECTOR-T      tstd
*INJECTOR-EOS    istd
```

DEFINITIONS:

pstd
: The standard pressure (kPa | psi).

tstd
: The standard temperature (deg C | deg F).

istd
: The EOS set number to be used with pstd, tstd. This must be a number of an EOS parameter set entered using the *EOSSET keyword in the COMPONENT PROPERTIES section of the data set.

DEFAULTS:

Initially, GEM internally sets pstd = 14.7 psia or 101.325 kPa and tstd = 60.0 deg F. or 15.56 deg C by default, and sets istd to the number of the last (largest) EOS set entered.

Then if *SEPARATOR appears in the INITIAL CONDITIONS DATA section, and it appears BEFORE these keywords, and the last stage of this separator is of EOS type (as it always is if the shorter form of input is selected; see the *SEPARATOR keyword), then pstd, tstd and istd are set to the pressure, temperature and EOS set number of this last stage.

These default settings will then be overridden by values entered using the *INJECTOR-P, *INJECTOR-T, and *INJECTOR-EOS keywords.

Note:

If *SEPARATOR does not appear, BUT some or all of the *INJECTOR-P, *INJECTOR-T or *INJECTOR-EOS keywords are used to alter pstd, tstd, or istd, then these alterations will also be made for the default initialization separator (see *SEPARATOR keyword).

Thus, these keywords set the default initialization separator, as well as set injection conditions, if the *SEPARATOR keyword does not appear.

To properly take effect, these keywords should appear after any *SEPARATOR keyword, so that the possible use of a final EOS stage for the initialization separator does not override any of pstd, tstd or istd being set here.

CONDITIONS:

These keywords may appear in either the INITIAL CONDITIONS keyword group or the WELL AND RECURRENT DATA keyword group. See the Notes above.

EXPLANATION:

The quantities pstd, tstd, and istd are used in the following ways:

1. For converting between surface and reservoir rates for injection wells through an EOS fluid properties calculation;
2. For converting between surface and molar rates for injection wells;
3. For converting between make-up gas surface and molar rates in gas recycling.
4. For computing relative reservoir fluid volumes;

The parameters pstd and tstd are not necessarily used in computing the surface rates of producing wells; see the explanations of the *STREAM-DEN subkeyword of the *SEPARATOR keyword for more information on how producer surface rates are calculated.

Example:

```
*INJECTOR-P 100.
*INJECTOR-T 15.
```

If SI units are used, this sets the standard pressure to 100.0 kPa and the standard temperature to 15.0 degrees Celsius.

Table Defining a Gas Plant (Optional) *PLANT-TABLE

PURPOSE:

*PLANT-TABLE introduces a table of distribution fractions which determine the amounts and compositions of fluid outputs from a gas plant. Such gas plants may be used in determining initial volumes of fluids at surface conditions in place in the reservoir (see *SEPARATOR keyword above).

A plant table in general consists of several sub-tables, with each sub-table containing distribution fractions for a different value of a key-component mole fraction. The plant is assumed to have three output streams, namely liquid, vapor, and middle. If fewer streams are needed, the user should enter zeros as distribution fractions for the unwanted output for all components.

FORMAT:

```
*PLANT-TABLE  table_no
        *KEYF  kf  *KEYL  kl
        *KEYFRAC  key_component_mole_fraction
           Repeated distribution_fraction_lines
```

where the distribution_fraction_lines (one for each hydrocarbon component) have the form :

```
liquid_fraction  vapor_fraction  middle_fraction
```

The m-th distribution_fraction_line for a given value of key_component_mole_fraction gives the distribution fractions for component m, m = 1, ... , nc, where nc is the number of hydrocarbon components entered in the FLUID PROPERTIES section.

DEFINITIONS:

table_no
: Integer which designates the table being entered as the table_no-th plant table. This integer will in turn be used in data lines introduced by the *SEPARATOR and *WELSEP keywords to associate a separator stage with a table of distribution fractions. The tables must be read consecutively; e.g. if a table is read after table 4, its table_no must be 5 or a data error will be reported.

*KEYF
: Subkeyword to introduce the hydrocarbon component number of the first component which is part of the "key component", which is in general actually a set of consecutively numbered components.

kf
: Integer index of the first hydrocarbon component which is included in the "key component".

*KEYL
: Subkeyword to introduce the hydrocarbon component number of the last component which is included in the "key component".

kl

Integer index of the last hydrocarbon component which is included in the "key component". kl must be greater than or equal to kf and the mole fractions of components kf, ..., kl in the plant feed are summed to obtain the key component mole fraction.

*KEYFRAC

Subkeyword which introduces the value of the key component mole fraction to which the following input lines of distribution fractions apply.

key_component_mole_fraction

A real number between zero and one inclusive which gives the value of the sum of the mole fractions of components kf through kl in the plant feed for which the following lines of distribution fractions hold exactly. In a given occurrence of the *PLANT-TABLE keyword, the key_component_mole_fractions must be entered in monotonically increasing order.

liquid_fraction

Real number which gives the mole (equivalently the mass) fraction of the current component which exits in the liquid output stream of the gas plant. The liquid_fraction, middle_fraction, and vapor_fraction for each component are normalized internally to sum to one. The value entered may be zero but must not be negative.

vapor_fraction

Real number which gives the mole (equivalently the mass) fraction of the current component which exits in the vapor output stream of the gas plant. The liquid_fraction, middle_fraction, and vapor_fraction for each component are normalized internally to sum to one. The value entered may be zero but must not be negative.

middle_fraction

Real number which gives the mole (equivalently the mass) fraction of the current component which exits in the middle output stream of the gas plant. The liquid_fraction, middle_fraction, and vapor_fraction for each component are normalized internally to sum to one. The value entered may be zero but must not be negative.

DEFAULTS:

If table_no is not entered after *PLANT-TABLE, the current last table index is incremented by one and the incremented index is used. If zeros are entered for all of liquid_fraction, middle_fraction, and vapor_fraction on one line of a plant table, then liquid_fraction = middle_fraction = vapor_fraction = 1/3 is defaulted.

CONDITIONS:

This keyword may appear in either the INITIAL CONDITIONS keyword group or the WELL AND RECURRENT DATA keyword group. A table cannot be modified once it has been read; a user wanting to change the plant table used by a given separator must enter a new table with number exceeding the current last table number by one and re-define the separator to refer to the new table.

EXPLANATION:

*PLANT can be specified as the stage type for a surface separator defined either with the *SEPARATOR keyword in the INITIAL CONDITIONS section or the *WELSEP keyword in the WELL AND RECURRENT DATA section. In either case *PLANT is followed by a table number in which are to be found the distribution fractions which are to be used to describe the gas plant stage. These tables are entered by the user with the *PLANT-TABLE keyword.

The gas plant has a given feed composition and total molar rate, from which can be calculated feed molar rates for each component. The output molar rate of component m in the liquid gas plant output is then computed as

liquid_molar_rate(m) = feed_molar_rate(m)*
liquid_fraction(m) ;

the expressions for the molar rate of component m in the other two outputs (middle and vapor) are exactly analogous. The liquid output is intended nominally to correspond to an oil, the middle to a natural gas liquid, and the vapor to a light gas output, but there is no restriction as to the use of these names and the user has free choice of the distribution fractions used for the outputs.

If the sum kx of the mole fractions of the components included in the "key component" lies between two of the tabular values the distribution fractions are determined by linear interpolation between the two bracketing mole fraction values k1 < kx < k2 :

$$\mathrm{frac}(\mathrm{kx}) = \frac{(\mathrm{kx}-\mathrm{k1})}{(\mathrm{k2}-\mathrm{k1})} * \mathrm{frac}(\mathrm{k2}) + \frac{(\mathrm{k2}-\mathrm{kx})}{(\mathrm{k2}-\mathrm{k1})} * \mathrm{frac}(\mathrm{k1}),$$

where frac stands for liquid_fraction, middle_fraction, or vapor_fraction. Note that this interpolation preserves the property that the fractions sum to one. If kx is smaller than the smallest key mole fraction entered after *KEYFRAC, then the distribution fractions for the smallest entered mole fraction are used without extrapolation. Similarly no extrapolation is done if the key component mole fraction in the feed exceeds the largest key component mole fraction entered after *KEYFRAC.

Example:

```
*PLANT-TABLE 2
    *KEYF 5 *KEYL 6
    *KEYFRAC 0.0
** liq_frac  vap_frac  mid_frac
      0.1      0.7       0.2      ** Component 1
      0.12     0.6       0.28     ** Component 2
      0.17     0.4       0.43     ** Component 3
      0.3      0.25      0.45     ** Component 4
      0.5      0.1       0.4      ** Component 5
      0.9      0.01      0.09     ** Component 6
*KEYFRAC 0.05
** liq_frac  vap_frac  mid_frac
      0.1       0.7       0.2     ** Component 1
      0.15      0.55      0.3     ** Component 2
      0.23      0.27      0.50    ** Component 3
      0.35      0.10      0.55    ** Component 4
      0.60      0.05      0.35    ** Component 5
      0.92      0.01      0.07    ** Component 6
```

All of the lines in the table above have liq_frac + mid_frac + vap_frac = 1; had this not held for some line, the liquid fraction would have been replaced internally with the normalized quantity

$$liq_frac_new = \frac{liq_frac}{(liq_frac + mid_frac + vap_frac)}$$

and analogously for mid_frac_new and vap_frac_new. GEM concludes that exactly two different values of the key component mole fraction have been entered in this table when it encounters a keyword other than *KEYFRAC after the nc-th line following *KEYFRAC 0.05.

Critical Depth (Conditional) *CDEPTH

PURPOSE:

*CDEPTH indicates the critical depth.

FORMAT:

*CDEPTH depth(i), i = 1, nreg

DEFINITIONS:

depth

Critical depth (m | ft). It is used with *VERTICAL *COMP (equivalent to *VERTICAL *BLOCK_CENTER *COMP) to determine the phase identity, oil or gas, of a single-phase fluid. A single phase hydrocarbon at any depth greater than the critical depth is identified as an oil, while a single phase hydrocarbon at a depth less than the critical depth is called gas.

nreg

The number of initialization regions into which the reservoir has been divided.

DEFAULTS:

Conditional keyword. Default values are: -1.0d+20 for *PHASEID *OIL and +1.0d+20 for *PHASEID *GAS.

If *PHASEID is not entered, there is no default for *CDEPTH. The *PHASEID keyword is entered in the Component Properties data section and is explained in that section of this manual.

CONDITIONS:

If it appears, this keyword must be in the INITIAL CONDITIONS keyword group. If it appears, and if *NGREGIONS also appears, then this keyword must follow *NREGIONS.

It is required with *VERTICAL *COMP (equivalent to *VERTICAL *BLOCK_CENTER *COMP). It must appear AFTER the *VERTICAL *COMP line. *CDEPTH can ONLY be used with the *COMP subkeyword of *VERTICAL; the other options of *VERTICAL do not support *CDEPTH .

EXPLANATION:

The *CDEPTH keyword defines the critical depth when *VERTICAL *COMP is used for initialization. For gas reservoirs, set depth to a value larger than the depth of the bottom of the reservoir. For oil reservoirs, set depth to zero.

Example:

```
For nreg = 1:
     *CDEPTH 5000.
For nreg = 3:
     *CDEPTH 5000. 4500. 4750.
```

Variation of Composition with Depth (Conditional) *ZDEPTH

PURPOSE:

*ZDEPTH indicates the variation of composition with depth

FORMAT:

*ZDEPTH (ireg) depth zoil

DEFINITIONS:

ireg
: Index of the initialization region for which the current specification of composition as a function of depth is to apply.

depth
: Depth (m | ft)

zoil
: Nc real numbers representing the composition of the nc components. If the entered numbers sum to between 0.9 and 1.1 (or to between 90 and 110), they are normalized to one; otherwise, an error is issued. nc is defined in the Component Properties section of this manual.

DEFAULTS:

Conditional keyword. The default for ireg is one if no region index has previously been read. If no integer is encountered on the line it is interpreted as belonging to the initialization region most recently specified with an integer.

CONDITIONS:

If it appears, this keyword must be in the INITIAL CONDITIONS keyword group. If it appears, and if *NREGIONS also appears, then *ZDEPTH must follow *NREGIONS.

*ZDEPTH must appear only once in the input of the depth-dependent compositions; the depth and composition information at the first depth may follow *ZDEPTH on the same line, but subsequent depth-composition lines must NOT start with *ZDEPTH.

*ZDEPTH is required with *VERTICAL *COMP, which is equivalent to *VERTICAL *BLOCK_CENTER *COMP.

The *ZDEPTH is also supported as one of the composition specification methods for *DEPTH_AVE initializations. Appearance of a *ZDEPTH line with initialization options other than *COMP and *DEPTH_AVE results in an error.

EXPLANATION:

The depth-zoil entries can be repeated to enter up to 100 compositions for all initialization regions together.

All depth-zoil lines must be entered in order of increasing depth within a given initialization region. No extrapolation of composition outside the depth range of the depth-zoil table is performed. Thus, if a grid block in a given initialization region has depth larger than the largest depth in the depth-zoil table for that region, then the composition at the last entry of

the table is assigned. If the depth-zoil table has only one depth entry, the composition is constant throughout the initialization region.

All of the depth lines for a single initialization region must be grouped consecutively.

Examples:

5 component fluid, single initialization region:

```
**       depth      composition
*ZDEPTH  1000.    0.33  0.20  0.12  0.25  0.10
         2000.    0.31  0.19  0.12  0.26  0.12
         3000.    0.29  0.19  0.11  0.26  0.15
         4000.    0.27  0.18  0.12  0.27  0.16
```

All blocks with depths smaller than 1000 will have compositions equal to that at 1000. All blocks with depth larger than 4000 will have compositions equal to those at 4000. The initialization region index ireg need not appear since it defaults to one.

3 component fluid, 3 initialization regions:

```
*ZDEPTH  1  1000.  0.34  0.33  0.33
            4000.  0.80  0.10  0.10
         2  1000.  0.60  0.30  0.10
            4000   0.80  0.10  0.10
         3  2000.  0.50  0.30  0.20
            4000.  0.90  0.05  0.05
```

Note that ireg may be omitted where it has not been incremented. The 1 specifying in the first line could have been omitted since the default would be 1 on the first line.

Datum Depth Specification (Optional) *DATUMDEPTH

PURPOSE:

*DATUMDEPTH specifies the datum depth. It is used for printout of datum pressure, pressure corrected to datum.

FORMAT:

```
*DATUMDEPTH    depth    (*INITIAL)
                        (*REFDENSITY  density)
                        (*REFDENSITY  *GRIDBLOCK)
```

DEFINITIONS:

depth
: A real number denoting the depth to the datum. (m | ft)

*INITIAL
: Keyword indicating that the initial equilibrium pressure distribution in the reservoir will be used to calculate the corrected datum pressures. This is the default.

*REFDENSITY
: Keyword indicating that the entered density should be used to calculate the corrected datum pressures.

density
: A real number denoting the density to be used for calculating the corrected pressures (Kg/m^3 | lb/ft^3). This entered density will be used to correct for the gravity head for every grid block (in the gas cap; in the oil zone; and in the water zone).

*GRIDBLOCK
: Keyword specifying that the grid block density should be used to calculate the datum pressures.

DEFAULTS:

Optional keyword. If this keyword is not present and the output of datum pressures is requested then no datum corrected pressure printout will appear. If *DATUMDEPTH is present without *INITIAL or *REFDENSITY then the default is *INITIAL.

If *REFDENSITY is specified without density or *GRIDBLOCK then the default is *GRIDBLOCK.

CONDITIONS:

This keyword if present must be in the INITIAL CONDITIONS keyword group.

EXPLANATION:

*DATUMDEPTH modifies the individual grid block pressures to a common depth by removing the gravity head from the pressures. This removal of the gravity head can be accomplished in three ways:

1. If *INITIAL is used (this is the default) then the pressures are corrected based on the initial equilibrium pressure distribution of the reservoir.

   datum_pressure(ib) = pressure(ib) - pressure_0(ib) + datum_pressure_0
   Here:
   datum_pressure(ib) is the corrected datum pressure for grid block ib.
   pressure(ib) is the actual pressure at grid-block ib.
   pressure_0(ib) is the grid block pressure at initial time (time = 0).
   datum_pressure_0 is the datum pressure at initial time.

   Please note that the above calculation assumes that:

   a) The phase densities do not change by a large amount during the simulation.
   b) The water-oil and gas-oil contacts do not move by a large amount.

   The above assumptions are valid for most Black-Oil simulation studies.

2. If *REFDENSITY density is used then the pressures are corrected to datum as follows.

   datum_pressure(ib) = pressure(ib) + density*g*(datumdepth - depth(ib))
   Here:
   density is the density specified using the *REFDENSITY keyword above.
   depth(ib) is the depth of grid block ib.
   g is the acceleration due to gravity.

3. If *REFDENSITY *GRIDBLOCK is used then the pressures are corrected to datum as follows.

   datum_pressure(ib) = pressure(ib) + density(ib)*g*(datumdepth - depth(ib))
   Here:
   density(ib) is the oil phase density of grid block ib if the oil phase is present in the block. If the oil phase is not present in that grid block then it is the gas phase density of grid block ib if the gas phase is present. If neither the oil nor the gas phases are present then density(ib) is the water density of grid block ib.

Examples:

```
1. *DATUMDEPTH 1000.0
2. *DATUMDEPTH 1000.0  *INITIAL
3. *DATUMDEPTH 1000.0  *REFDENSITY
4. *DATUMDEPTH 1000.0  *REFDENSITY *GRIDBLOCK
5. *DATUMDEPTH 1000.0  *REFDENSITY 871.0
```

Examples 1 and 2 above are equivalent.

Examples 3 and 4 above are equivalent.

To print out the datum pressures, use the subkeyword *DATUMPRES with *OUTPRN *GRID.

Specification of Capillary Pressure Values at Contacts (Optional) *GOC_PC, *WOC_PC *WGC_PC

PURPOSE:

*GOC_PC, *WOC_PC and *WGC_PC introduce the value of the oil-gas capillary pressure at the gas-oil contact, the value of the water-oil capillary pressure at the oil-water contact and the value of the gas-water capillary pressure at the water-gas contact respectively. These values are used in the *VERTICAL *DEPTH_AVE initialization options.

FORMAT:

*GOC_PC pcggoc(i), i = 1, nreg

*WOC_PC pcwwoc(i), i = 1, nreg

*WGC_PC pcwgwc(i), i = 1, nreg

DEFINITIONS:

pcggoc
: A real number specifying the value of the oil-gas capillary pressure at the gas-oil contact in the i-th initialization region (kPa | psi).

pcwwoc
: A real number specifying the value of the water-oil capillary pressure at the water-oil contact in the i-th initialization region (kPa | psi).

pcwgwc
: A real number specifying the value of the gas-water capillary pressure at the water-gas contact in the i-th initialization region (kPa | psi).

nreg
: The number of initialization regions into which the reservoir has been divided.

DEFAULTS:

Optional keywords. If this keyword is not present then the *USER_INPUT and all *VERTICAL *BLOCK_CENTER initialization options are unaffected because pcggoc and pcwwoc do not enter these initializations. For the *VERTICAL *DEPTH_AVE options the default values are pcggoc = 0 , pcwwoc = 0 and pcwgwc = 0 for water wet case.

For the oil wet case pcggoc =0 and pcwgwc=0, however pcwwoc = Pw-Po is set to equal the minimum value of Pcwo=Pw-Po specified via the SWT table applicable to the specific initialization region under consideration. For example if the following SWT table is assigned to a given initialization region and pcwwoc is NOT specified via the keyword WOC_PC then the value assigned to pcwwoc = Pw-Po = -5.0

```
SWT
** So        **kro   **krow  **Pcwo = Pw-Po
       0.0500    0.0      0.80     20.0
       0.2000    0.02    0.56     19.0
       0.3000    0.08    0.34     17.9
       0.4000    0.18    0.18     16.8
       0.5000    0.34    0.08     11.6
       0.6000    0.48    0.02      7.6
       0.7000    0.64    0.0        1.9
       0.8000    0.64    0.0       -2.7
       0.8831    0.64    0.0       -5.0
```

CONDITIONS:

Optional keywords. These keywords, if present, must be in the INITIAL CONDITIONS keyword group. If they appear, and if *NREGIONS also appears, then these keywords must follow *NREGIONS.

No error message is printed if a *VERTICAL *DEPTH_AVE option is chosen but *GOC_PC or *WOC_PC or *WGC_PC does not appear in the data; the values of pcggoc or pcwwoc or pcwgwc are set to their respective defaults and the initialization proceeds.

EXPLANATION:

Under the *DEPTH_AVE initialization options, the water-oil contact is defined as the depth at which the water-oil capillary pressure has the value pcwwoc. Normally this value is zero, which is the default value in GEM for water wet case, but the *WOC_PC keyword allows the user to set any desired value for the water-oil capillary pressure at the water-oil contact depth. Under the *BLOCK_CENTER initialization options the water-oil contact depth is defined by saturation values rather than by capillary pressure values and pcwwoc does not enter the computation, with capillary pressures determined from the established values of the saturations. If *WOC_PC is entered when a *BLOCK_CENTER option is invoked, the entered pcwwoc value is ignored.

*GOC_PC allows the user to specify a value for pcggoc, the value of the oil-gas capillary pressure at the gas-oil contact. Again the default value is zero. Under the *DEPTH_AVE initialization options, the gas-oil contact depth is defined as the depth at which the oil-gas capillary pressure has the value pcggoc. This keyword is ignored under the *USER_INPUT and the *VERTICAL *BLOCK_CENTER options.

Similarly, *WGC_PC allows the user to specify a value for pcggwc, the value of the gas-water capillary pressure at the water-gas contact. Again the default value is zero. Under the *DEPTH_AVE initialization options, the water-gas contact depth is defined as the depth at which the gas-water capillary pressure has the value pcggwc. This keyword is ignored under the *USER_INPUT and the *VERTICAL *BLOCK_CENTER options.

It is important that the values of *GOC_PC, *WOC_PC, and *WGC_PC, if entered, must lie within the range of corresponding tabular entries of capillary pressures in the Rock-Fluid data section.

Examples:

For 3 initialization regions

```
*GOC_PC  -0.5  0.0   0.5
*WOC_PC   0.5  0.4  10.0
*WGC_PC   0.5  0.2   3.5
```

Water Saturation Input Used with Gravity-Capillary Initialization (Conditional) *SWINIT

PURPOSE:

*SWINIT indicates input of initial water saturation to be used with *BLOCK_CENTER and *DEPTH_AVE gravity-capillary equilibrium initialization. *SWINIT values can be different from connate water saturation (fraction).

ARRAY:

*SWINIT

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. It can be used with both *BLOCK_CENTER and *DEPTH_AVE initialization options.

Care has to be taken when this option is employed, reasonable values of saturation must be entered, as highly altered Pcow curves will significantly affect other aspects of the model.

For *BLOCK_CENTER initialization, the user must be using rock curves with oil-water capillary pressure for this option to have the desired effect, as the *SWINIT keyword scales Pcow to establish vertical equilibrium at *SWINIT. This is not a limitation when using *DEPTH_AVE initialization as a capillary pressure offset is used to maintain equilibrium.

Various limitations exist on how *SWINIT can modify vertical equilibrium, these are listed below:

In the Gas Cap, *SWINIT can vary between zero and the critical (not connate) water saturation. If *SWINIT greater than critical water saturation is found, Sw in that block will be set equal to the critical water saturation.

In the oil zone, *SWINIT must be greater than the connate water saturation. If *SWINIT less than connate is found, Sw in the block will be set equal to connate water saturation, Pcow will be scaled but the block will not be in equilibrium.

*BLOCK_CENTER initialization only: If, in an oil zone block, original Pcow or scaled Pcow is less than 0.1 kPa (0.0145 Psi) no scaling will be done and a warning will be issued.

*BLOCK_CENTER initialization only: If Pcow, in an oil zone block, must be scaled by a factor greater than 200 or by a factor less than 0.005 to be in equilibrium, the factor will be limited to 200 or 0.005 and a warning will be issued. Pcow will be scaled by the reduced factor, but the fluids will not be in equilibrium.

*DEPTH_AVE initialization only: SWINIT does not scale an existing Pcow curve but adds an offset to the curve (even if Pcow is zero at all saturations). This eliminates the two restrictions above.

Below the water-oil contact, *SWINIT values are ignored. Please see keyword entry *SWOC for specification of waster saturation below the water-oil contact.

In summary

Gas Cap:

0 < Swinit < Sw(critical)
Sw reset to Sw(critical) if > Sw(critical)

Oil Zone:

Sw reset to Sw(connate) if Sw < Sw(connate)
Pcow scaled only if both original Pcow and scaled Pcow > 0.1 kPa
(*BLOCK_CENTER initialization only)
Pcow scaling is limited by a maximum scaling factor of 200 and a minimum scaling factor of 0.005. Scaling will occur with the maximum or minimum factors but the grid blocks will not be in equilibrium (*BLOCK_CENTER initialization only).

Water Zone below Water-Oil Contact:

Swinit entry is ignored. Use *SWOC, if required.

EXPLANATION:

The *SWINIT keyword is used with the *BLOCK_CENTER or *DEPTH_AVE equilibrium. It allows the user to input all of the required keywords for *BLOCK_CENTER or *DEPTH_AVE equilibrium option as well as initial water saturation.

*SWINIT initial water saturation is used to modify the Pcow curve in such a way so that the initial water saturation values are honored.

Non Equilibrium Water and Oil Saturation Used with Gravity-Capillary Equilibrium Initialization (Conditional) *SWNEQ

PURPOSE:

*SWNEQ indicates input of non-equilibrium initial water saturation override to be used with gravity-capillary equilibrium initialization.

ARRAY:

*SWNEQ

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group.

*SWNEQ need not be defined for every block in the reservoir, just the blocks where the user requires that the initial saturations obtained from the gravity-capillary initialization are overridden. This input is not valid for USER_INPUT initialization.

EXPLANATION:

The *SWNEQ keyword allows the user to override the equilibrium water saturations obtained from gravity-capillary initialization in a portion of the reservoir. An example of its use would be to initialize the saturations in and near a water filled fracture, created during a well stimulation procedure which is not to be explicitly modeled. The entire reservoir would be initialized using equilibrium calculations while a user defined area surrounding the fractured well might be given very high water saturation. The pressures in the overridden blocks are still obtained from the equilibrium initialization.

The *IJK reading option is used to precisely define the region of saturation override. Areas not to be overridden must not be defined in the SWNEQ *IJK range.

The "*ALL" array input option maybe used with *SWNEQ. Values of *SWNEQ which are not to be overridden, but which must be included due to the "*ALL" input format, must be given the value of -1.0.

Example:

```
*VERTICAL *DEPTH_AVE
*SWNEQ *IJK
     4:28 14:18 4:6   0.8
     5:27 15:17 4:6   0.9
     6:26 16:16 5:5   0.99
```

Specification of Method for Computing Oil Saturations in Gas Zone *GASZONE

PURPOSE:

*GASZONE specifies whether a residual oil saturation is to be set initially in the uppermost part of the reservoir where the difference in the oil and gas phase pressures exceed the largest tabulated value the oil-gas capillary pressure. This keyword applicable only when the *DEPTH_AVE sub-option of *VERTICAL is in effect.

FORMAT:

```
*GASZONE     (*OIL)
             (*NOOIL)
```

DEFINITIONS:

*OIL

Indicates that oil is to be allowed in the gas zone above the upper limit of the transition zone. See the EXPLANATION section below for details.

*NOOIL

Indicates that an initial oil saturation of zero is to be assigned in the uppermost regions of the reservoir where the computed difference in the pressures of the oil and gas phases exceeds the largest value which appears in the oil-gas capillary pressure table (i.e. above the upper limit of the transition zone).

DEFAULTS:

Optional keyword. If *GASZONE is not present then *NOOIL is assumed and oil saturations of zero are assigned initially in the gas zone. If *GASZONE is encountered with no following subkeyword, then *GASZONE *NOOIL is assumed.

CONDITIONS:

This keyword, if present, must be in the INITIAL CONDITIONS keyword group. It applies only for the *VERTICAL *DEPTH_AVE initialization options. If *GASZONE is entered when the *USER_INPUT or *BLOCK_CENTER initialization option is specified, then an error message is issued. The *BLOCK_CENTER initialization always uses the *NOOIL option, and this is not alterable through data.

EXPLANATION:

If *GASZONE *NOOIL is specified and the computed pressure difference between the oil and gas phases exceeds the tabulated values on the capillary pressure curve, then the water saturation is determined first from the oil-water phase pressure difference, and Sg is set to min(Sg(tab), 1-Sw). So is set to zero.

If *GASZONE *OIL is specified and the computed pressure difference between the oil and gas phases exceeds the tabulated , then Sw is determined first from the oil-water phase pressure difference, and Sg is set to min(Sg(tab), max(1-Sw-So(min),0) where So(min) = max(Sorg+Swc-Sw,0). So is set 1-Sw-Sg.

If *GASZONE *OIL is specified and the computed pressure difference between the oil and gas phases exceeds the tabulated values, then Sw is determined first from the oil-water phase pressure difference, Sg is set to min(Sg(tab), max(1-Sw-So(min),0)), where So(min) = max(Sorg+Swc-Sw,0). So is set to 1 - Sw - Sg.

Example:

```
*GASZONE  *OIL
```

Critical Temperature Multiplier (Optional) *TCMULT

PURPOSE:

*TCMULT indicates input of a critical temperature multiplier used with the single phase identification method *PHASEID *TCMIX and with single hydrocarbon phase relative permeability interpolation controlled by *TRLIMS.

FORMAT:

*TCMULT tcmult(i), i = 1, nreg

DEFINITIONS:

tcmult
: Critical temperature multiplier. This multiplier is used to tune the Li mixing rule for pseudo-critical temperature to modify the point at which a single hydrocarbon phase is identified as an oil or a gas.

nreg
: The number of initialization regions into which the reservoir has been divided.

DEFAULTS:

Optional keyword. If *PHASEID *TCMIX or *TRLIMS are not used, *TCMULT has no effect. If these keywords are used, and TCMULT has not been specified by the user, tcmult may be calculated by GEM as explained below.

CONDITIONS:

If it appears, this keyword must be in the INITIAL CONDITIONS keyword group. If it appears, and if *NGREGIONS also appears, then this keyword must follow *NREGIONS.

EXPLANATION:

Use of the Li pseudo-critical temperature mixing rule for single hydrocarbon phase identification is explained under *PHASEID, and for single phase relative permeability interpolation under *TRLIMS. The mixing rule is given by:

$$T_{c,mix} = \frac{\sum_i z_i v_{c,i} T_{c,i}}{\sum_i z_i v_{c,i}}$$

TCMULT allows input of a factor which multiplies this calculated critical temperature to modify the point at which a single hydrocarbon phase is identified as an oil or a gas. For example, increasing $T_{c,mix}$ by entering tcmult greater than one will make it more likely that a single phase block will be identified as an oil.

If TCMULT is not entered, the default multiplier is one. GEM will calculate a multiplier for each initialization region if necessary to ensure that phase identification from the Li mixing rule is consistent with the specification of DGOC or CDEPTH. At the end of initialization a calculation will be performed to determine the maximum reduced temperature ($T_{ro,max}$ = max($T/T_{c,mix}$)) for all blocks in a region identified as having an oil phase composition, and the minimum reduced temperature for all blocks identified as having a gas composition ($T_{rg,min}$).

If *PHASEID *TCMIX is used and *TRLIMS is not, a multiplier will be calculated if $T_{ro,max} > 1$ or $T_{rg,min} < 1$. This ensures that blocks above DGOC or CDEPTH will be labelled as gas on the first timestep, consistent with the specified initialization.

If *TRLIMS is used, interpolation limits troil and trgas are defined. A multiplier will be calculated if $T_{ro,max} >$ troil or $T_{rg,min} <$ trgas. This ensures that blocks above DGOC or CDEPTH will use the gas relative permeability curve directly, without interpolation, and similarly for oil.

If $T_{ro,max} >$ troil and $T_{rg,min} <$ trgas, the multiplier will be set to the midpoint of $T_{ro,max}$ and $T_{rg,min}$.

Example:

```
For nreg = 1:
     *TCMULT  0.8
For nreg = 3:
     *TCMULT  0.93 0.95 0.88
```

Initial Aqueous/Mineral Concentrations *MOLALITY-AQUEOUS, *MOLALITY-AQUEOUS-PRIMARY, *MOLALITY-GASEOUS, *SALINR , *VOLUMEFRACTION-MINERAL

PURPOSE:

These keywords indicate input of initial aqueous or mineral concentrations in the model. In addition to the ususal data for initial conditions, the following information is required.

FORMAT:

*MOLALITY-AQUEOUS
*MOLALITY-AQUEOUS-PRIMARY
*MOLALITY-GASAQU
*SALINR (*PPMWT|*PPMVOL|*MOLAL|*MOLAR|*WTFRAC|*MOLFRAC) salinr
*VOLUMEFRACTION-MINERAL

DEFINITIONS:

*MOLALITY-AQUEOUS

This keyword is used to specify the initial concentrations of the aqueous components (in molality). It must be followed nc_aqu values for the concentrations.

*MOLALITY-AQUEOUS-PRIMARY

This keyword is used to specify the initial concentrations of the primary aqueous components. The secondary aqueous components are calculated internally by solving the chemical-equilibrium equations. It must be followed by nc_aqu – ncheq values, where nc_aqu is the number of aqueous components and ncheq is the number of chemical equilibrium reactions defined with the keyword N-CHEM-EQUIL.

*MOLALITY-GASAQU

This keyword specify the initial gaseous solubility in molality when the keyword *AQFILL is invoked. Without *AQFILL, there is a trace of gas in the aquifer and the gaseous solubility is calculated from the concentration of this trace of gas and this keyword in not required.

*SALINR

This keyword provides an optional input for initial concentration of aqueous component when only one aqueous component viz. NaCl is defined. It may be followed by a unit specifier, then a series of real numbers for the salinity of each initialization region, 1 to nregions. If the unit specifier is omitted the numbers are interpreted as NaCl molality.

*PPMWT

Subkeyword of *SALINR indicating units of salinity will be parts per million by weight (mg NaCl / kg brine).

*PPMVOL

Subkeyword of *SALINR indicating units of salinity will be parts per million by volume (mg NaCl / L brine).

*MOLAL

Subkeyword of *SALINR indicating units of salinity will be molarity (gmol NaCl / kg H2O).

*MOLAR

Subkeyword of *SALINR indicating units of salinity will be molarity (gmol NaCl / L brine).

*WTFRAC

Subkeyword of *SALINR indicating units of salinity will be weight fraction (g NaCl / g brine).

*MOLFRAC

Subkeyword of *SALINR indicating units of salinity will be mole fraction (gmol NaCl / gmol brine).

salinr

Real numbers representing concentration of aqueous component NaCl, one for each initialization region.

*VOLUMEFRACTION-MINERAL

Volume fraction of minerals with respect to bulk volume of rock (nc_mnr values).

DEFAULTS:

The default concentration for aqueous/mineral components is zero.

CONDITIONS:

If these keywords appear, these must be in the INITIAL CONDITIONS keyword group. If these appear, and if *NGREGIONS also appears, then the keyword must follow *NREGIONS.

Note that the *SALINR input is acceptable only when the model has one and only one aqueous component with a component name ‘NaCl’.

EXPLANATION:

*SALINR is an alternate way of specifying initial reservoir salinity compared with the more general *MOLALITY *AQUEOUS keyword and related inputs. It offers the choice of a number of salinity unit options and requires specification of only one aqueous component, ‘NaCl’ in the aqueous phase. Also see related keywords *NC-AQUEOUS, *COMPNAME-AQUEOUS, *SALIN-CALC and *INJ-SALIN.

Numerical Methods Control

Notes on Numerical Methods Control

When running in serial mode, GEM uses incomplete to complete LU factorization (ILU) as a preconditioner to GMRES acceleration to solve the linear systems arising in compositional simulation. The factorization is performed to the degree, max_deg, specified by the keyword *SDEGREE. This provides a general method for obtaining approximate LU factors. As max_deg increases, the approximations become better and better until all terms possible have been included and the LU factors are complete. However this completeness is achieved at the cost of increased computation time and storage.

Setting *MODILU *ON can improve convergence behavior of the iterative matrix solution routine for almost incompressible single phase problems.

Red black ordering (*SORDER *REDBLACK or *SORDER *RCMRB) allows the iterative matrix solution routine to eliminate approximately half the reservoir unknowns, thereby reducing storage requirements for the LU factors. Red-black ordering is the default ordering for GEM.

For most problems a low degree factorization will suffice, but if convergence of the linear solution becomes intolerably slow there are several approaches that can be taken:

a) The north specified in *NORTH can be increased, but this results in increased storage for the GMRES procedure and increased execution time in each iteration.

b) The max_deg can be increased, but this requires more storage for the LU factors and increased iteration costs.

c) It is important to note that there can be a trade-off between a) and b) but either one or both provide mechanisms to deal with problems that the lower degree factorizations have difficulty solving and for which a direct method would be too expensive to use.

GEM also has access to parallel technology using work described in “A Shared-Memory Parallel Black-Oil Simulator with a Parallel ILU Linear Solver”, SPE 79713, (D. A. Collins, J. E. Grabenstetter, and P. H. Sammon, presented at the SPE Reservoir Simulation Symposium held in Houston, Texas, 3–5 February 2003). This solver, the PARASOL solver, uses the same general techniques as AIMSOL, the serial solver, but it takes advantage of special orderings so it can make use of several processors; see the *SOLVER keyword.

For most applications, the default solution method is sufficient. Use the default values for: *UNDERRELAX, *PRECC, *SORDER, *DEGREE, *PIVOT, *MODILU, *ITERMAX.

To control the length of the simulation run, the following keywords may be used:

*MAXCPU, *MAXSTEPS, *DTMIN.

To control the size of the timesteps, use:

*DTMIN, *DTMAX, *NORM, *MAXCHANGE.

For initial trial runs, use *NORM *UNKNOWN. Refer to the EXPLANATION section of *NORM for more information.

To control the convergence criteria, use:

*PRECC, *CONVERGE *ITERMAX.

To control the implicitness of the run, use:

*AIM.

GEM has a numerical dispersion control option that is activated with the *TWOPTFLUX keyword. This option blends usual upstreamed fluxes with higher-order, averaged, two-point fluxes under the control of a Total Variation Diminishing (TVD) flux limiter. The TVD limiter ensures that the overall numerical scheme maintains numerical stability and eliminates over- and under-shoot in the calculations. The limiter allows the higher-order flux calculation to predominate in reservoir regions away from the edges of saturation fronts and fluid banks, and reverts to the more stable upstream scheme near these frontal regions where compositional variable and/or saturations are changing rapidly. Much better resolution of fronts (over a few grid cells) can be seen when using this method.

There are no required or mandatory keywords in this section and there is no critical keyword ordering. The defaults used in the numerical solution techniques provide a robust and efficient solution to most simulation problems. You should override the defaults only if you have a good understanding of the solution methods involved. Inappropriate overriding of the defaults may result in the use of much more CPU time than would otherwise be required for a problem.

Usage in Other Sections

Most of the keywords in this section may be used in the Well and Recurrent Data section: The exceptions are *SOLVER *DPLANES and *JACPAR (cannot change between serial and parallel algorithms during a run), and *TWOPTFLUX (must be used throughout the run).

Numerical Methods Control Identifier (Optional) *NUMERICAL

PURPOSE:

*NUMERICAL identifies the beginning of all numerical methods control keywords.

FORMAT:

*NUMERICAL

DEFAULTS:

Optional keyword. It is required if you wish to override the defaults for one or more of the keywords in this group.

CONDITIONS:

The NUMERICAL METHODS CONTROL keyword group follows the INITIAL CONDITIONS keyword group in the data file. This keyword must be the first keyword in the NUMERICAL METHODS CONTROL keyword group.

EXPLANATION:

The defaults used in the numerical solution techniques provide a robust and efficient solution to most simulation problems. You should override the defaults only if you have a very good understanding of the solution methods involved. Inappropriate overriding of the defaults may result in the use of much more CPU time than would otherwise be required for a problem.

Maximum Timesteps (Optional) *MAXSTEPS

PURPOSE:

*MAXSTEPS is used to specify the maximum number of timesteps for the simulation run.

FORMAT:

*MAXSTEPS *num*

DEFINITIONS:

num
An integer to specify the maximum number of timesteps allowed.

DEFAULTS:

Optional keyword. The default value is 99999; therefore if this keyword is not present in the input data file, the number of timesteps for the simulation is practically unlimited.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group. Note that 'num' corresponds to the number of timesteps to be taken in the current simulation (even if it is a restarted run and timesteps have been taken in previous runs).

EXPLANATION:

The simulation will be terminated when the number of timesteps (in the current run) is equal to the number specified by *MAXSTEPS.

Example:

```
** Limit the number of timesteps to 100
*MAXSTEPS 100
```

Maximum CPU Seconds, Maximum Elapsed Seconds (Optional) *MAXCPU *MAXCLK

PURPOSE:

*MAXCPU is used to specify the maximum CPU seconds for the simulation run.

*MAXCLK is used to specify the maximum elapsed seconds for the simulation run.

FORMAT:

*MAXCPU *maxcpu*
*MAXCLK *maxclk*

DEFINITIONS:

maxcpu
A non-negative number to specify the maximum CPU time in seconds allowed for this simulation run.

maxclk
A non-negative number to specify the maximum elapsed time in seconds allowed for this simulation run.

DEFAULTS:

Optional keywords. If *MAXCPU is not present in the input data file, the default is that CPU time for the simulation is not explicitly limited. If *MAXCLK is not present in the input data file, the default is that elapsed time for the simulation is not explicitly limited.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

Simulation will be terminated when the CPU or elapsed time exceeds the number specified by *MAXCPU or *MAXCLK respectively.

These keywords allow the program to terminate gracefully, having written output and restart records before running out of CPU or elapsed time.

For serial runs CPU and elapsed time are practically the same however for parallel runs the elapsed time may be different from the CPU time. For parallel runs, specification of a maximum elapsed time via *MAXCLK is more meaningful.

Command line options -maxcpu *maxcpu* or -maxclk *maxclk* can be used to have same effect as *MAXCPU or *MAXCLK.

Example:

```
** Limit the cpu time to one hour.
*MAXCPU 3600.0

** Limit the elapsed time to 3o minutes
*MAXCLK  1800.
```

Maximum and Minimum Timestep Size (Optional)

***DTMAX, *DTMIN**

PURPOSE:

*DTMAX identifies the maximum time-step size.

*DTMIN identifies the minimum time-step size.

FORMAT:

*DTMAX *max_time_size*
*DTMIN *min_time_size*

DEFINITIONS:

max_time_size
A real number to specify the maximum time-step size allowed (day | day).

min_time_size
A real number to specify the minimum time-step size allowed (day | day).

DEFAULTS:

Optional keyword. The default value for *max_time_size* is 365 days. The default value for *min_time_size* is 1.0E-05 day.

CONDITIONS:

These keywords may be located in the NUMERICAL METHODS CONTROL keyword group, and may also be part of recurrent data in the WELL AND RECURRENT DATA keyword group.

When restarting a run, if the user wishes to change either value or both values in the restarted run, the keyword(s) must be included in the recurrent data, after the restart time.

EXPLANATION:

If the time-step size calculated by the automatic time-step selector is larger than *max_time_size*, it is set to *max_time_size*.

If the automatic time-step selector cuts the time- step size to smaller than min_time_size specified by *DTMIN, the simulation is stopped.

The time-step size is always automatically adjusted so that a timestep will end exactly at the time specified by a *TIME or *DATE keyword in recurrent data.

Example:

```
** Limit the maximum time-step size to half a day,
** and limit the minimum time-step size to 6 hours
*DTMAX 0.5
*DTMIN 0.25
```

Normal Variation in Variables Per Timestep (Optional)

***NORM, *MAXCHANGE**

PURPOSE:

*NORM identifies the typical changes in the basic variables over a timestep. The timestep size is reduced, if the maximum change in pressure or saturation is greater than that specified by *MAXCHANGE.

FORMAT:

*NORM	(*PRESS)	*dpn*
	(*SATUR)	*dsn*
	(*GMOLAR)	*dzn*
	(*AQUEOUS	*daq* | *OFF)
	(*TEMP)	*dtn*
	(*UNKNOWN)	
*MAXCHANGE	(*PRESS)	*maxpn*
	(*SATUR)	*maxsn*
	(*GMOLAR)	*maxzn*
	(*TEMP)	*maxtn*

DEFINITIONS:

*PRESS

This subkeyword identifies pressure (kPa | psi | kPa).

*SATUR

This subkeyword identifies saturations (fraction, dimensionless).

*GMOLAR

This subkeyword identifies global composition (fraction, dimensionless).

*AQUEOUS

This subkeyword identifies target change in aqueous reactions (dimensionless).

*TEMP

This subkeyword identifies temperature (C | F).

*OFF

This subkeyword allows *NORM *AQUEOUS control to be switched off.

*UNKNOWN

The norms are unknown. Use internal criterion to set the timestep size.

dpn, dsn, dzn, daq, dtn

Real numbers to specify the typical changes. ($0 < daq < 3$)

maxpn, maxsn, maxzn, maxtn

Real numbers to specify the maximum allowable changes

DEFAULTS:

Optional keyword. If *NORM is not present in the data file *UNKNOWN is assumed and the default for *MAXCHANGE are:

maxpn	= 10000 kPa or 1450 psia
maxsn	= 0.5
maxzn	= 0.5
maxtn	= 20 °C or 36 °F

If *NORM *AQUEOUS *daq* is not specified, or if no chemical equilibrium reactions are present in the data set, aqueous norms are not used.

If the norms are entered, the defaults for *MAXCHANGE are 2 times the *NORM.

CONDITIONS:

These keywords must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

*NORM specifies the typical changes in the basic variables during a timestep. The timestep size is adjusted internally such that the average change in pressure, saturation, or global mole fraction during a timestep equals the entered value.

Typical values are:

dpn	=	1000 kPa or 145 psi
dsn	=	0.15
dzn	=	0.15
dtn	=	10 °C or 18 °F

for simple field scale simulations. For slim tube simulations, try decreasing the pressure norm to about 10 psi.

Alternatively, enter the keywords:

*NORM *UNKNOWN

In this case, the timestep size will not be controlled by the norms but by the convergence behavior of the Newtonian iterations. Experience shows that, with the current version of GEM, simulation runs faster with appropriately fine-tuned norms than with *NORM *UNKNOWN. Unfortunately, it takes many trial runs to fine-tune the norms.

Note that omitting the keyword *NORM is the same as entering *NORM *UNKNOWN.

The suggested procedure to control the timesteps is as follows:

1. Run the simulation problem for a few timesteps with *NORM *UNKNOWN. If the convergence behavior is acceptable, continue the run as is and optionally set appropriate *DTMAX. For example, set a maximum of 3 months for field scale simulations, a maximum of 1 month for pattern studies and a maximum of one-tenth the pore volume throughput time for slim-tube simulations.
2. If there are lots of timestep repeats set either the *DTMAX or *NORM as follows.

3. If after the timestep cut the timestep size is acceptable to be used throughout the run, then set the successful timestep size before the repeat as the maximum timestep size. Use the keyword *DTMAX to do this.
4. If after the timestep cut the timestep size is unacceptably small, determine the controlling variable from the output. The controlling variable is the variable, pressure, saturation, or composition, that causes the repeat. This is normally the variable that changes the most during the troublesome timestep.
5. Set the norm of the controlling variable roughly equal to the changes of that variable before the troublesome timestep. Use the keywords *NORM *PRESS, *NORM *SATUR or *NORM *GMOLAR.

   Sample runs using the *NORM *UNKNOWN keywords have been included in the standard release. Also refer to PROBLEM WITH SMALL TIMESTEPS OR LONG EXECUTION TIMES in the Tutorial section for more information.

The introduction of *daq* parameter via *NORM *AQUEOUS keyword to the timestep size selection strategy makes it possible to simulate CO_2 sequestration scenarios involving many years of mineral reactions after several years of CO_2 injection with timesteps automatically becoming large enough in the later part of the simulation that the scenario can be simulated in a reasonably short elapsed computation time. Typically the aqueous part of the timestep selection keeps the timesteps quite short in the early stages of the scenario when CO_2 is injected and allows the timesteps to grow after CO_2 injection ceases and the system moves to equilibrium. The timesteps can be made short enough in this way that the material balance errors in the linear combinations of the aqueous species concentrations invariant to the aqueous equilibrium reactions can be kept acceptably small. Typically values of *daq* of about 0.3 suffice however user is expected to adjust it depending on the acceptable material balance error level and the desired timestep size. Larger values of *daq* will result in larger timesteps during CO_2 injection, but with larger material balance error. Values of *daq* larger than 3 will have no effect on the timestep selection strategy.

The *MAXCHANGE keyword specifies the maximum changes in the basic variables during Newtonian iterations. During the Newtonian iterations, if any change in a variable value exceeds *MAXCHANGE, the timestep size is reduced and the timestep is repeated.

Example:

```
*NORM *PRESS  150.0
*NORM *GMOLAR   0.08
```

Adaptive-Implicit Switching (Optional) *AIM

PURPOSE:

*AIM controls the adaptive-implicit switching option.

FORMAT:

*AIM	(*OFF)	
	(*STAB)	*level*
	(*THRESH)	*dnimp*
	(*STAB *AND-THRESH)	*level dnimp*

DEFINITIONS:

*OFF

Adaptive-implicit option is not used. The problem will be solved with fixed implicitness.

*STAB

Adaptive-implicit option with stability switching criterion is used.

level

An integer number to specify the frequency of checking for stability switching.

level=1: check selected blocks after convergence of each timestep;

level=2: check all non-well blocks after convergence of each timestep;

level=3: check all non-well blocks after each Newtonian iteration but switch a given block at most once per timestep.

For *AIM *STAB *AND-THRESH, *level*=3 is not permitted.

*THRESH

This subkeyword identifies threshold values for adaptive implicit switching.

dnimp

A positive real number. When threshold switching is used an explicit grid block is switched to implicit if changes in water saturations or hydrocarbon component global mole fractions in the block during the current timestep exceed *dnimp* times *dsn* or *dzn*. The checking is done after the timestep has converged.

*STAB *AND-THRESH

This subkeyword invokes a combined stability and threshold based strategy for adaptive implicit switching. The stability switching is done at the timestep level, while threshold switching is done at the timestep or Newton iteration level.

level

The *level* can have value of 1, or 2.

1. Check selected blocks for switching after timestep converged
2. Check all non-well grid blocks for switching after timestep converged.

dnimp

The value of *dnimp* under *AIM *STAB *AND-THRESH, should be a real number having a positive or a negative value. An explicit grid block is switched to implicit if changes in water saturations or hydrocarbon component global mole fractions in the block during the current timestep/iteration exceed abs(*dnimp*) times *dsn* or *dzn*. The checking is done after the timestep has converged when *dnimp* is positive; and, at each Newton iteration when *dnimp* is negative.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default is:

*AIM *STAB 1

The default value for level when *AIM *STAB is entered is 1.

The default value for dnimp when *AIM *THRESH is entered is 1.0.

The combined stability and threshold checking criterion,*AIM *STAB *AND-THRESH, is off by default.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group. No more than one of *STAB or *THRESH may be specified.

EXPLANATION:

This keyword controls the switching of a grid block between an explicit and a fully implicit formulation.

Switching is based on either of the 3 following approaches:

1. Threshold changes in water saturation and hydrocarbon component global mole fractions (*THRESH). When the *THRESH option is used switching is only from explicit to implicit.

   For simulations of dual-porosity or dual-permeability systems, if this switching criterion is used, it is recommended to initialize all fracture blocks to be implicit using the keyword *AIMSET.
2. CFL stability test (*STAB); with this approach switching is done in both directions.
3. Combined CFL stability test (*STAB) and threshold changes in water saturation and hydrocarbon global mole fraction (*AND-THRESH). Unlike option 1 above, the switching is both ways i.e., from explicit to implicit and implicit to explicit. This option is not supported by Builder 2009.10 and is being released as a test (Beta) feature.

Examples:

```
** Threshold values for adaptive implicit switching
*AIM *THRESH 0.7
```

-or-

```
** Stability switching with default level of 1
*AIM *STAB
```

-or-

```
** Stability switching with combined stability and
** threshold checking
*AIM *STAB *AND-THRESH 2 -0.2
```

Convergence of Newton's Method (Optional) *CONVERGE, *CONV-RESONLY

PURPOSE:

*CONVERGE is used to specify the convergence criterion for Newton's Method when solving the non-linear conservation equations in the simulator.

FORMAT:

*CONVERGE	(*PRESS)	*tolp*
	(*HC)	*tolh*
	(*WATER)	*tolw*
	(*TEMP)	*tolt*
	(*MAXRES)	*tolr*
	(*CEQVLE)	*tol_vle*
	(*CEQAQU)	*tol_aqu*
	(*CEQADS)	*tol_ads*
	(*CEQSLD)	*tol_sld*
	(*CEQSDP)	*tol_sdp*
	(*CEQCHE)	*tol_che*
	(*CEQMNR)	*tol_mnr*
	(*CEQIEX)	*tol_iex*
	(*CEQALL)	*tol_ceall*
*CONV-RESONLY	*ON | *OFF	

DEFINITIONS:

*PRESS

Indicates changes convergence tolerance for pressure (kPa | psia).

*HC

Indicates relative changes convergence tolerance for a hydrocarbon component molar density (dimensionless).

*WATER

Indicates relative changes convergence tolerance for water molar density (dimensionless).

*TEMP

Indicates relative changes convergence tolerance for temperature (°C | °F).

*MAXRES

Indicates the maximum scaled residual allowed for any single equation at convergence (dimensionless).

*CEQVLE, *CEQAQU, *CEQADS, *CEQSLD. *CEQSDP, *CEQCHE, *CEQMNR *CEQIEX

Indicates the maximum sum of squared residual allowed for any single constraint equation at convergence (dimensionless). The three letters following *CEQ have the following meaning:

VLE	Vapor-liquid equilibrium
AQU	Gas-aqueous equilibrium
ADS	Gas adsorption
SLD	Solid precipitation (asphaltene)
SDP	Solid deposition (asphaltene)
CHE	Chemical equilibrium
MNR	Mineral equilibrium
IEX	Ion-exchange

*CEQALL

Indicates the maximum sum of squared residual allowed for all of the single constraint equations at convergence (dimensionless).

tolp, tolh, tolw, tolt

Real numbers indicating the changes tolerance value for pressure, hydrocarbon component molar density, water molar density and temperature.

tolr

tolr can be either:

1. A real number indicating the maximum scaled residual allowed for any single governing conservation equation at convergence; or
2. A subkeyword specifying the convergence tolerance. It can be entered as *LOOSER, *LOOSE, *NORMAL, *TIGHT, or *TIGHTER. In this case, the values used for tolr are as follows:

	tolr
*LOOSER	0.01
*LOOSE	0.005
*NORMAL	0.001
*TIGHT	0.0005
*TIGHTER	0.0001

tol_vle, tol_aqu, tol_ads, tol_sld, tol_sdp, tol_che, tol_mnr

Real numbers indicating the sum of square residual tolerance value for individual equations representing vapor-liquid equilibrium, gas-aqueous equilibrium, gas adsorption, solid (asphaltene) precipitation, solid (asphaltene) deposition, chemical equilibrium and mineral equilibrium, respectively.

tol_ceall

Real number indicating the sum of square residual tolerance value for each of equations representing vapor-liquid equilibrium, gas-aqueous equilibrium, gas adsorption, solid (asphaltene) precipitation, solid (asphaltene) deposition, chemical equilibrium and mineral equilibrium.

*CONV-RESONLY

When *ON, indicates that completer convergence of block residuals is a necessity before the Newton iteration is accepted as converged.

DEFAULTS:

Optional keywords. The defaults are:

*CONVERGE	(*PRESS)	3.55 kPa or 0.514 psia
	(*HC)	0.01
	(*WATER)	0.01
	(*TEMP)	0.001 °C or °F
	(*MAXRES)	0.001
	(*CEQ…)	1.E-10
*CONV-RESONLY	*OFF	

CONDITIONS:

These keywords must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

*CONVERGE specifies the criterion for judging the convergence of the Newtonian iterations.

Two different types of convergence criterion are applied: a change criterion, in which relative changes in primary variables are checked, and a residual criterion, in which the residuals of the equations being set iteratively to zero are checked. If either criterion is satisfied, the iteration is judged to have converged.

Example:

```
*CONVERGE  *MAXRES  *TIGHT
*CONVERGE  *PRESS  2.50
*CONV-RESONLY  *ON
```

*CONVERGE *CEQ…, specifies the criterion for judging the convergence of constraint equation residuals. The constraint equation residual checking is activated when number of allowed iterations for checking (specified via *NCHECK-CEQ *nn*) is larger than zero. If the constraint equation residuals do not converge as per specified tolerance, the Newton cycle is repeated. The status of constraint equation residual convergence can be viewed by using keyword *DIARY *CHANGES-UNCONV (specified in INPUT OUTPUT CONTROL section or in WELLS AND RECURRENT DATA section) in the log file, under column *cer*. The sum of square residuals can be monitored via *WPRN *ITER *ALL (specified in INPUT OUTPUT CONTROL section).

The allowed range for *CONVERGE *CEQ… is 1.E-02 to 1.E-12. In absence of keyword *NCHECK-CEQ, the keyword *CONVERGE CEQ… sets *ncheck_ceq* to 3.

In GEM a Newtonian iteration is by default accepted to be converged when either all the variables or all the block residuals have converged to specified (or default in case not specified) tolerance. Keyword *CONV-RESONLY when *ON forces all the block residuals to converge before the iteration is accepted to have converged. In general in certain situations it may improve material balance accuracy due to larger number of Newtonian iterations before convergence, however in some other instances the convergence may be slower or may not be achieved if the residual convergence criterion is too stringent. Use of *CONV-RESONLY may also cause larger runtimes in certain models. It should, therefore be cautiously turned *ON or *OFF.

Maximum Newton Cycles for Constraint Equation Residual Checking (Optional) *NCHECK-CEQ

PURPOSE:

*NCHECK-CEQ is used to specify the maximum number of allowable Newtonian iterations for constraint equation residual checking.

FORMAT:

*NCHECK-CEQ *ncheck_ceq*

DEFINITIONS:

ncheck_ceq
An integer indicating the maximum number of Newtonian iterations for which constraint equation residual checking is to be done.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default maximum number of Newtonian cycles for constraint equation residual checking is 0, i.e., constraint equation residual checking is by default off.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword specifies the maximum number of Newton iterations allowed in a timestep for checking the constraint equation residuals. The constraint equation residual checking is activated by either specifying *CONVERGE *CEQ… (which also sets *ncheck_ceq* = 3) or by specifying a number greater than zero after *NCHECK-CEQ.

Example:

```
*NCHECK-CEQ  5
```

Maximum Newtonian Cycles (Optional) *NEWTONCYC

PURPOSE:

*NEWTONCYC is used to specify the maximum number of allowable Newtonian iterations.

FORMAT:

*NEWTONCYC *maxn*

DEFINITIONS:

maxn
An integer indicating the maximum value.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default maximum number of Newtonian cycles is 10.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword specifies the maximum number of Newton iterations allowed in a timestep for solution of the basic reservoir equations. If a solution within the convergence tolerances is not achieved in maxn or fewer iterations, the simulator reduces the timestep size and repeats the timestep.

Examples:

```
*NEWTONCYC  8
```

Underrelaxation Option (Optional) *UNDERRELAX

PURPOSE:

*UNDERRELAX controls the underrelaxation option.

FORMAT:

*UNDERRELAX *urvalue*

DEFINITIONS:

urvalue
A real number to specify the underrelaxation value.

DEFAULTS:

Optional keyword. The default value is 0.5.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

The underrelaxation parameter is employed by the model after the fifth Newtonian iteration to aid convergence if the program diverges. Variable changes are reduced by the factor of urvalue below the values calculated from the Newtonian iteration.

Example:

```
*UNDERRELAX  0.7
```

Convergence Tolerance for Linear Solver (Optional) *PRECC

PURPOSE:

*PRECC is used to specify the convergence tolerance for the linear equation solver (AIMSOL) internal iteration, which uses the GMRES procedure.

FORMAT:

*PRECC *precc*

DEFINITIONS:

precc
Real number specifying the tolerance.

DEFAULTS:

Optional keyword. The default is 5.0E-04.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

The linear equation solver inner iteration is judged to have converged when the current root mean square (rms) residual is less than precc times the initial rms residual.

Example:

```
*PRECC  1.E-05
```

Orthogonalization (Optional) *NORTH

PURPOSE:

*NORTH specifies the maximum number of GMRES steps to be performed in the iterative part of the linear equation solver before resetting.

FORMAT:

*NORTH *num*

DEFINITIONS:

num
An integer defining the maximum number of GMRES steps before resetting.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default value is: 40.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

The maximum number of GMRES steps defaults to 40. This should be adequate for the majority of cases, but increasing to larger values can help if there are solver failures.

Example:

```
*NORTH  80
```

Solver Equation Ordering (Optional) *SORDER

PURPOSE:

*SORDER controls the ordering of equations in ILU factorization.

FORMAT:

```
*SORDER    (*NATURAL)
           (*REDBLACK)
           (*RCM)
           (*RCMRB)
```

DEFINITIONS:

*NATURAL
Use natural ordering.

*REDBLACK
Use generalized red-black ordering.

*RCM
Use reverse Cuthill-McKee ordering.

*RCMRB
Use reverse Cuthill-McKee/red-black ordering.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default is:

```
*SORDER  *REDBLACK
```

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

The red-black ordering allows the linear solver AIMSOL to eliminate approximately half of the reservoir unknowns, thereby reducing storage requirements for the LU factors.

Example:

```
*SORDER   *REDBLACK
```

Solver Factorization Degree (Optional) *SDEGREE

PURPOSE:

*SDEGREE controls the maximum degree of fill terms used in the factorization.

FORMAT:

*SDEGREE (*max_deg*)
(*GAUSS)

DEFINITIONS:

max_deg
An integer to specify maximum degree of fill terms.

*GAUSS
Keyword specifying that Gaussian elimination be used (no neglect of fill terms).

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default is:

*SDEGREE 1, if REDBLACK or RCMRB ordering is used;
*SDEGREE 2, if NATURAL or RCM ordering is used.

If both *SORDER and *SDEGREE are absent, first degree factorization with red-black ordering will be used.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword controls the maximum degree of fill terms retained in the approximation of the LU factors via incomplete Gaussian elimination, where max_deg is an integer. A value of max_deg greater than the matrix bandwidth or use of the *GAUSS keyword results in complete Gaussian elimination, in which case the exact LU factors are obtained.

In general, Gaussian elimination or values of max_deg larger than 2 are required only for very difficult simulations (e.g. those with extreme permeability contrasts). Larger values of max_deg result in more calculations and a longer simulation run time.

Example:

```
** Use Gaussian elimination
*SDEGREE  *GAUSS
```

Pivot Stabilization (Optional) *PIVOT

PURPOSE:

*PIVOT controls the diagonal submatrix inversion pivot stabilization.

FORMAT:

*PIVOT (*ON)
(*OFF)

DEFINITIONS:

*ON
Pivot stabilization is performed.

*OFF
No pivot stabilization is performed.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default is no pivot stabilization:

*PIVOT *OFF

Specifying *PIVOT without a subkeyword will default to:

*PIVOT *ON

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword selects the pivot stabilization of diagonal submatrix inversion. In certain cases which are rarely encountered in reservoir simulation, not using pivoting may result in an unnecessary run termination due to an attempted division by zero. Use of pivoting increases computation time somewhat.

Example:

```
*PIVOT
```

Modified ILU Factorization (Optional) *MODILU

PURPOSE:

*MODILU is used to select a modified incomplete LU factorization.

FORMAT:

*MODILU (*ON)
(*OFF)

DEFINITIONS:

*ON
Use modified ILU factorization.

*OFF
Do not use modified ILU factorization.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default is not to use a modified ILU factorization:

*MODILU *OFF

Specifying *MODILU without a subkeyword will default to:

*MODILU *ON

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

Modified ILU produces L and U matrices whose product, in favorable cases, more accurately represents the original Jacobian matrix than does the LU product resulting from normal ILU. This is achieved at the cost of some computation time in the factorization but may accelerate the GMRES iteration. In unfavorable cases, however, MILU may actually retard convergence of the GMRES iteration. Almost incompressible single phase problems tend to be favorable cases for modified ILU.

Example:

```
** Use modified ILU factorization.
*MODILU  *ON
```

Maximum Iterations (Optional) *ITERMAX

PURPOSE:

*ITERMAX is used to specify the maximum number of GMRES iterations (inner iterations) allowed in the Jacobian matrix solution routine.

FORMAT:

*ITERMAX *maxn*

DEFINITIONS:

maxn
An integer indicating the maximum value.

DEFAULTS:

Optional keyword. If it is not present in the input data file, the default maximum number of iterations is 80.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL and/or WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword specifies the maximum number of iterations allowed in the solution of the linearized set of equations by the linear equation solver AIMSOL.

The minimum allowable value for maxn is 1. This is recommended with *GAUSS only.

The maximum allowable value for maxn is 200.

Numerical Dispersion Control (Optional) *TWOPTFLUX

PURPOSE:

*TWOPTFLUX activates a numerical dispersion control option.

FORMAT:

*TWOPTFLUX (*IRREGULAR)

DEFINITIONS:

*IRREGULAR
A sub-option to select an alternate form of the TVD flux limiter function.

DEFAULTS:

Optional keyword. The higher-order scheme is activated only when *TWOPTFLUX appears, and the Fromm form of the limiter function is used.

CONDITIONS:

This keyword must be located in the NUMERICAL METHODS CONTROL keyword group.

EXPLANATION:

This option blends usual upstreamed fluxes with higher-order, averaged, two-point fluxes under the control of a Total Variation Limiting (TVD) flux limiter. The TVD limiter ensures that the overall numerical scheme maintains numerical stability and eliminates over- and under-shoot in the calculations. The limiter allows the higher-order flux calculation to predominate in regions away from the edges of saturation fronts and fluid banks, and reverts to the more stable upstream scheme near these frontal regions where compositional variable and/or saturations are changing rapidly. Much better resolution over fronts (over a few grid cells) can be seen.

Two TVD limiter choices are available. The Fromm limiter is the default, while the *IRREGULAR option chooses a Minimally Extended Scheme (MES) limiter. The Fromm limiter works best when the grid has a high degree of regularity; that is, the areal grid spacings are relatively uniform, and equal, in the areal directions, and there are few large variations in cell thicknesses. If these conditions are not met, the *IRREGULAR option should be used. The latter choice provides less dispersion control, but is more stable in the presence of variations in grid properties. Some experimentation with this choice may be required to obtain the best reduction, and the results are also some-what sensitive to the choice of timesteps.

The implementation uses an explicit formulation for the averaged fluxes to reduce numerical work.

Further details can be found in the paper:

Sammon, Peter H., Kurihara, M., Liang, J., "Applying High-Resolution Numerical Schemes in Reservoirs Described by Complex Corner-Point Grids," paper SPE 66344, presented at the 16th SPE Reservoir Simulation Symposium, Houston, Texas, February 11-14, 2001.

AIMSOL/PARASOL (Optional) *SOLVER

PURPOSE:

Choose which solver to use, AIMSOL or PARASOL (In order to use PARASOL, the parallel computing licensing feature must be active).

FORMAT:

*SOLVER (*AIMSOL | *PARASOL)

DEFINITIONS:

*AIMSOL
CMG's non-Parallel Iterative Solver.

*PARASOL
CMG's Parallel Iterative Solver.

DEFAULTS:

If *SOLVER is absent then *AIMSOL is assumed.

EXPLANATION:

This keyword specifies which linear-equation solver to use. *SOLVER *PARASOL is required in order to solve the linear system of equations in parallel. (Another keyword, *DPLANES, is required to build the equations in parallel.) Appropriate licensing is required for all this option.

Number of Parasol Classes (Optional) *PNPROSL

PURPOSE:

Choose the number and scaling of GMRES vector operation classes (a class is defined as a disjoint set of blocks) used in the scaling and GMRES iteration.

FORMAT:

*PNPROSL *nprosl*

DEFINITIONS:

nprosl

The number of parallel processes to target for the scaling and also for the vector operations in the GMRES iteration.

DEFAULTS:

Optional keyword. If *PNPROSL is absent then *nprosl* is equal to the target number of level one classes.

CONDITIONS:

This keyword is used only with *SOLVER *PARASOL.

EXPLANATION:

This keyword controls the number of solver parallel classes used for scaling and for vector operations in the GMRES iteration.

The classes defined by PNPROSL are used in the scaling and the low level parallelization of GMRES vector operations and are different from the classes defined by *PPATTERN. The default of *PNPROSL is recommended

Red-Black Ordering Check for Parasol (Optional) *CHECKRB

PURPOSE:

Choose when to abandon using Red-Black Ordering for a PARASOL class (a class is defined as a disjoint set of blocks).

FORMAT:

*CHECKRB (*ON (*redmin*) | *OFF)

DEFINITIONS:

*CHECKRB *OFF
: Always use Red-Black ordering for a class.

*CHECKRB *ON
: Red-Black ordering is abandoned for a class in which the fraction of red blocks is too small (when the number of red blocks is less than the fraction redmin times the number of black blocks).

redmin
: Fraction which determines when red-black ordering is abandoned (see the explanation under *CHECKRB *ON above). Any positive value not exceeding 1 is valid. If no value for redmin is entered after *ON, redmin defaults to 0.6.

DEFAULTS:

Optional keyword. If *CHECKRB is absent then *OFF is assumed. *redmin* defaults to 0.6.

CONDITIONS:

This keyword is used only with *SOLVER *PARASOL.

EXPLANATION:

It may not always be efficient to perform Red-black ordering in situations where the number of red blocks is small. This keyword allows the user to abandon Red-Black ordering in this situation. This keyword has no effect if a red-black ordering has not been specified or defaulted. When red-black ordering is abandoned because of application of this option, the factorization degree within the class is increased by one. For example, if the original direction was a treat a PARASOL class with red-black system reduction and degree 1 factorization of the reduced system, and *CHECKRB found the number of red blocks to be fewer than redmin times the number of black blocks, then the class would be treated with no system reduction and degree two factorization.

Factorization Degree Within Parasol Classes (Optional)

***PDEGAA**

PURPOSE:

Choose the factorization degree within PARASOL classes (a class is defined as a disjoint set of blocks).

FORMAT:

*PDEGAA *idegaa*

DEFINITIONS:

idegaa
Factorization degree within each class

DEFAULTS:

Optional keyword. If *PDEGAA is absent then *idegaa* is equal to the value of *SDEGREE (see *SDEGREE in the Numerical Methods Section).

CONDITIONS:

This keyword is used only with *SOLVER *PARASOL.

EXPLANATION:

This keyword allows the user to control the degree of factorization used within a class for *SOLVER *PARASOL.

Factorization Degree Between Parasol Classes (Optional)

***PDEGAB**

PURPOSE:

Choose the factorization degree between PARASOL classes (a class is defined as a disjoint set of blocks).

FORMAT:

*PDEGAB *idegab*

DEFINITIONS:

idegab

Factorization degree between classes.

DEFAULTS:

Optional keyword. If *PDEGAB is absent then *idegab* is *idegaa*+1 (see *PDEGAA).

CONDITIONS:

This keyword is used only with *SOLVER *PARASOL.

EXPLANATION:

This keyword allows the user to control the degree of factorization used between classes for *SOLVER *PARASOL. PARASOL allows red-black elimination only within a class; thus when 1st degree red-black ordering is used it is important that factorization of at least degree 2 be used between classes to raise the quality of the factorization above that obtained with the 1st degree natural ordering.

Parasol Class Partitioning Pattern (Optional) *PPATTERN

PURPOSE:

*PPATTERN sets the basic partitioning of the reservoir into non-connected regions and separators that makes possible the parallelization of the linear solution.

FORMAT:

*PPATTERN *AUTOPSLAB *inum*
or
*PPATTERN *AUTOP2D *I *n* / *J *m* / *K *l*
or
*PPATTERN *ipatrn*
or
*PPATTERN *PARTITION
{ *partition_table* }
or
*PPATTERN *PPARTITION
{ *p_partition_table* }
or
*PPATTERN *GPARTITION
{ *p_partition_table* }
or
*PPATTERN *APARTITION
{ *p_partition_table* }

DEFINITIONS:

*AUTOPSLAB *inum*

inum is the target number of level-1 classes. There are *inum*-1 separating plane classes, and 0 or 1 class containing demotions. The direction taken is one with the maximum number of planes that do not cut the dominant transmissibility direction. Thus *inum* corresponds to the target number of processors desired. The partitioning the reservoir under this method takes care of the presence of null blocks, if any, thus distributing the active blocks as equally as possible amongst the domains. This is the recommended way of specifying the class partitioning.

*AUTOP2D

Automatic parallel partitioning in 2 dimensions (any two directions in 3D reservoir model). Following *AUTOP2D, two in *I *n* / *J *m* / *K *l* specifies which two directions will be partitioned, into *n*, *m*, or *l* slabs, while their positions in the statement will give the order of directions to be partitioned. I, J, K represents the 1st, 2nd and 3rd coordinate in the 3D reservoir grid.

*PPATTERN *AUTOP2D *I 2 *J 2 specifies that it divides the reservoir grid in I direction (1st coordinate) into 2 slots first, then divides each slot into 2 slabs in J direction (2nd coordinate).

The requested 2D partitioning can internally degrade to 1D partitioning (AUTOPSLAB), if the requested partitioning in 2D cannot be satisfied while respecting another internal criterion: Minimum number planes of active cells, which is currently fixed to a default value of 3. The divided domains of level-1 classes have to meet this min number of active planes in both cutting directions.

Under this criterion of min number of active planes and workload balancing of number of active blocks among level-1 classes, the user- requested 2D partitioning of *m*X*n* can lead to different partitioning schemes in effect. One possible scheme is regular cutting, consisting of *m* slots of *n* domains of level-1 classes in each of the slots. The other scheme can be l slots containing varying number of domains of level-1 classes in the slots, depending on the internal automatic optimization procedure, while respecting the two criteria.

If the variation of workload among domains of level-1 classes is greater than certain value, with a current default of 20%, the second scheme will be applied to try to minimize the workload variation. The total number of level-1 classes will be less than or equal to the requested multiplication of *m*X*n*. In order to respect the min number of active planes between separator classes, there can be level-1 classes in effect than the requested total number of level-1 classes. The workload balancing optimizes the workload among the domains of level-1 classes through relocating domains among slots, shifting positions of separator planes etc.

Warning for a fundamental grid consisting of a cylindrical system (*GRID *RADIAL): the code may generate uncertain errors in current version of 2D partitioning if the angular direction is chosen for partitioning.

A related command line option '-autop2d' is available – please see under explanation below.

ipatrn

ipatrn can have the values 0, 1, 2, 3, 4, 5, 6,7, 8 or 9. Figure 1 provides a geometrical representation of different classes under *ipatrn* 1 through 7. Table 1 summarizes the class distribution in the *ipatrn*s. Column 5 of the table shows the number of level-1 classes under each *ipatrn*, corresponding to the target number of threads desired.

Note: Unlike *AUTOPSLAB above, the specification under *PPATTERN using *ipatrn* does not adjust the partitioning automatically due to presence of null blocks. The user is expected to select a particular *ipatrn* based on the reservoir geometry, dominant flow direction, and distribution of null blocks in the reservoir.

partion_table

One to 64 table rows, each starting on a new line and each with the following structure. The first four columns are quoted character strings.

column 1: '*class partitioned*'

column 2: '*1st major new class*'

column 3: '*2nd major new class*'

column 4: '*separator class*'

column 5: (*I | *J | *K)

column 6: *ind*

Each row directs the partitioning of the first class into two major and one separator classes, with the original class no longer existing after the partition. The partitioning is planar, with the separator plane normal to the I, J, or K axis as specified, with index value *ind*. Initially there is the one class 'FIELD'; each row creates three new classes and destroys one, for a net gain of two classes per row. The names serve only to identify the classes while the pattern is being created; they are not referred to thereafter.

p_partion_table

One to 64 table rows, each starting on a new line and each with the following structure. The first four columns are quoted character strings.

column 1: '*class partitioned*'

column 2: '*1st major new class*'

column 3: '*2nd major new class*'

column 4: '*separator class*'

Like *partition_table*, except that the simulator uses an algorithm to decide what direction plane to use and where to put it.

*PPARTITION

This partitioning method equalizes class sizes as much as possible and minimizes the size of the separator class.

*GPARTITION

Use Alan George's rooted-level-structure partition method, which is like *PPARTITION but doesn't use planes.

*APARTITION

Use the "agglomeration partition" method, which is like *GPARTITION but provides classes somewhat more nearly equal in size but somewhat less regular in shape.

DEFAULTS:

Optional keyword. If *PPATTERN is absent then *AUTOPSLAB 2 is assumed.

CONDITIONS:

This keyword is used only with *SOLVER *PARASOL.

EXPLANATION:

The parallelization of the solver requires the partitioning of the reservoir into disjoint sets of blocks (classes). The classes are further organized into levels.

There must be no flow between blocks which are in different classes at the same level.

Example:

Consider a 101 x 50 x 10 reservoir and consider partitioning it into 3 classes as follows:

Class 1:	I = 1:50	J = 1:50	K = 1:10;
Class 2:	I = 52:101	J = 1:50	K = 1:10;
Class 3:	I = 51	J = 1:50	K = 1:10.

The large classes, 1 and 2, have no direct flow interactions because all flow between them must go through blocks in class3. Classes 1 and 2 are at Level 1; class 3 is at level 2. Each of the three following data fragments can be used to specify this class partitioning.

```
** Method 1
*PPATTERN *AUTOPSLAB 2

** Method 2
*PPATTERN 2

** Method 3
*PPATTERN *PARTITION
   'FIELD' 'Class 1' 'Class 2' 'Class 3' *I 51
```

ipatrn	Description	Geometrical Representation
0	Single class (like AIMSOL)	
1	Two crossed vertical planes	
2	Single vertical plane cutting the reservoir into two	
3	Three parallel vertical separating planes	
4	Seven parallel vertical separating planes	
5	Four parallel vertical separating planes	
6	Like *ipatrn* 2, but separating plane is horizontal	
7	Like *ipatrn* 2, but with 2 vertical separating planes	

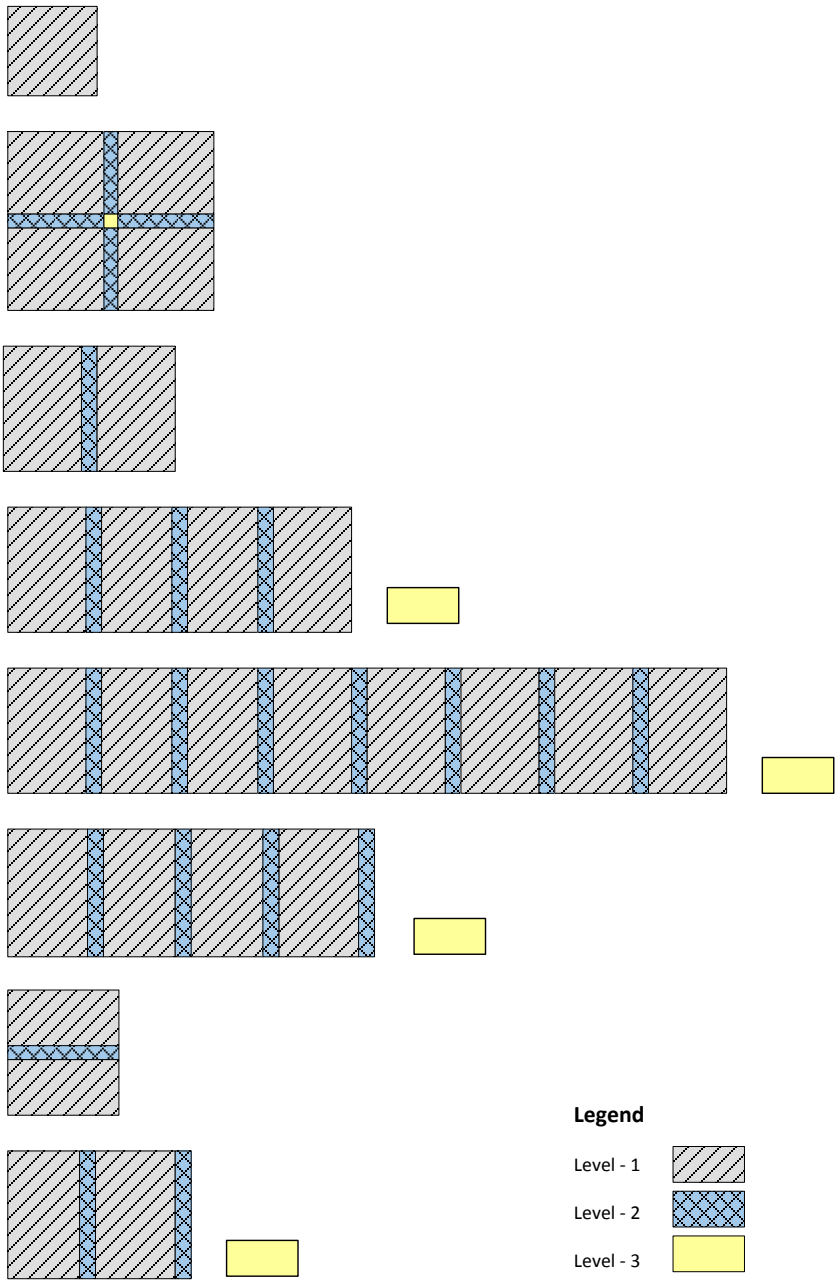


Figure 1: Geometrical representation of Class distribution under different ipatrns

Table 1: Summary of number of levels and classes under different *ipatrn*s

ipatrn	Description	Total levels	Total classes	Class Distribution Level 1	Level 2	Level 3	Remarks
0	Single Class (like AIMSOL)	1	1	1	0	0	
1	Two crossed vertical planes	3	9	4	4	1	
2	Single vertical plane cutting the reservoir into two	2	3	2	1	0	
3	Three parallel vertical separating planes	3	8	4	3	1	Level-3 contains demotions only
4	Seven parallel vertical separating planes	3	16	8	7	1	Level-3 contains demotions only
5	Four parallel vertical separating planes	3	9	4	4	1	Level-3 contains demotions only
6	Like *ipatrn* 2, but separating plane is horizontal	2	3	2	1	0	
7	Like *ipatrn* 2, but with 2 vertical separating planes	3	5	2	2	1	Level-3 contains demotions only
8	Fifteen parallel vertical separating planes	3	32	16	15	1	Level-3 contains demotions only
9	Thirty-one parallel vertical separating planes	3	64	32	31	1	Level-3 contains demotions only

Command line option -autop2d: Automatic parallel partitioning in two-dimensions can also be accomplished via -autop2d pn_1qn_2 where p and q could be any of i, j, or k, with p≠q; and n_1 and n_2 are number of partitions desired in p and q directions. Same rules apply as the ones described under *AUTP2D above. Note that no blanks are allowed in the specification string pn_1qn_2, and in case of an error, the command line input -autop2d is ignored without any message. An example of correct specification is -autop2d i8j2 that will specify 8 partitions in I direction and 2 in J direction.

Cut Plane Direction Override: In automatic partitioning (*PPATTERN *AUTOPSLAB), the partitioning of solver domains is done with an internal logic. Normally if sufficient gridblocks exist the chosen direction is perpendicular to the highest direction of transmissibility. In many situations this may not be desirable. The selected direction in those cases can be specified by the user with command line option -pslabdir i | j | k, where i, j, or k indicate the desired direction of partitioning.

Parallel Equation and Residual Building (Optional) *DPLANES

PURPOSE:

Choose the parallel equation and residual building using a red-black ordering method of domain decomposition. (In order to use *DPLANES, the parallel computing licensing feature must be active).

FORMAT:

*DPLANES *ipldom*

DEFINITIONS:

ipldom
Target number of planes per domain.

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword should not be used in the presence of *JACPAR *ON (-jacpar).

EXPLANATION:

This keyword invokes parallel equation and residual building in GEM, where the equations are built in a different order. For the STARS and IMEX simulators, this new ordering is affected by choices made for planes of cells, and these planes are chosen in the grid direction with the largest number of non-trivial planes where *ipldom* is the number of corresponding non-trivial planes in this direction per domain. The GEM simulator under this option builds equations slightly differently and the specification of number of these planes is not required. Rather, a certain type of red-black ordering is used.

For data input consistency, the GEM simulator accepts the appearance of the *DPLANES keyword as an indication that parallel equation and residual building is desired. The value *ipldom* need not be specified, but if it is, it should be a positive integer.

Either *DPLANES (-doms) or *JACPAR *ON (-jacpar) is required to build the equations and residuals in parallel. Both should not be used together.

Some related command line options are available:

Command-line argument	Equivalent to keywords
-doms	*DPLANES
-lb (0 | 1)	*JLOADBAL (*UNIFORM | *IMPLICIT)
-jacpar	*JACPAR *ON
-parasol *n*	*SOLVER *PARASOL
	*PPATTERN *AUTOSLAB *n*
	*PNPROSL *n*
	*PNTHRDS *m*

-pnthrds *n*	*PNTHRDS *n* where *n* is an integer greater than 0, representing the desired target number of threads and *m* is the minimum of *n* and the number of logical CPUs
-autop2d pn_1qn_2	*PPATTERN *AUTOP2D *pn_1 *pn_2 where p,q = i,j,or k; p≠q and n_1 and n_2 are number of partitions desired in p and q directions.

Load-Balancing in Parallel Equation and Residual Building (Optional) *JLOADBAL, *JLOAD-CHIMP

PURPOSE:

Choose load-balancing option for the parallel equation and residual building using *DPLANES (-doms).

FORMAT:

*JLOADBAL (*UNIFORM | *IMPLICIT)

*JLOAD-CHIMP *delimp*

DEFINITIONS:

*UNIFORM
: Assign grid-blocks uniformly assuming that each grid block has same computational work-load.

*IMPLICIT
: Assign implicit grid-blocks first followed by impes grid-blocks so that work-intensive grid-block are processed first followed by the impes grid-blocks.

delimp
: Percentage change in implicitness to trigger parallel pointer recalculation.

DEFAULTS:

Optional keyword. If *DPLANES (-doms) is used, *JLOADBAL *UNIFORM is the default. If *JLOAD-CHIMP is absent, *delimp* = 0, i.e. parallel pointer recalculation will be done even with a minimal change in implicitness pattern.

CONDITIONS:

These keywords may appear in Numerical Methods Control as well as in Wells and Recurrent Data sections. They should not be used in the presence of *JACPAR *ON (-jacpar).

EXPLANATION:

This keyword invokes load-balancing for parallel equation and residual building in GEM, where the equations are built in a order discussed under *DPLANES keyword. Normally there is heavier workload in implicit grid blocks compared to the one in impes grid blocks. Hence processing implicit blocks ahead of the impes blocks can balance the workload amongst the operating threads. It is normally useful to choose the load-balancing option *IMPLICIT when the implicitness in the model is changing considerably over the run. However, the benefit of processing the implicit blocks ahead of the impes blocks in some models may be offset by the additional pointer regeneration work that is needed for *JLOADBAL *IMPLICIT.

For fully implicit models or models with very small amount of implicitness use of *JLOADBAL *UNIFORM is recommended which is also the default. The option can also be invoked by command line input, -*lb* 0 or -*lb* 1, for *UNIFORM and *IMPLICIT respectively. Note that if both command line and keyword input in the data file are used, the latter will get precedence.

The parallel pointer recalculation could be expensive for models with a large number of gridblocks. There is a trade-off between time saving due to load-balancing and pointer recalculation. *JLOAD-CHIMP helps user control pointer recalculation by specifying a percentage change in implicitness.

Example:

```
** Recalculate parallel pointers when percentage change in
** implicitness exceeds 1.5%
*JLOAD-CHIMP   1.5
```

Some related command line options are available:

Command-line argument	**Equivalent to keywords**
-doms	*DPLANES
-lb (0 | 1)	*JLOADBAL (*UNIFORM | *IMPLICIT)
-jacpar	*JACPAR *ON
-parasol *n*	*SOLVER *PARASOL *PPATTERN *AUTOSLAB *n* *PNPROSL *n* *PNTHRDS *m*
-pnthrds *n*	*PNTHRDS *n* where *n* is an integer greater than 0, representing the desired target number of threads and *m* is the minimum of *n* and the number of logical CPUs
-autop2d pn_1qn_2	*PPATTERN *AUTOP2D *pn_1 *pn_2 where p,q = i,i,or k; p≠q and n_1 and n_2 are number of partitions desired in p and q directions.

Parallel Jacobian and Residual Building based on Parasol Classes (Optional) *JACPAR

PURPOSE:

Choose the parallel equation and residual building using a method of domain decomposition based on Parasol classes. (In order to use *JACPAR, the parallel computing licensing feature must be active).

FORMAT:

*JACPAR *ON|*OFF

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword should not be used in presence of *DPLANES (-doms).

EXPLANATION:

This keyword allows the user to control parallel Jacobian building when also using *SOLVER *PARASOL. *SOLVER *PARASOL is required in order to solve the linear system of equations in parallel. With this option, the Jacobian domains are based on the Parasol classes. *JACPAR or *DPLANES is required to solve the Jacobian building in parallel.

If the number of threads, *PNTHRDS, is not specified, then the number of threads is set to minimum of number of domains n, and the number of logical CPUs m.

Some related command line options are available:

Command-line argument	Equivalent to keywords
-doms	*DPLANES
-lb (0 | 1)	*JLOADBAL (*UNIFORM | *IMPLICIT)
-jacpar	*JACPAR *ON
-parasol *n*	*SOLVER *PARASOL *PPATTERN *AUTOSLAB *n* *PNPROSL *n* *PNTHRDS *m*
-pnthrds *n*	*PNTHRDS *n* where *n* is an integer greater than 0, representing the desired target number of threads and *m* is the minimum of *n* and the number of logical CPUs
-autop2d pn_1qn_2	*PPATTERN *AUTOP2D *pn_1 *pn_2 where p,q = i,j,or k; p $\neq$ q and n_1 and n_2 are number of partitions desired in p and q directions.

To use n logical CPU's where n is greater than or equal two, on a machine with two or more logical CPU's, the command line options:

"-jacpar –parasol n"

may be used.

Either *DPLANES (-doms) or *JACPAR *ON (-jacpar) is required to build the equations and residuals in parallel. Both should not be used together.

Number of Threads (Optional) *PNTHRDS

PURPOSE:

Choose the number of threads to be used for the simulation.

FORMAT:

*PNTHRDS *ithrds*

DEFINITIONS:

ithrds
Number of threads used.

DEFAULTS:

Optional keyword. If parallel Jacobian building and PARASOL are not specified, then the *PNTHRDS default is 1; otherwise, the default is the minimum of 2 and the number of processors in the current machine.

CONDITIONS:

This keyword if present should exist only in the Numerical Methods Control section. It is not allowed in any other section.

EXPLANATION:

If *PNTHRDS is set to a number greater than the number of processors, performance will degrade. If *PNTHRDS is set to greater than two, then the solver *PPATTERN should be changed in order to load balance properly, otherwise poor performance is likely to occur.

Some related command line options are available:

Command-line argument	**Equivalent to keywords**
-doms	*DPLANES
-lb (0 | 1)	*JLOADBAL (*UNIFORM | *IMPLICIT)
-jacpar	*JACPAR *ON
-parasol *n*	*SOLVER *PARASOL *PPATTERN *AUTOSLAB *n* *PNPROSL *n* *PNTHRDS *m*
-pnthrds *n*	*PNTHRDS *n* where *n* is an integer greater than 0, representing the desired target number of threads and *m* is the minimum of *n* and the number of logical CPUs
-autop2d pn_1qn_2	*PPATTERN *AUTOP2D *pn_1 *pn_2 where p,q = i,j,or k; p $\neq$ q and n_1 and n_2 are number of partitions desired in p and q directions.

Number of Threads for Parallel Grid-Processing (Optional)

***GRTHRDS**

PURPOSE:

Choose the number of threads to be used for parallel processing of grid-tasks.

FORMAT:

*GRTHRDS *ngrthrds*

DEFINITIONS:

ngrthrds
Number of threads used for parallel grid-processing.

DEFAULTS:

Optional keyword. If *GRTHRDS is not specified, the number of threads for parallel grid-processing are inherited from other keywords (command-line options) used for setting threads for Jacobian building and/or for parallel linear solver.

CONDITIONS:

This keyword if present should exist only in the Numerical Methods Control section. It is not allowed in any other section.

EXPLANATION:

Certain tasks like connection building in corner-point grid models can be performed in parallel for which *GRTHRDS allows the number of threads to be specified. Normally use of this keyword is not necessary when the simulator is run in parallel processing mode for Jacobian building and/or for parallel solution of linearized equations via PARASOL, in which case the number of threads for grid-processing are the same as set by other keywords (command-line input).

Specifying *ngrthrds* larger than the number of available CPUs will result is setting *ngrthrds* equal to number of CPUs. Setting *ngrthrds* = 1 (via *GRTHRDS 1) will ensure that grid-processing tasks are done in serial mode.

Related command line option is: -grthrds *ngrthrds*

Number of Threads for Parallel Geomechanics-Processing (Optional) *PGTHRDS

PURPOSE:

Choose the number of threads to be used for parallel processing of geomechanics-tasks.

FORMAT:

*PGTHRDS *npgthrds*

DEFINITIONS:

npgthrds
Number of threads used for parallel geomechanics-processing.

DEFAULTS:

Optional keyword. If *PGGTHRDS is not specified, the number of threads for parallel geomechanics-processing are inherited from other keywords (command-line options) used for setting threads for Jacobian building and/or for parallel linear solver.

CONDITIONS:

This keyword if present should exist only in the Numerical Methods Control section. It is not allowed in any other section.

EXPLANATION:

Certain tasks like connection building in corner-point grid models can be performed in parallel for which *PGTHRDS allows the number of threads to be specified. Normally use of this keyword is not necessary when the simulator is run in parallel processing mode for Jacobian building and/or for parallel solution of linearized equations via PARASOL, in which case the number of threads for geomechanics-processing are the same as set by other keywords (command-line input).

Specifying *npgthrds* larger than the number of available CPUs will result is setting *npgthrds* equal to number of CPUs. Setting *npgthrds* = 1 (via *PGTHRDS 1) will ensure that grid-processing tasks are done in serial mode.

Related command line option is: -pgthrds *npgthrds*

Setting Schedule for OPENMP Parallelization (Optional)

***OPENMP_SCHEDULE**

PURPOSE:

Set the schedule of worksharing loops and the chunk size. Use *OPENP_SCHEDULE if you want to set the schedule type separately from the OMP_SHEDULE environment variable

FORMAT:

*OPENMP_SCHEDULE [*STATIC | *DYNAMIC | *GUIDED | *AUTO] ichunk

DEFINITIONS:

ichunk

An integer indicating the chunk-size.

DEFAULTS:

Optional keyword. If *OPEN_SCHEDULE is absent then schedule is defaulted to *DYNAMIC in GEM. The chunk-size *ichunk*, if absent, is set equal to 1 by default for all the schedules other than STATIC and AUTO. See explanation below.

CONDITIONS:

This keyword if present should exist only in the Numerical Methods Control section. It is not allowed in any other section.

EXPLANATION:

The **schedule** specifies how iterations of the associated loops are divided into contiguous non-empty subsets, called chunks, and how these chunks are distributed amongst threads. GEM complies with OPENMP Application Program Interface Version 3.0. Following description is based on that interface.

STATIC

Iterations are divided into chunks of size *ichunk* and the chunks are assigned to the threads in the team in a round robin fashion in the order of the thread number.

When no *ichunk* is specified, the iteration space is divided into chunks that are approximately equal in size, and at most one chunk is distributed to each thread. Note that the size in this case is unspecified or, specified as a value less than 1.

DYNAMIC

The iterations are distributed to threads in the team in chunks as the threads request them. Each thread executes a chunk of iterations then requests another chunk, until no chunks remain to be distributed. Each chunk contains *ichunk* iterations, except for the last chunk to be distributed, which may have fewer than *ichunk* iterations.

GUIDED

The iterations are assigned to threads in the team in chunks as the executing threads request them. Each thread executes a chunk of iterations then requests another chunk until no chunks remain to be assigned.

For *ichunk* = 1, the size of each chunk is proportional to the number of unassigned iterations divided by the number of threads in the team, decreasing to 1. For *ichunk* = *k* (*k*>1), the size of each chunk is determined in the same way, with the restriction that the chunks do not contain fewer than *k* iterations (except for the last chunk to be assigned, which may have fewer than *k* iterations).

AUTO

In this case the decision regarding scheduling is delegated to the compiler and/or operating system. The user gives the system freedom to choose any possible mapping of iterations to threads in the team.

Example 1:

```
*NUMERICAL
  *OPENMP_SCHEDULE *STATIC 1 ** Specify schedule type and
                             ** chunk size
  …
  …
```

Example 2:

```
*NUMERICAL
  *OPENMP_SCHEDULE *STATIC   ** Specify schedule type but use
                             ** default chunk setting

  …
  …
```

Setting OPENMP SCHEDULE via Command-line

GEM also has inbuilt command line switches to alternatively set the openmp schedule. These switches are activated as

-omp_sch_xxx *n*

where xxx could be any of static, dynamic, guided, or auto. The trailing integer *n* specifies the desired chunk size and is optional, in which case the defaults are applied as under:

STATIC without any chunk size (which has a different behavior than STATIC,1 as discussed under the *OPENMP_SCHEDULE keyword.)

DYNAMIC,1

GUIDED,1

AUTO

Example 3:

```
D:\test> gm201110.exe –f gmflu001.dat –doms –parasol 4 –omp_sch_guided 2
```

Setting OPENMP SCHEDULE via Environment Variable

The Openmp schedule can also be set by environment variable OMP_SHEDULE. The syntax is as under on windows and linux/unix (korn shell assumed) platforms.

Windows

set OMP_SCHEDULE=xxx(, *n*)

Linux/Unix

export OMP_SCHEDULE=xxx(, *n*)

where xxx is to be replaced by one of STATIC, DYNAMIC, GUIDED, or AUTO (in capital letters). Chunk-size *n*, if specified should be a non-zero positive number. In case of error, in the specification GEM's internal defaults are enforced without any warning.

Example 4:

```
p:5503/net/rdnas/test> export OMP_SCHDULE=STATIC,1
p:5503/net/rdnas/test> nohup gm201110.exe –f xyz.dat –log &
  …
```

Overides in case of multiple specifications

Note that keyword entry in the GEM input file gets precedence over other two modes of input. The command-line input gets precedence over the OMP_SHEDULE variable specification.

Normally, for GEM runs, setting of OPENP SCHEDULE is not necessary since it is managed internally; i.e., the default setting of *DYNAMIC 1 is changed internally to a different setting depending on the problem type.

The relevant environment settings as *applied* are printed to the output file just before time-stepping for models using parallel processing with more than four threads. Few selected environment variable settings are also printed to the output and log files at the end of the run.

Reference:

http://www.openmp.org OpenMP Application Program Interface, Version 3.0 May 2008.

Automatic Setting of KMP_AFFINITY *AUTO_KMP_AFFINITY

PURPOSE:

Use internal GEM feature to set the KMP Affinity automatically for windows and linux_x64 platforms under certain conditions.

FORMAT:

*AUTO_KMP_AFFINITY [*ON | *OFF]

DEFINITIONS:

*ON
Turn the AUTO_KMP_AFFINITY feature on.

*OFF
Turn the AUTO_KMP_AFFINITY feature off.

DEFAULTS:

Optional keyword. If *AUTO_KMP_AFFINITY is absent then automatic setting of KMP affinity is done only for the runs using all the cores available in the machine, provided KMP_AFFINITY environment variable is not already set.

CONDITIONS:

This keyword if present should exist only in the Numerical Methods Control section. It is not allowed in any other section.

EXPLANATION:

Setting of KMP affinity environment variable to bind threads to processors can improve runtime of GEM models using parallel processing on Intel Xeon 5500, 5600, or 7500 series when the run is made using more than 4 cores on Windows and Linux platforms. Normally this is achieved by setting KMP_AFFINITY environment variable prior to starting the run. GEM echoes all the pertinent environment settings within the output file and provides the same information for selected variables towards the end of primary log as well as the output file.

In case the user has not set the KMP_AFFINITY environment variable, GEM attempts to bind the threads to processors by setting KMP_AFFINITY through internal system calls. This is referred to AUTO_KMP_AFFINITY setting and is applied by default to models using all the cores, i.e. running with same number of threads as the available cores in the machine. It has been seen that for majority of the cases, AUTO_KMP_AFFINITY setting provides similar run times as obtained by setting KMP_AFFINITY environment variable. However, if necessary, *AUTO_KMP_AFFINITY *OFF could be used in the Numerical Control Settings section to disable AUTO_KMP_AFFINITY setting.

While running concurrent multiple jobs with less than available number of cores, it is recommended to use *AUTO_KMP_AFFINITY *OFF, unless the KMP_AFFINITY environment setting is managed outside of GEM, i.e., via a Scheduling software or set manually before running the jobs. This can also be achieved via command-line option

-auto_kmpaff_off. The keywords in data file get precedence over command-line option.

Note that certain operating systems may not respond to either KMP_AFFINITY or its internal setting within GEM. In such a situation one would need to contact the operating system vendor.

While running a single job using all the cores, setting KMP_AFFINITY=compact,0 provides good runtimes, whereas for a single job using less than all available cores or multiple simultaneous jobs using less than all the available cores, setting KMP_AFFINTY=compact,1 may provides good runtimes.

On IBM POWER (AIX, Linux), XLSMPOPTS environment variable needs to be set for binding the threads to the processors. Equivalent of AUTO_KMP_AFFINITY, i.e., internal setting within the GEM code, is not available for these platforms.

Solver Pointer Generation (Optional) *INACTIVE-WELL-SOLVE

PURPOSE:

Choose work-saving method of solver pointer generation.

FORMAT:

*INACTIVE-WELL-SOLVE *ON | *OFF

DEFAULTS:

Optional keyword. By default work saving method of solver-pointer generation is set to *ON.

EXPLANATION:

Gem versions later than 2009.14 employ a work-saving method for solver pointer generation, wherein the frequency of pointer generation calls is reduced by including inactive wells and inactive layers in the solver's linear system. The saving in CPU time can be significant depending on the model size, i.e., number of grid-blocks, number of wells and number of well and layer changes or frequency of constraint switching from rate to BHP and vice-versa, during a simulation run. In general, models with a large number of well changes, that are typically seen in large history-match runs will benefit from this method.

The inactive-well-solve method of solver pointer generation can be optionally switched off, by specifying *INACTIVE-WELL-SOLVE *OFF, if required, for instance for models with a small number of wells that do not require frequent well and layer change or switching of the constraints from rate to BHP and vice-versa.

Generally use of this keyword is seldom required.

Constraint Equation Residual Reduction (Optional)

***CER-REDUCTION**

PURPOSE:

*CER-REDUCTION is used to turn the constraint equation residual reduction method ON or OFF. This method allows the constraint (non-flash) equations to be solved to a specified convergence tolerance by using an inner iteration method.

FORMAT:

*CER-REDUCTION *ON | *OFF

DEFAULT:

If keyword *CER-REDUCTION is absent the constraint equation residual reduction is OFF by default. Note that the method could also be turned on by presence of *ITERCER n keyword, with n entered as a non-zero positive integer, even if *CER-REDUCTION is not specified.

The presence of *CER-REDUCTION *ON by default sets *ITERCER 3, which could be set to any other desired value using keyword *ITERCER (see details for this keyword under Numerical Methods Control section).

CONDITIONS:

If present the keyword may appear in Numerical Methods Control section or in Wells and Recurrent Data section.

EXPLANATION:

*CER-REDUCTION switches on/off the non-flash equation residual reduction method by performing inner (Newton) iterations to reduce the residuals to specified tolerance. Note that these iterations are separate from the Newton cycles performed during the timestep convergence. For solubility models, OGW flash must be turned on for the method to be applicable. Similarly, the method is not applicable for models using *FLASH-METHOD *INNER. In both the above cases, the method will be turned off with a warning message. The convergence tolerance is specified via keyword *CONVERGE *CEQXXX, where XXX stands for AQU, ADS, CHE, …, etc, as discussed under keyword *CONVERGE.

Example:

```
*NUMERICAL
*CER-REDUCTION *ON**    ** turns on CER reduction and sets
                        ** number of CER iterations to 3
…
…
*RUN
*DATE 2000 1 1
…
*DATE 2010 1 1
*CER-REDUCTION *OFF     ** turns off the CER reduction.
…
```

Iterations for Constraint Equation Residual Reduction (Optional) *ITERCER

PURPOSE:

*ITERCER is used to specify the number of allowed iterations for constraint equation residual reduction (CER) using inner (Newton) iterations.

FORMAT:

*ITERCER *n*

DEFINITION:

n
A non-zero positive integer, specifying the number of CER iterations.

DEFAULT:

If keyword *ITERCER is absent the number of CER iterations is assumed to be 0, and the method is set to OFF by default. Note that the number of CER iterations is set to 3 with *CER-REDUCTION *ON keyword, even if *ITERCER is not specified.

The presence of *ITERCER by default sets *CER-REDUCTION *ON, which could be set to *OFF by using keyword *CER-REDUCTION *OFF later (see details for this keyword under Numerical Methods Control section).

CONDITIONS:

If present the keyword may appear in Numerical Methods Control section or in Wells and Recurrent Data section.

EXPLANATION:

*ITERCER n sets the number of CER reduction iterations to n, and also switches on/off the non-flash equation residual reduction method by performing inner (Newton) iterations to reduce the residuals to specified tolerance. Note that these iterations are separate from the Newton cycles performed during the timestep convergence. For solubility models, OGW flash must be turned on for the method to be applicable. Similarly, the method is not applicable for models using *FLASH-METHOD *INNER. In both the above cases, the method will be turned off with a warning message.

Example 1:

```
*NUMERICAL
*ITERCER 5    ** sets CER reduction iterations to 5, also
              ** sets *CER-REDUCTION *ON
```

Example 2:

```
*NUMERICAL
*ITERCER 3       ** sets CER reduction iterations to 3.
…
*RUN
*DATE 2010 5 15
…
*DATE  2012 05 15
*ITERCER 0        ** Turn off CER reduction.
```

Cache Alignment (Optional) *CACHE_ALIGN

PURPOSE:

*CACHE_ALIGN enables or disables cache alignment for parallel processing runs.

FORMAT:

*CACHE_ALIGN *ibytes*

DEFINITIONS:

ibytes
If *ibytes* is > 0 then aligns arrays to *ibytes*. If *ibytes* = 0, then disables cache.

DEFAULTS:

Optional keyword. By default cache alignment is off (i.e., *ibytes* = 0). If *CACHE_ALIGN is present without any integer, then *CACHE_ALIGN 64 is assumed for Win32 and Intel 64 (Win_x64 and Linux_x64) platforms and *CACHE_ALIGN 128 is assumed for POWER platforms, for parallel processing runs.

CONDITIONS:

This keyword is used only with keywords which enable parallel processing such as *SOLVER *PARASOL or *DOMS or *JACPAR.

EXPLANATION:

For parallelization options, selected arrays used by multiple threads are padded to cache boundaries so that multiple threads do not attempt to access the same cache line simultaneously. For some parallel loops, if enabled, it may improve performance by avoiding occurrence of false cache sharing.

Example:

```
** Disable cache alignment
     *CACHE_ALIGN 0
```

Fully Implicit Alignment (Optional) *FI_ALIGN

PURPOSE:

*FI_ALIGN is used to enable or disable fully implicit alignment.

FORMAT:

*FI_ALIGN (*ON|*OFF)

DEFINITIONS:

(*ON|*OFF)
Enables (*ON) or disables (*OFF) fully implicit alignment.

DEFAULTS:

Optional keyword, defaults to *OFF.

CONDITIONS:

This keyword is used only with keywords which enable the parallel linear solver, *SOLVER *PARASOL.

EXPLANATION:

Fully implicit alignment pads adaptive implicit based arrays on class boundaries so that as the number of fully implicit blocks changes, the portion of these arrays related to a particular class does not change.

This option may also be disabled or enabled via the command-line option “-fi_align off” or “-fi_align” respectively. If both keyword and command-line option are present, keyword gets precedence over the command line option.

Example:

```
** Disable fully implicit alignment
     *FI_ALIGN *OFF
```

Adaptive Implicit Alignment (Optional) *AI_ALIGN

PURPOSE:

*AI_ALIGN enables or disables adaptive implicit alignment.

FORMAT:

*AI_ALIGN (*ON|*OFF)

DEFINITIONS:

(*ON|*OFF)
Enables (*ON) or disables (*OFF) adaptive implicit alignment.

DEFAULTS:

Optional keyword. *AI_ALIGN *OFF is the default, except for adaptive implicit setting specified via *AIM *STAB *AND-THRESH where *AI_ALIGN *ON is the default.

CONDITIONS:

This keyword is used only with keywords which enable the parallel linear solver, *SOLVER *PARASOL.

EXPLANATION:

Adaptive implicit alignment pads adaptive implicit based arrays on class boundaries so that as the number of fully implicit blocks changes, the portion of these arrays related to a particular class does not change. The boundaries are calculated on estimates according to the current target of maximum number of fully implicit blocks as initialized by the minimum of (the initial fraction of fully implicitness times 1.1 and 1.0). If at a particular time in the simulation there is no longer sufficient room within a class, then the corresponding adaptive implicit arrays are de-allocated, the maximum current fraction of fully implicitness is increased to the minimum of (twice the current value, the current value + 0.2 and 1.0) and then the corresponding adaptive implicit arrays are re-allocated.

This option may also be disabled or enabled via the command-line option “-ai_align off” or “-ai_align” respectively. If both keyword and command-line option are present, keyword gets precedence over the command line option.

Example:

```
** Enable adaptive implicit alignment
     *AI_ALIGN *ON
```

Geomechanics

Summary of Geomechanical Model

Manual for STARS and GEM

This manual chapter applies to both STARS and GEM simulators. Description differences are indicated by paragraphs starting with **STARS:** or **GEM:**. A keyword page that applies to only one simulator will indicate it in the header line (e.g., *PGDILA is available only for STARS).

The displayed units of dimensioned quantities correspond to the unit system (SI, field, etc.) specified by the simulator's *INUNIT data. For example, a quantity with unit "length" appears here as (m | ft | cm); STARS *INUNIT supports all three unit systems but GEM *INUNIT supports only the first two.

Overview

There are two separate model options:

1. Plastic and Nonlinear Elastic Deformation Model
2. Single-Well Boundary Unloading Model

Each of these options is described in detail below.

Required Keywords

This entire section is optional. If either of the two options is chosen, the keywords required for that option are described below.

Plastic and Nonlinear Elastic Deformation Model

The plastic deformation model performs a finite-element elasto-plastic stress analysis of the reservoir formation using a specific set of displacement and traction boundary conditions. The theory of plasticity provides the theoretical description of the relationship between stresses and strains for a material which exhibits an elasto-plastic response. Detail discussion on the theory of plasticity may be found in many textbooks on the subject, for example, Hoffman and Sachs (1953) or Prager (1959).

When a material behaves elastically, its stress-strain properties can be described by two material constants. For example, Young's modulus and Poisson's ratio is a set of such constants. However, the material may exhibit plastic behavior at an increased stress state. In this case, a yield criterion to prescribe the stress state at which plastic flow commences must be included. This is further complicated by the fact that different class of material exhibit different elasto-plastic characteristics. The post yield stress-strain behavior where deformation consists of both elastic and plastic components requires additional relationships to describe plastic flow.

Plastic strain is considered to be irreversible which occurred after the material reaches a yield state at a certain stress level. The yield criteria, Mohr-Coulomb and Drucker-Prager, which are suitable for the description of geologic material, are currently available to prescribe the yielding condition. There is also an isotropic strain-hardening option. This option allows the material to gain strength as it accumulates plastic strains. More information about constitutive laws and yield criteria for geologic material may be found in Desai and Christian (1977).

The behavior of cyclic loading and unloading as a result of cyclical injection and production processes can be modelled. During injection, the stress state at a location may reach a yield condition and begins to accumulate plastic strains. Shear dilatancy is a component of the resulting volumetric dilatation. Upon production, the material may be unloaded, resulting in the stress state dropping-off from the yield surface. During this period, the material may lose some of the reversible elastic strains.

For an elasto-plastic material, a cap model may be added to its yield criteria to avoid the unlimited hydrostatic compression which may occur in the material during production.

For nonlinear elasticity, the rock behavior may be allowed to obey either a hypoelastic constitutive model or a hyperelastic constitutive model. In the hypoelastic constitutive model, the Poisson's ratio is kept constant whereas bulk modulus and shear modulus vary with the mean effective stress. The model also has loading, unloading and neutral loading cases which are distinguished by the work done criteria dW. If dW is positive, the rock is under a loading condition. If dW is negative, the rock is under an unloading condition. When dW equals zero, neutral loading will occur. In the hyperelastic constitutive model, both values of Poisson's ratio and Young's modulus vary with minimum principle stresses. The model also has loading and unloading case which is distinguished by the shear stress criteria. If the current shear stress is greater than the reference shear stress, the loading condition is applied. If the current shear stress is less than the reference shear stress, the unloading condition occurs. The main advantage of these two constitutive models versus the elasto-plastic model is the computing efficiency. However, both nonlinear elastic constitutive models are only good for pre-failure behavior. Unlike the elasto-plastic model, once shear failure occurs, the nonlinear elastic models cannot predict the post failure phenomena.

The above constitutive models and cap models can be used in 2D or 3D problems in either the Cartesian grid, corner-point grid or axisymmetric radial grid. A 3D radial (cylindrical) grid is converted in the geomechanics module into a corner-point grid type with straight lines replacing arcs between block corners. Each block's angle must not exceed 160 degrees, thereby preventing block volumes approaching zero. Also available is a plane-strain (pseudo 3D) approach, whereby 2D computations are performed successively on each y-z (or x-z) plane, based on the assumption that strain on the direction perpendicular to the plane is negligible.

The geomechanics module solves for the force equilibrium of the formation and calculates the volumetric dilatation/compression as a result of both elastic and plastic straining. The pore volume changes may be caused by a combination of compression/tension or shear stresses. These changes in pore volume and the associated changes in transmissibilities are used in the reservoir model for calculating mass and energy balances in the reservoir.

Sign conventions:

> Compressive stress is positive and tensile stress is negative.
>
> Shear stress is positive when its direction follows the coordinate direction.

The option is enabled by the presence of any one of these keywords:

*BCDOMAIN	*GCINCRMT	*MCPREHD	*SOLVERG
*BCOEF	*GCOUPLING	*MCRCSL	*SPECGRAV
*BIOTSCOEF	*GDI	*MCSWINDX	*STIFFCOM1
*CALIB_POR	*GDJ	*MOHRCOUL	*STIFFCOM2
*COHESHARD	*GDK	*MOHRCOUL	*STIFFINIT
*COHESION	*GEODEPTH	*NB	*STIFFTANG
*DFRICANGLE	*GEODOMAIN	*NE	*STRESS
*DISPLACTOL	*GEOGRID	*NINCS	*STRESS3D
*DLOADBC	*GEOMECH	*NITERGEO	*STRESSGRAD
*DLOADBC3D	*GEOM3D	*NLINEAR	*STRESSGRAD3D
*DLOADIJK	*GEORBLOCK	*NODE4	*TDMAX
*DRUCKER	*GEOROCK	*NODE8	*TDMIN
*ECOEF	*GEOTYPE	*NODE9	*TRESCA
*ELASTMOD	*GEXPONENTN	*NTB	*URBCOEF
*EXPN1	*GLOADBC	*NTE	*URECOEF
*EXPN2	*GNULL	*PLOADBC	*UREXPN1
*FORCETOL	*GPATM	*PLOADBC3D	*UREXPN2
*FPVOLM	*GPOLY	*PLSTRAINY	*URNB
*FRANGSOFT	*GPTOLMUL	*POISSRATIO	*URNE
*FRATIO	*GROTATEI	*PRESCBC	*URNTB
*FRICANGLE	*GROTATEJ	*PRESCBC3D	*URNTE
*FRICANGMN	*GROTATEK	*PRINTGEO	*VISFLOWR
*FRICANGMX	*GTRANSLI	*RCONBK	*VISINIT
*GAMMA	*GTRANSLJ	*RCONBT	*VISPARA
*GAUSSPNT	*GTRANSLK	*RCONFT	*VISPOWER
*GCAPMODEL	*GULBULKMOD	*RCONLF	*VISSCHEME
*GCAPLOC	*HARDEN	*RCONRT	*VISSTEP
*GCAPD	*INTPOWER	*RCONTP	*VISTIME
*GCAPW	*MCOEF	*RIGIDNULL	*VONMISES
*GCAPR	*MCCMODEL	*RIGIDTOP	*WRADIUS
*GCAPTEN	*MCETHA	*RPLTABD	*YLDSTRESS
*GCAPMAT	*MCOCR	*RPWTABD	
*GCFACTOR	*MCOMINDX	*SHEARMOD	

Of these keywords some are mandatory, representing the minimum data required in order to use the option (all other data have defaults):

*GEOMECH	Main keyword for coupling
*ELASTMOD	Young's elastic modulus
*POISSRATIO	Poisson's ratio
*STRESS or *STRESS3D	Initial stresses for 2D and 3D, respectively
*GEOM3D	For 3D finite elements
*GEOGRID, *GDI, *GDJ, *GDK	For independent geomechanics grid

Note: For a hyperelastic constitutive model, keywords *ELASTMOD and *POISSRATIO are not mandatory.

Multiple Rock Types

There is a multiple rock-type option which is enabled with the *GEOROCK and *GEOTYPE keywords. If these keywords are absent, it is assumed that there is only one rock type. The following keywords are applied to the current rock type number:

> *ELASTMOD, *POISSRATIO, *YLDSTRESS, *COHESION, *HARDEN, *FRICANGLE, *BIOTSCOEF, *GRTEMTAB, *GRPORTAB, *GRPRESTAB, *GRSTRESTAB

Single-Well Boundary Unloading Model

The boundary unloading model is restricted to an axisymmetric radial grid analysis where the wellbore is located at the axis. The model performs an elasto-plasticity analysis as in the plastic deformation model described above. In addition, it allows the user to specify the external boundary stress to be unloaded as a result of sand movement into the wellbore. This reduction in well boundary compressive stress may lead to tension failure adjacent to the perforations.

This model is enabled by the presence of the keyword *UNLOADSTR. The keyword *WRADIUS is used to define the well radius at which boundary unloading is to take place. In addition to *UNLOADSTR, all required keywords for the plastic deformation model must be present and the other optional keywords may also be used to define parameters for the model.

It should be noted that during the finite-element analysis, the failed finite elements remain in the model at all time. They are not detached from the model even though the failed elements may contain tension fractures and have little or no strengths. Additionally, the effects of the removal of sand grains from the matrix are not taken into account. These may cause the inadequate description of boundary conditions for the finite elements adjacent to failed elements. These aspects are currently being worked on by CMG.

Host Grid versus Independent Grid

The geomechanics model uses a deformable finite-element grid to perform is calculations. There are two options for specifying the origin of the geomechanics grid description.

1. Use **Host Grid** when keyword *GEOGRID is absent. The geomechanics grid is constructed such that it exactly overlays the grid of the host fluid-flow simulator at initial conditions. More precisely, the two grids are collocated which means that corresponding cell sides and vertices occupy the same location in 3D space. As the simulation progresses, the Geomechanics grid is allowed to move but the assumption of one-to-one cell correspondence remains. Not all host grid features are supported by the geomechanics grid; see "Restricted Host Grid Options", below.
2. Use **Independent Grid** specified by *GEOGRID and its associated keywords. Communication between the geomechanics model and its host simulator is accomplished using a mapping algorithm between the geomechanics and host grids. This technique places few restrictions on the host grid options that can be accommodated.

Restricted Host Grid Options

When geomechanics uses the host (fluid-flow) grid, the following host grid options may not be used:

1. Local refined grids (*REFINE) and dynamic gridding (*DYNAGRID).
2. Faulted grids.
3. Grids in which the corners of adjacent blocks do not coincide. This condition can occur for corner-point grids, especially those generated by mapping software. Also, this condition can be found in Cartesian-based grids that are known generally as variable-thickness and variable-depth and require the *VARI sub-option of keyword *GRID in the Reservoir Description data section. If this grid condition does exist in data without *GEOM3D, a fatal error message is issued when geomechanics is initialized. For *GEOM3D data this condition is not detected by the host simulator but is detected in Builder, so such data should be passed through Builder to do this check. A conversion tool is available for *VARI data.

Converting *VARI to Corner-Point

Geomechanics will reject host *GRID *VARI data that has non-coincident adjacent block corners, as described above. Use keyword *CONVERT-TO-CORNER-POINT in the "Reservoir Description" data section to convert such data internally to a corner-point grid that does satisfy this restriction. Each new single corner location is simply the average of the different *VARI corner locations. Volumes and transmissibilities of individual grid blocks will differ from the *VARI grid, but fractional changes should be reasonable for a well-formed *VARI grid and global quantities like total pore volume should be little different. More extreme variable depth and thickness situations may not convert satisfactorily, in which case some manual adjustment of the original *VARI data is recommended. In addition, deliberate modelling of faults cannot be converted with this keyword. In all cases, you can view both grid types in Results using data sets with and without the keyword.

References:

Hoffman, 0. and Sachs, G., "Introduction to the Theory of Plasticity for Engineers," McGraw-Hill, 1953.

Prager, W., "An Introduction to Plasticity," Addison-Wesley, Amsterdam and London, 1959.

Desai, C.S. and Christian, J.T., "Numerical Method in Geotechnical Engineering," Chapter 2 & 3, McGraw-Hill, 1977.

Output of Geomechanics Responses

A large number of possible geomechanics response quantities are available for viewing in CMG's graphical post-processor "Results". These quantities are available in the list associated with *OUTSRF *GRID, under the heading "The following are available only with *GEOMECH". Typical quantities are stresses, strains, displacements and material model parameters like Young's modulus. Each quantity can be used also with the special history types *BLOCKVAR, *MAXVAR, MINVAR and AVGVAR. Special history type *BLOCKVAR allows you to plot a single quantity in a single block versus time, not limited to output times defined by *WSRF *GRID.

In addition, special history type *STRESSDIFF allows you to plot stress differential (maximum principal stress minus minimum principal stress) of a grid cell versus time, whereas a true triaxial stress-strain plot consists of *STRESSDIFF versus a strain.

STARS: No *GEOMECH output is available to the text ".out" file via *OUTPRN.

GEM: Most *GEOMECH output is available to the text ".out" file via *OUTPRN.

The geomechanics status/output file with name suffix ".geo" contains the echo of entered data, grid construction and equation solution. See keyword *PRINTGEO.

Geomechanics Porosities

Available for viewing in Results are several kinds of porosity generated by geomechanical calculations. Generally, porosity is pore volume divided by bulk or gross volume, but there are several different kinds of pore volume and bulk volume.

True Porosity is current pore volume divided by current bulk volume, each generated by geomechanics calculations. This ratio is dumped to SR2 via SRF_GRID quantity VPOROSTGEO and appears as Results menu item "True Porosity - Geomechanics".

Geo-corrected Porosity is current pore volume divided by initial bulk volume. This ratio is dumped to SR2 via *OUTSRF *GRID VPOROSGEO and appears as Results menu item "Porosity – Geo-corrected". Current pore volume is generated by geomechanics calculations, but initial bulk volume is static. The initialization process ensures that each geomechanics cell has the same initial bulk volume and pore volume as the corresponding fluid-flow cell.

Fluid-flow equations cannot use geomechanical porosity directly but instead get porosity from a correlation whose coefficients are provided by the Geomechanics module. Since the fluid-flow porosity is based upon initial bulk volume, it is comparable to the *Geo-corrected Porosity*. The magnitude of difference between the *Geo-corrected Porosity* and the fluid-flow porosity is an indication of the quality of the coupling between fluid-flow and geomechanics simulation. This difference is dumped to SR2 via SRF_GRID quantity PORDIFF and appears as Results menu item "Void Porosity Difference".

STARS: In geomechanics calculations all pore volumes correspond to "void" pore volume as opposed to "fluid" pore volume (see Appendix F.2).

Visualizing Geomechanics Grid Deformation

Keyword *WSRF *GRIDDEFORM allows the user to view grids in Results that deform with time as calculated by the geomechanics module, but there are some restrictions and other issues. See **Visualizing Geomechanics Grid Deformation** in the EXPLANATION for keyword *WSRF.

Storage Usage

Storage usage by the geomechanics option can be significant, especially for *GEOM3D. Stress/strain equations are solved at block corners and for the specified directions, so the number of equations per block may be much larger than those used to solve the fluid flow.

Casing Treatments with Radial Grids

A radial grid normally is used to model a single wellbore. The boundary between wellbore and formation is the inner-most radial face of the Geomechanics grid. This grid face can have two types of boundary condition in the radial direction, according to the wellbore's casing treatment.

1. **Cased Hole**: A rigid casing corresponds to the boundary being **constrained**, that is, grid nodes on that boundary are not allowed to move in the radial direction.
2. **Open Hole**: No casing corresponds to the boundary being **unconstrained**, that is, grid nodes on that boundary are allowed to move radially. In addition, fluid

pressure in the wellbore provides stress support in the radial direction. For example, sudden drawdown lowers the supporting stress from the wellbore which in turn causes the hole to contract.

A node is constrained by applying a prescribed displacement of zero. For 2D versus 3D finite-element grids (keyword *GEOM3D absent or present) this is done with different keywords which have different defaults. The following is a summary for radial grid types.

2D Radial Grids

When Reservoir Description data specifies *GRID *RADIAL with nj = 1, the host grid is **2D radial** since there is no resolution in the angular direction. The Geomechanics 2D finite-element grid conforms exactly to the host grid. The wellbore-formation boundary is the lower face of the inner-most radial ring which is accessed via I-direction index i = 1. For open-hole treatment, fluid pressure comes from the wellbore. See below Fig. 1.

Most well configurations are allowed (e.g, cycle between injector and producer, inject into some layers while producing from others). Wellbore pressure is well-defined for an open well but is not for a closed well. If it is important to model an equilibrated wellbore pressure during a closed well state, keep the well open with a small rate.

Use keyword ***PRESCBC** to constrain nodes. By default *PRESCBC constrains the wellbore-formation boundary (cased-hole treatment), so you must over-ride the default to obtain an open-hole treatment.

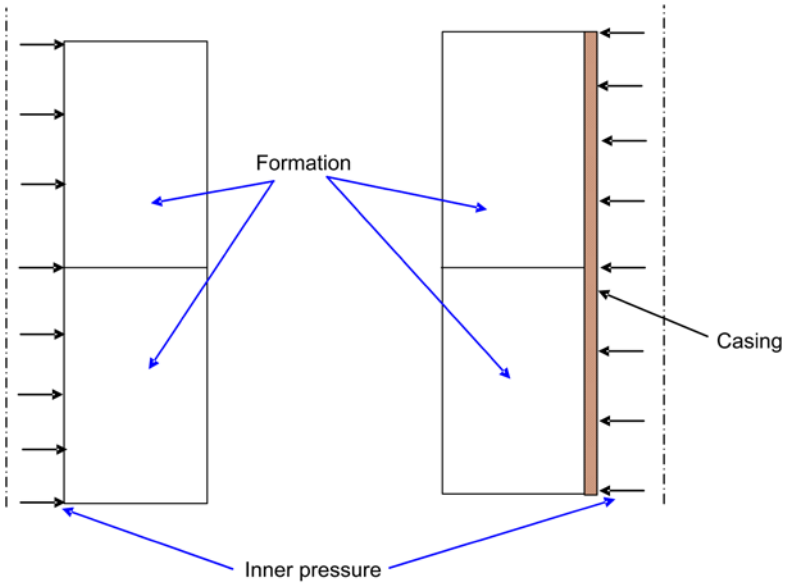


Figure 1: Wellbore hole treatment for 2D radial grid, without casing (left), and with casing (right)

3D Cylindrical Grids

When Reservoir Description data specifies *GRID *RADIAL with nj > 1, the host grid is **3D cylindrical** since there is resolution in the angular direction. However, this grid has no angular resolution in the inner-most radial ring (i = 1). The Geomechanics 3D finite-element

grid (*GEOM3D must be used) cannot conform to the host grid and so makes the following approximations:

1. The inner-most ring of the geomechanics grid is $i = 2$.
2. A corner-point grid is used, with straight lines replacing arcs between block corners. Each block's angle must not exceed 160 degrees, thereby preventing block volumes approaching zero.

Input and output uses the same I-J-K index system as the host grid, but data entered for $i = 1$ is ignored. Therefore, use I-direction index $i = 2$ to address the wellbore-formation boundary. For open-hole treatment, fluid pressure comes from the host grid $i = 1$. See below Fig. 2.

Use keyword ***PRESCBC3D** to constrain nodes. By default *PRESCBC3D does not constrain the wellbore-formation boundary (open-hole treatment), so you must over-ride the default to obtain a cased-hole treatment (opposite of the 2D case).

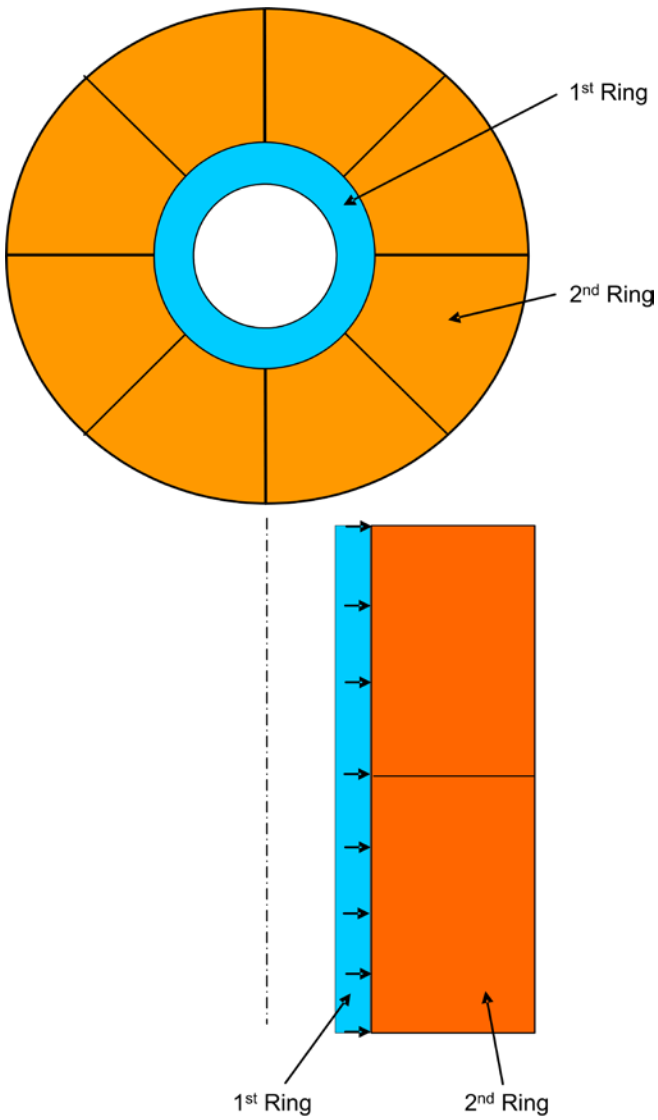


Figure 2: Wellbore hole treatment for 3D radial grid with nj = 8.

Post Processing of Prior Fluid/Heat Response for Geomechanics

A geomechanics calculation is linked to the host fluid-flow simulator by the flow of information, either one-way or two-way depending on the coupling option (see keyword *GCOUPLING). The more rigorous two-way coupling requires that the fluid-flow and geomechanics calculations be done in the same run. This can be inconvenient when many study runs are required (e.g., for history matching) and the geomechanics calculations cause longer run times (e.g., full 3D treatment via *GEOM3D). However, one-way geomechanics coupling is often sufficiently accurate and can be done conveniently with a pre-existing fluid-flow simulation response.

In one-way coupling the pressures and temperatures (if thermal) resulting from the fluid-flow simulator are passed to the geomechanics module to compute stress and strain field at a given time step, and there is no information flow from geomechanics back to the fluid flow calculations. Therefore it is not necessary for geomechanics calculations to be done in the same run as the fluid flow. Consequently it is possible to use the pressures and temperatures (if thermal) from time steps of a prior, existing fluid-flow simulation as input for geomechanics calculations.

The geomechanics post-processing feature (keyword *GEOPOSTPRO) allows the host simulator to scan through the simulation results files and/or restart files of a prior run and perform geomechanics calculations for time steps where pressure and temperature (if thermal) information are available. During such post-processing no fluid-flow time-step calculations are performed. This option allows you to evaluate multiple scenarios of geomechanics properties with one prior fluid-flow simulation result.

Both STARS and GEM share the *GEOPOSTPRO keyword syntax, but auxiliary control of the feature differs with simulator-specific keywords.

Geomechanical Model Identifier (Optional) *GEOMECH

PURPOSE:

Enable the geomechanical model.

FORMAT:

*GEOMECH

DEFINITIONS:

*GEOMECH
This indicates that keywords from the optional geomechanical model section will follow.

CONDITIONS:

Use of the geomechanical model is optional. However, if one of the sub-models is enabled, the mandatory data for that sub-model must be provided.

EXPLANATION:

The geomechanical model consists of a collection of two sub-models:

1. Plastic and nonlinear elastic deformation, and
2. Single-well boundary unloading analysis.

Each of these sub-models is enabled by the presence of at least one of its associated keywords. Model 1 and model 2 cannot be run together.

See the summary at the beginning of this section.

3D Finite Element

***GEOM3D**

PURPOSE:

Use 3D Finite Elements in computation.

FORMAT:

*GEOM3D

DEFAULT:

If keyword *GEOM3D and *PLSTRAINY are absent, then 2D plane strain finite elements normal to the X (I) direction are used.

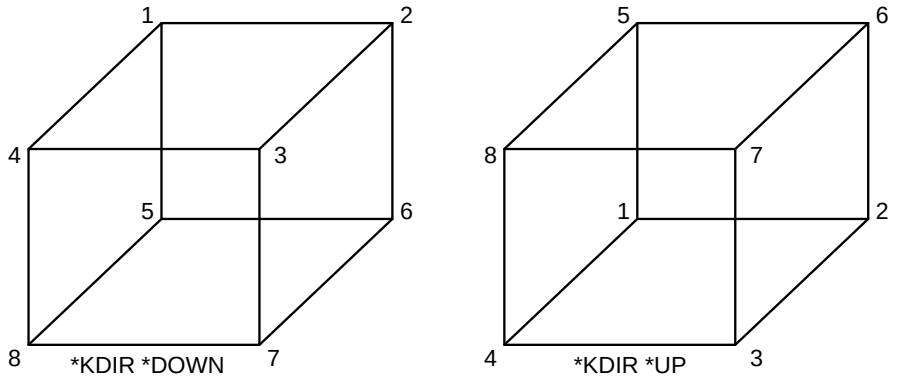
CONDITIONS:

*GEOM3D must appear immediately after keyword *GEOMECH when 3D finite elements are used.

This keyword is not allowed together with *PLSTRAINY.

EXPLANATION:

Currently, the 3D finite element has 8 nodes which are locally ordered as follows:



Example:

```
*GEOMECH
*GEOM3D   ** using 3D finite elements
```

Independent Geomechanics Grid *GEOGRID, *GDI, *GDJ, *GDK, *GEODEPTH, *GNULL, *GTRANSLI, *GTRANSLJ, *GTRANSLK, *GROTATEI, *GROTATEJ, *GROTATEK, *RCONBT, *RCONTP, *RCONLF, *RCONRT, *RCONBK, *RCONFT, *INTPOWER, *GPOLY, *GMONETONE

PURPOSE:

Use and specify geomechanics grid that is independent of the host simulator's grid.

FORMAT:

*GEOGRID (*GCART | *GRADIAL | *GCORNER) *ni nj nk*
*GDI (*GCON | *GIVAR) *I-size*
*GDJ (*GCON | *GJVAR) *J-size*
*GDK (*GCON | *GKVAR) *K-size*
*GEODEPTH (*GCENTER | *GTOP) *i j k depth*
*GNULL *IJK

*GTRANSLI *I-trans*
*GTRANSLJ *J-trans*
*GTRANSLK *K-trans*
*GROTATEI *I-angle*
*GROTATEJ *J-angle*
*GROTATEK *K-angle*

*RCONBT (*ALL | *IJK)
*RCONTP (*ALL | *IJK)
*RCONLF (*ALL | *IJK)
*RCONRT (*ALL | *IJK)
*RCONBK (*ALL | *IJK)
*RCONFT (*ALL | *IJK)

*INTPOWER *n*
*GPOLY

*GMONETONE *ALL
or
*GMONETONE { *IJK *host_range geo_range* }

DEFINITIONS:

*GEOGRID (*GCART | *GRADIAL | *GCORNER) *ni nj nk*

Use and specify a geomechanics grid that is independent of the host grid. The number of cells in each grid direction is *ni* (I-direction), *nj* (J-direction) and *nk* (K-direction). The K axis orientation (k increasing up or down) is the same as the host simulator grid which was specified by *KDIR.

For *GCART the geomechanics grid is Cartesian and *GEOM3D is used. The I, J and K grid directions are associated with X, Y and Z spatial directions, respectively.

For *GRADIAL the geomechanics grid is 2D radial (*nj* = 1) or 3D radial (*nj* > 1). The I, J and K grid directions are associated with radial, angular and Z spatial directions, respectively.

For *GCORNER the geomechanics grid is specified using keywords available to *GRID *CORNER (*ZCORN, *XCORN, *YCORN, *COORD and *CORNERS). When the manual entries for those keywords refer to *DI, *DJ or *DK, use *GDI, *GDJ and *GDK, respectively, in the *GCORNER context. For example, you may combine *ZCORN with *GDI and *GDJ. For *GCORNER the finite element method *GEOM3D is used.

*GDI (*GCON | *GIVAR) *I-size*

Specify cell size (m | ft | cm) in the I direction.

For *GCON, *I-size* is a single number that is applied to all *ni* cells.

For *GIVAR, *I-size* is *ni* separate numbers for cells with i=1 to i=*ni* in that order.

*GDJ (*GCON | *GJVAR) *J-size*

Specify cell size in the J direction. For *GCART *J-size* is linear length (m | ft | cm). For *GRADIAL *J-size* is angle in degrees and the total angle described may not exceed 360 degrees.

For *GCON, *J-size* is a single number that is applied to all *nj* cells.

For *GJVAR, *J-size* is *nj* separate numbers for cells with j=1 to j=*nj* in that order.

*GDK (*GCON | *GKVAR) *K-size*

Specify cell size (m | ft | cm) in the K direction. The K axis orientation (k increasing up or down) is the same as the host simulator grid which was specified by *KDIR.

For *GCON, *K-size* is a single number that is applied to all *nk* cells.

For *GKVAR, *K-size* is *nk* separate numbers for cells with k=1 to k=*nk* in that order.

*GEODEPTH (*GCENTER | *GTOP) *i j k depth*

Specify depth (m | ft | cm) of grid via *depth* for reference cell (*i*,*j*,*k*). The reference for depth is the same as for the host simulator's grid.

For *GCENTER, or if both *GCENTER and *GTOP are absent, *depth* is applied to the center of cell (*i*,*j*,*k*).

For *GTOP, *depth* is applied to the top of cell (*i*,*j*,*k*).

*GNULL *IJK

Specify which blocks in the geomechanics grid (specified by *GEOGRID) will be null, that is, there will be no geomechanics computation.

*GTRANSLI *I-trans*

Specify I-direction translation (m | ft | cm) of the geomechanics grid relative to the host simulator grid. For *GCART and *GCORNER, X = 0 in the geomechanics grid corresponds to X = *I-trans* in the host simulator grid.

*GTRANSLJ *J-trans*

Specify J-direction translation (m | ft | cm) of the geomechanics grid relative to the host simulator grid. For *GCART and *GCORNER, Y = 0 in the geomechanics grid corresponds to Y = *J-trans* in the host simulator grid.

*GTRANSLK *K-trans*

Specify K-direction translation (m | ft | cm) of the geomechanics grid relative to the host simulator grid. For *GCART and *GCORNER, Z = 0 in the geomechanics grid corresponds to Z = *K-trans* in the host simulator grid.

*GROTATEI *I-angle*

Specify rotation angle about I axis (degrees) of the geomechanics grid.

*GROTATEJ *J-angle*

Specify rotation angle about J axis (degrees) of the geomechanics grid.

*GROTATEK *K-angle*

Specify rotation angle about K axis (degrees) of the geomechanics grid.

*RCONBT (*ALL | *IJK)

Specify zero K-direction (vertical) deformation on bottom of host (fluid-flow) grid.

*RCONTP (*ALL | *IJK)

Specify zero K-direction (vertical) deformation on top of host (fluid-flow) grid.

*RCONLF (*ALL | *IJK)

Specify zero I-direction deformation on left side of host (fluid-flow) grid.

*RCONRT (*ALL | *IJK)

Specify zero I-direction deformation on right side of host (fluid-flow) grid.

*RCONBK (*ALL | *IJK)

Specify zero J-direction deformation on back side of host (fluid-flow) grid.

*RCONFT (*ALL | *IJK)

Specify zero J-direction deformation on front side of host (fluid-flow) grid.

*ALL

This keyword refers to all blocks in the grid.

*IJK

This keyword refers to selected blocks via one or more lines of the form

i1(:i2) j1(:j2) k1(:k2)

Each line specifies a single index or a range of indices in the I, J and K directions. The allowed range for each index is from 1 to the maximum number of blocks in that direction, and each lower index must not exceed the corresponding upper index.

*INTPOWER *n*

Specify power *n* in the inverse distance interpolation (IDI) used to interpolate variables between host and geomechanics grids.

*GPOLY

Using polynomial approach to interpolate deformation and stress between geomechanics and host grids.

*GMONETONE

Specify one-to-one mapping when all or part of the geomechanics grid is the same as (collocated with) the host **fundamental** grid. A block is collocated in the two grids when its spatial coordinate is the same for the two grids, for all three directions and eight corners. This option allows for faster mapping between the two grids. This option is especially useful when the host grid has local grid refinement.

*ALL The entire geomechanics grid is collocated with the entire host fundamental grid.

{ *IJK *host_range geo_range* }

Regions of the host fundamental grid are collocated with regions of the geomechanics grid. Specify each pair of rectangular regions with one *IJK line. Each *IJK line consists of *host_range* (I-J-K range in the host grid) and *geo_range* (I-J-K range in the geomechanics grid). Each I-J-K range is in the form *i1(:i2) j1(:j2) k1(:k2)*; the indices must not exceed the corresponding index range. For each *IJK line, *host_range* and *geo_range* must specify the same number of blocks in each of the three directions.

DEFAULTS:

If *GEOGRID is absent then the geomechanics grid is constructed to exactly overlay the host simulator's grid, in which case some restrictions are imposed on acceptable host grid options.

If *GEODEPTH is absent then *depth* = 0 is assumed.

If *GTRANSLI is absent then *I-trans* = 0 is assumed.

If *GTRANSLJ is absent then *J-trans* = 0 is assumed.

If *GTRANSLK is absent then *K-trans* = 0 is assumed.

If *GROTATEI is absent then *I-angle* = 0 is assumed.

If *GROTATEJ is absent then *J-angle* = 0 is assumed.

If *GROTATEK is absent then *K-angle* = 0 is assumed.

If *RCONBT is absent then the bottom of the host grid is not constrained.

If *RCONTP is absent then the top of the host grid is not constrained.

If *RCONLF is absent then the left side of the host grid is not constrained.

If *RCONRT is absent then the right side of the host grid is not constrained.

If *RCONBK is absent then the back side of the host grid is not constrained.

If *RCONFT is absent then the front side of the host grid is not constrained.

If *INTPOWER is absent then *n* = 2 is assumed.

If *GMONETONE is absent, no one-to-one mapping is applied.

CONDITIONS:

The intersection of the 3D spaces covered by the independent geomechanics and host simulator grids must not be null, that is, the two grids must have some 3D space in common.

If *GEOGRID is used, the minimum required keywords are *GDI, *GDJ and *GDK.

The translational and rotational options are available only with *GEOGRID.

EXPLANATION:

Grid Orientation

Host and geomechanics Cartesian grids are assumed to have the same grid orientation, that is, the geomechanics grid X, Y and Z directions are the same as the host grid X, Y and Z directions, respectively. Specifically, the Z axis orientation (k index increasing up or down) is the same as the host simulator grid which is specified by *KDIR.

Grid Translation

The origin (X,Y,Z) = (0,0,0) of the geomechanics grid is the corner of cell (1,1,1) that is furthest away from the rest of the grid. The origin of the host grid is defined similarly. By default the origins of the two grids are collocated.

Often the geomechanics grid extends beyond the host simulator grid, sometimes in both the negative and positive direction of a spatial axis. The *GTRANSLI/J/K keywords allow you to translate the origin of the geomechanics grid relative to the host grid origin. This lets you control the position the geomechanics grid over the host grid.

For example, a host grid covers a pay zone only but the geomechanics grid covers overburden, pay zone and underburden. Therefore, we want the geomechanics grid to extend beyond the host grid in both the –Z and +Z directions. If the host grid uses *KDIR *UP (origin on bottom) and the underburden thickness modelled by the geomechanics grid is 50 ft, then the vertical translation of the geomechanics grid would be specified by

```
*GTRANSLK -50
```

This would put the geomechanics grid origin 50 ft below the bottom of the host grid (pay zone).

Grid Rotation

The geomechanics grid can be rotated about the X, Y or Z axis to match with reservoir orientation. This option can also be combined with the option of grid translation. When rotation and translation are combined, the geomechanics grid is rotated first then translated.

For instance, the geomechanics grid is rotated about the X axis, Y axis and Z axis with angles of 30 degrees, 70 degrees and 120 degrees, respectively.

```
*GROTATEI 30    ** In degrees
*GROTATEJ 70    ** In degrees
*GROTATEK 120   ** In degrees
```

Plotting Results for Geomechanics Grid

Use keyword *GOUTSRF *GGRID to create a separate SR2 file set for writing results for the geomechanics grid. Geomechanics quantities (e.g., displacements) reported for individual cells may be more accurate with this SR2 file set which uses the "native" geomechanics grid. The same quantities reported in the host SR2 file set have gone through an interpolation step

and so may have lost some accuracy, especially if the two grids have cells sizes that are significantly different.

Constraining Host Grid Deformation

Deformations are calculated for all host grid cell corners via the 3D mapping. On a 2D plane boundary that the host and geomechanics grid have in common, the calculated host grid deformations normal to the plane may not be zero even though the geomechanics grid is constrained there to no deformation in that direction. Keywords *RCONBT, etc., force the host grid to have no deformation on the indicated boundary in the normal direction, in order to match the typical (default) host grid behavior on that boundary.

For the purpose of keywords *RCONBT, etc., "boundary" means the first non-null cell encountered when searching from the indicated direction. For example, for *KDIR *UP the bottom grid layer is k = 1 so that *RCONBT causes the searching in an I-J column to start at k = 1 and increases k until a non-null cell is encountered. For *KDIR *DOWN the bottom grid layer is k = nk so that *RCONBT searching in a column starts at k = nk and decreases k until a non-null cell is found.

The constraints of host grids can be applied on their outer boundaries or on any host grid. If one of the constrained keyword is used with *ALL, the outer boundary corresponding to the keyword is fully constrained. By convention:

Left:	from I =1 to I = NI;	Right:	from I = NI to I = 1	
Back:	from J =1 to J = NJ;	Front:	from J = NJ to J =1	
Top:	from K =1 to K = NK;	Bottom:	from K = NK to K = 1	For KDIR DOWN
Top	from K =NK to K = 1;	Bottom:	from K = 1 to K = NK	For KDIR UP

It should be noted that this kind of constraints is only applied to the host grids but not geomechanics grids. The constraint for geomechanics grids is controlled by the keyword *PRESCBC or *PRESCBC3D.

One-to-One Mapping

Mapping between the host and geomechanics grids involves identifying how block corners in one grid relate to block corners in the other grid. In the general case, identifying these relationships involves significant searching. However, very often the relationship between the two grids is known beforehand since the geomechanics grid is based on the host grid.

Keyword *GMONETONE allows you to specify the inter-grid relationship explicitly so that general searching can be minimized or eliminated. This is accomplished by indicating which blocks in one grid are collocated with which blocks in the other grid. Conventional mapping is used for blocks not flagged via *GMONETONE.

This search saving is especially significant when the host grid has local refinements, resulting in many more small host blocks than geomechanics blocks. *GMONETONE links geomechanics blocks to fundamental blocks in the host grid, eliminating the need to search through the host grid’s numerous child blocks during inter-grid mapping.

For each collocation indicated by *GMONETONE, the block coordinates are checked for actual collocation status to decide if the one-to-one mapping can be applied.

Example:

```
*GEOGRID *GCART 50 40 10     ** Cartesian grid
*GDI  *GCON   10             ** Size of grid
```

```
*GDJ  *GCON   5              ** Size of grid
*GDK  *GKVAR  2*1 5*3 3*10   ** Size of grid
*GNULL  *IJK                 ** Null blocks
    20:30   15:30    3:5
*GEODEPTH  *GTOP   1000      ** Top of first block
*GMONETONE *ALL   ** Entire grid same as host grid
*GTRANSLI  -100              ** X translation
*GTRANSLJ  -200              ** Y translation
*GTRANSLK   500              ** Z translation
*GROTATEI   30               ** Rotate about X
*GROTATEJ   70               ** Rotate about Y
*GROTATEK   120              ** Rotate about Z
*INTPOWER  3                 ** Power used for IDI
** constrain the host grid in full (*ALL) or part (*IJK)
*RCONBT *ALL                 ** On the bottom
*RCONLF *ALL                 ** On the left
*RCONRT *ALL                 ** On the right
*RCONBK *ALL                 ** On the back
*RCONFT *ALL                 ** On the front
*RCONTP *IJK                 ** On the top of layer 5
   1:10  1:10  5
```

In this example the geomechanics grid is actually an extension of the host grid, that is, host block (1,1,1) is collocated with geomechanics block (18,1,3).

```
*GRID *CORNER 16 25 9
. . .
*GEOGRID *GCORNER 33 25 14
*GMONETONE  ** --Host grid---   ----Geo Grid----
     *IJK      1:16  1:12 1:9   18:33  1:12 3:11
     *IJK      1:16 13:25 1:9   18:33 13:25 3:11
```

References:

Tran, D., Nghiem, L. and Buchanan L.: "Aspects of Coupling between Petroleum Reservoir Flow and Geomechanics," paper ARMA 09-08 presented at the 43rd US Rock Mechanics Symposium and 4th US-Canada Rock Mechanics Symposium, Asheville, NC, USA, June 28-July 1, 2009.

Tran, D., Buchanan, L. and Nghiem, L.: "Improved Gridding Technique for Coupling Geomechanics to Reservoir Flow," SPE Journal, Vol. 15, pp. 64-75, 2010.

Independent Geomechanics Graphics *GOUTSRF

PURPOSE:

Enable independent graphics file (SR2) for geomechanics.

FORMAT:

*GOUTSRF *GGRID (*ALL | *NONE | (*REMOVE) *item_list*)

DEFINITIONS:

*GGRID (*ALL | *NONE | (*REMOVE) *item_list*)

Specifies grid quantities (one value for each grid block) to be written to the geomechanics SR2 file at times determined by the host simulator via *WSRF *GRID. Generally, each item on the GSRF_GRID list is flagged for writing as either enabled or disabled. The simulation starts with all items disabled. Use *item_list* (selected keywords in the GSRF_GRID list) to enable individual items. Use *ALL to enable all items. Use *REMOVE with *item_list* to disable individual items, or use *NONE to disable all items.

Enabling GSRF_GRID items for writing can increase the size of the geomechanics SR2 files. The availability of some items depends on the use of other geomechanics keywords or options.

GSRF_GRID List

The GSRF_GRID list consists of all properties and quantities in the following table.

PRES	Fluid pressure
TEMP	Temperature
STRESI:	Effective I-direction stress (X or R)
STRESJ:	Effective J-direction stress (Y or theta)
STRESK:	Effective K-direction stress (Z)
STRESSH:	Shear stress (Y-Z or R-Z) for plane strain only
STRESMXP:	Maximum principal stress (+ for compressive, – for tensile)
STRESMNP:	Minimum principal stress (+ for compressive, – for tensile)
STRESINT:	Intermediate principle stress (+ for compressive, – for tensile)
VMSTRESS:	Von Mises stress
STRNEPL:	Effective Plastic strain
STRESEFF:	Mean effective stress (+ for compressive, – for tensile)
STRESSM:	Mean total stress (+ for compressive, – for tensile)
STRESNORM:	Effective stress normal to fracture
PRMXDIR:	Vector of maximum principle effective stress (*GRID only)
PRMNDIR:	Vector of minimum principle effective stress (*GRID only)
STRAINI:	I-direction normal strain (X or R)
STRAINJ:	J-direction normal strain (Y or theta)
STRAINK:	K-direction normal strain (Z)

STRAINSH:	Shear strain
STRNMXP:	Maximum principle strain
STRNMNP:	Minimum principle strain
STRAINVOL:	Volumetric strain
VERDSPLGEO:	Vertical displacement "up" based on geomechanics, at centre of cell
TVERDPLGEO:	Vertical displacement "up" based on geomechanics, at top of cell
SUBSIDGEO:	Vertical displacement "down" (subsidence) based on geomechanics, at centre of cell (negative of VERDSPLGEO)
TSUBSIDGEO:	Vertical displacement "down" (subsidence) based on geomechanics, at top of cell (negative of TVERDPLGEO)
VDISPL:	Vector of grid displacement
YLDSTATE:	Stress state = 0 In Elastic state = 1 On shear failure envelope = 2 On the compressive cap = 3 At the corner (intercept between cap and shear failure envelope) = 4 On the tensile cutoff surface
BIOT:	Biot's constant
GCOHESION:	Cohesion value
HARDENING:	Hardening parameter
POISSON:	Poisson's ratio
POISSONEQ:	Equivalent Poisson's ratio for jointed rock mass
YIELD:	Yielding stress
YOUNG:	Young's elastic modulus
YOUNGEQ:	Equivalent Young's modulus for jointed rock mass
JYOUNGMI:	Young's modulus of jointed rock mass in the I direction
JYOUNGMJ:	Young's modulus of jointed rock mass in the J direction
JYOUNGMK:	Young's modulus of jointed rock mass in the K direction
FRICANGLE :	Friction angle
THEXPCOEF:	Linear thermal expansion coefficient of rock
THCOMPR:	Thermal compressibility of rock
BULKVOL:	Bulk volume
GEORTYPE:	Rock type number for geomechanical material
STRESRAD:	Radial total stress (available only for 3D radial grid)
STRESTAN:	Tangential total stress (available only for 3D radial grid)
ESTRESRAD:	Radial effective stress (available only for 3D radial grid)
ESTRESTAN:	Tangential effective stress (available only for 3D radial grid)
SHSTRRAD:	Radial shear stress (available only for 3D radial grid)
DISPLRAD:	Radial displacement (available only for 3D radial grid)
DISPLTAN:	Tangential displacement (available only for 3D radial grid)
REACFORCE	Reaction force (See **Reaction Force** below)
SAFACTOR	Shear safety factor (see **Shear Safety Factor** below)
TSAFACTOR	Tensile safety factor (see **Tensile Safety Factor** below)

DEFAULTS:

If *GOUTSRF is absent then a separate geomechanics SR2 file set will not be created.

CONDITIONS:

The host simulator (via keyword *WSRF *GRID) determines the times at which the specified quantities are written to the geomechanics SR2 file.

*GOUTSRF *GGRID may appear in the Well and Recurrent Data section as well as the Geomechanics data section.

EXPLANATION:

Since an SR2 file set can contain only one grid, the *GEOGRID option requires a separate geomechanics SR2 file set for the independent geomechanics grid. Keyword *GOUTSRF triggers the creation of a geomechanics SR2 file set that is separate from the host simulator's SR2 file set. The geomechanics SR2 files have the same base name as the host output files, with extensions ".gmch.irf" and ".gmch.mrf". The geomechanics SR2 file set uses the same units as the host simulator's SR2 file set.

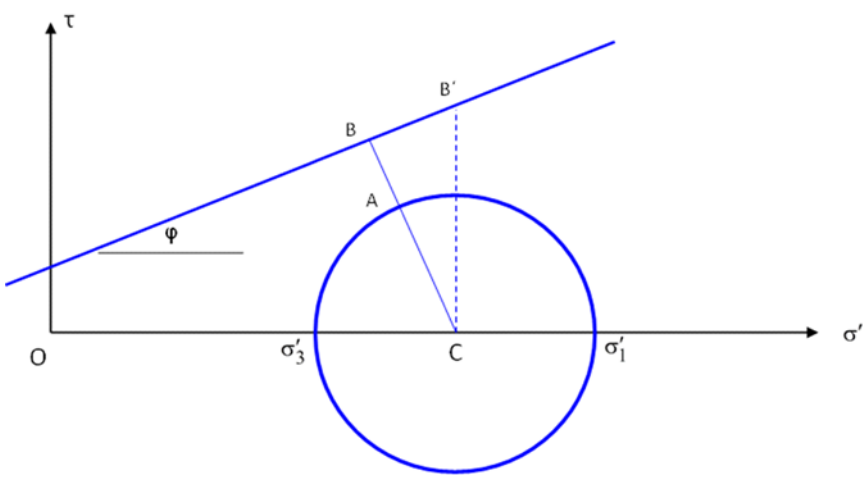
If *GOUTSRF *GGRID is used when *GEOGRID is absent, the geomechanics SR2 file set is created and the information written to it is the same as what can be written to the host SR2 file set via the geomechanics-specific sub-keywords of *OUTSRF *GRID.

Reaction Force

Generally, a geomechanics solution balances all the forces over all the nodes. A node on a prescribed boundary is not free to move, so it exerts a reaction force to counter the forces from neighbouring nodes. Consequently, a node with no prescribed boundary condition has no (zero) reaction force. Reaction force is a vector and so has components in the grid directions. The reaction force specified by sub-keyword REACFORCE is the sum of all reaction forces acting on nodes of that element (host grid cell). To report reaction forces at the nodes of the geomechanics grid, see keyword *PRINTGEO.

Shear Safety Factor

Consider the Mohr Coulomb circle and failure envelope shown below.



Mohr Coulomb circle and the failure envelope

For this set of stresses the safety factor is CB/CA where

$$CA = \frac{(\sigma'_1 - \sigma'_3)}{2}$$

$$CB = \cos(\varphi) \cdot CB'$$

Here φ is friction angle, and the estimated value of CB´ is based on the failure function of the constitutive model. The location of C is given by

$$OC = \frac{(\sigma'_1 + \sigma'_3)}{2}$$

The safety factor is defined as: SF = 1 – (minimum of (1, CA/CB)).

If the ratio CA/CB is greater than 1, the safety factor (SF) is zero. If the safety factor is greater than 0, the material is still elastic and safe from failure since the stresses are under the failure envelope. If safety factor is 0 or near 0, the material either has failed or will fail soon.

However, the shear safety factor can be used as a measure of **how close** a material is to failure. This quantity is available at different locations and times, so it can be used to reveal trends in space and time. For example, at a specified time you may wish to find the minimum value of safety factor over the whole grid. Alternately, at a specified location you could observe the trend of safety factor over time to see if and how it approaches failure.

The maximum reported value of safety factor is 1, which corresponds to "very far from failure". Therefore grid plots at early times may show 1 in most cells.

The shear safety factor specified by sub-keyword SAFACTOR corresponds to the stresses at the centre of the element (host grid cell). To report shear safety factor at the nodes of the geomechanics grid, see keyword *PRINTGEO.

Tensile Safety Factor

This type of safety factor is defined as:

$$TSF = \frac{\sigma'_3}{\sigma_3}$$

Where (according to the above figure):

$\sigma'_{3:}$ Minimum effective stress as shown in the section of Shear Safety Factor

$\sigma_{3:}$ Minimum total stress

If this value closes to 0, the stress is near the tensile fracture which is assumed to be at the origin of τ and σ´coordinates. The maximum value of tensile safety factor is 1. Even though the value of tensile safety factor is greater than 0; i.e. the material will not fail due to the tensile stress, the material may fail due to the shear stress when its value of SF is 0.

Example:

```
*GEOMECH
   ** Independent geomechanics grid definition
   *GEOGRID *GCART 50 40 10
   …
   *GOUTSRF *GGRID *PRES *STRESEFF * STRAINVOL
*TIME 0
   …
   *GOUTSRF *GGRID *TEMP *NONE  ** Disable all
*TIME 100
   …
   *GOUTSRF *GGRID *STRNEPL
*TIME 200
*STOP
```

Global Amalgamation ***GAMALI, *GAMALJ, *GAMALK**

PURPOSE:

Apply amalgamation to geomechanics grid.

FORMAT:

*GAMALI ni_{am} (*GCON i_{raw}:i_{am} | *GVAR { i_{raw}:i_{am} })
*GAMALJ nj_{am} (*GCON j_{raw}:j_{am} | *GVAR { j_{raw}:j_{am} })
*GAMALK nk_{am} (*GCON k_{raw}:k_{am} | *GVAR { k_{raw}:k_{am} })

DEFINITIONS:

*GAMALI ni_{am} (*GCON i_{raw}:i_{am} | *GVAR { i_{raw}:i_{am} })

Specify a geomechanics grid whose I direction is based on the raw grid but modified with global amalgamation. ni_{am} is the number of I-direction blocks after amalgamation and must range from 1 to the raw-grid ni.

Amalgamation directive i_{raw}:i_{am} is a ratio of raw-grid blocks i_{raw} to amalgamated-grid blocks i_{am}. You may specify i_{am} = 1 for amalgamation or $i_{am} = i_{raw}$ for no amalgamation; no other values are allowed. When i_{am} = 1, all i_{raw} raw blocks are amalgamated into a single new block. When amalgamation is complete, the resulting number of I-direction blocks must be ni_{am}.

*GCON — Starting with i=1, apply the directive multiple times until raw blocks are exhausted. Near i=ni, the directive is not applied to a block group smaller than i_{raw}.

*GVAR — Starting with i=1, apply each listed directive once, in the specified order. Use i_{am}= i_{raw} for a group of i_{raw} blocks that are to remain unchanged. All the i_{raw} must sum to the raw-grid ni, and all the i_{am} must sum to ni_{am}.

*GAMALJ nj_{am} (*GCON j_{raw}:j_{am} | *GVAR { j_{raw}:j_{am} })

Specify global amalgamation of blocks in the J direction, from the raw-grid nj to nj_{am}. Directive j_{raw}:j_{am} works in the J direction in the same way that i_{raw}:i_{am} works in the I direction (see *GAMALI).

*GAMALK nk_{am} (*GCON k_{raw}:k_{am} | *GVAR { k_{raw}:k_{am} })

Specify global amalgamation of blocks in the K direction, from the raw-grid nk to nk_{am}. Directive k_{raw}:k_{am} works in the K direction in the same way that i_{raw}:i_{am} works in the I direction (see *GAMALI).

DEFAULTS:

If *GAMALI is absent, no amalgamation is done in the I direction; the corresponding comment applies for the J and K directions as well.

CONDITIONS:

These keywords may not appear together with *PLSTRAINY (plain strain method).

These keywords may not be used when the geomechanics grid is radial.

Each of these keywords should appear no more than once.

If the geomechanics grid is extended as well as amalgamated, the amalgamation specified by these keywords is applied before the extending.

Even if *GEOGRID is absent, a separate geomechanics SR2 file (see *GOUTSRF) is generated since these keywords make the geomechanics grid different from the host simulator grid.

EXPLANATION:

The raw geomechanics grid may be specified in one of two ways: (1) a copy of the host simulator's fundamental grid (*GEOGRID absent) or (2) via *GEOGRID. Keywords *GAMALI, *GAMALJ and *GAMALK allow you to modify that raw grid by global amalgamation, in each direction separately. Global amalgamation means that amalgamation in one grid direction is applied to all blocks in the plane normal to that direction.

Amalgamation is based on the corner coordinates of the raw grid. For a Cartesian grid, local and total grid volumes will be conserved. For a corner-point grid, local volumes may not be conserved depending on the variation of grid-line trajectories. The conservation of total grid volume will depend on the degree to which the grid's outer surfaces are non-planar, but the difference should not be large in normal cases.

Amalgamating the geomechanics grid has no effect on the host grid itself, but it does affect the mapping between host and geomechanics grid.

Example:

The host simulator's Cartesian grid is 33×25×14 with *KDIR *DOWN and *GEOGRID is absent, so the raw geomechanics grid is the same 33×25×14 grid. Use the following data to reduce the number of geomechanics blocks by selective amalgamation.

```
*GRID *CART 33 25 14  ** Host grid

. . .
*GEOMECH
*GEOM3D

** Copy host grid, then amalgamate
*GAMALI 20 *GVAR 3:1 3:1 6:6 3:1 2:1 1:1 3:1 6:6 3:1 3:1
*GAMALJ 15 *GVAR 6:1 13:13 6:1
*GAMALK  7 *GCON 2:1
```

The resulting geomechanics grid is 20×15×7. In the K direction amalgamation is done with the one reduction ratio "2:1", resulting in each of 7 raw-block pairs amalgamated to one block. In the J direction the three directives result in the first 6 blocks and last 6 blocks being amalgamated into one block each while the middle 13 blocks are unchanged; there are 6+13+6 = 25 (host nj) raw blocks and 1+13+1 = 15 (nj_{am}) new blocks.

In another example, *GEOGRID specifies the raw geomechanics grid which is then amalgamated.

```
*GRID *RADIAL 10 20 9  ** Host grid

. . .

*GEOMECH
*GEOM3D
*GEOGRID *GCART 22 22 11  ** Geomechanics raw grid

** Amalgamate the separate geomechanical grid
*GAMALI  16   GVAR  2:1 2:1 2:1 10:10  2:1  2:1  2:1
*GAMALJ  16   GVAR  2:1 2:1 2:1 10:10  2:1  2:1  2:1
*GAMALK  11   GCON  1:1
```

The resulting geomechanics grid is 16×16×11. In the I and J directions the first and last three directives result in the first and last 6 blocks being amalgamated into 3 blocks each while the middle 10 blocks are unchanged; there are 6·2+10 = 22 raw blocks and 6·1+10 = 16 (nj_{am}) resulting blocks. No amalgamation is done in the K direction, so keyword *GAMALK could be absent.

Extending Geomechanics Grid *GEOEXTD, *GEXMINI, *GEXMAXI, *GEXMINJ,*GEXMAXJ, *GEXMINK,*GEXMAXK, *GELEVATB, *GELEVATT, *GEXDI, *GEXDJ, *GEXDK

PURPOSE:

Use and specify geomechanics grid that is extended from host simulator's grid.

FORMAT:

*GEOEXTD ni_1:ni_2 nj_1:nj_2 nk_1:nk_2
*GEXDI (*GCON | *GVAR) *I-size*
*GEXDJ (*GCON | *GVAR) *J-size*
*GEXDK (*GCON | *GVAR) *K-size*
*GEXMINI *xmin*
*GEXMAXI *xmax*
*GEXMINJ *ymin*
*GEXMAXJ *ymax*
*GEXMINK *zmin*
*GEXMAXK *zmax*
*GELEVATB
*GELEVATT

DEFINITIONS:

*GEOEXTD ni_1:ni_2 nj_1:nj_2 nk_1:nk_2

Extending grid option can be used with host grid or geomechanics grid.

When the keyword GEOGRID is absent, the keyword GEOEXTD is automatically applied to the host grid. Otherwise, the keyword GEOEXTD is applied to the geomechanics grid.

ni_1:ni_2 : number of extending grids on the left and right of the field along the I direction. This starts counting extra grids from the field to outer.

nj_1:nj_2 : number of extending grids on the back and front of the field along the J direction. This starts counting extra grids from the field to outer.

nk_1:nk_2 : number of extending grids on the top and below of the field along the K direction. KDIR UP and KDIR DOWN are applied. This starts counting extra grids from the field to outer.

It should be noted that when only one single grid is used (i.e. the keyword GEOGRID is absent), the plane strain approach can be used with the extending grid.

When the extending grid option is employed, the total number of grids along the I direction should be: ni_1 + NI + ni_2 which is used by the geomechanics module, where NI is the number of existing grids in the I direction. In the other directions, the total number of grids is estimated in the same manner.

*GEXDI (*GCON | *GVAR) *I-size* (m | ft | cm)

Specify cell size in the I direction.

For *GCON, *I-size* is a single number that is applied to all ni_1 and ni_2 cells.

For *GVAR, *I-size* is ni_1+ni_2 separate numbers for cells with i = 1 to i = ni_1 and i =1 to i = ni_2 in that order.

*GEXDJ (*GCON | *GVAR) *J-size* (m | ft | cm)

Specify cell size in the J direction. For *GRADIAL *J-size* is angle in degrees and the total angle described may not exceed 360 degrees.

*GCON, and *GVAR subkeywords are similar to *GEXDI but *GVAR applies to J direction.

*GEXDK (*GCON | *GVAR) *K-size* (m | ft | cm)

Specify cell size in the K direction. The K axis orientation (k increasing up or down) is the same as the host simulator grid which was specified by *KDIR.

*GCON, and *GVAR subkeywords are similar to *GEXDI but *GVAR applies to K direction.

*GEXMINI *xmin* (m | ft | cm)

Instead of using the keyword *GEXDI to express the grid size of extending grid, this keyword is used along with the value *xmin* to indicate the minimum bound of the extending grid on the left side along the I direction. In this case, the size of each extending grid is automatically computed so that the last extending grid is at the location of *xmin*.

*GEXMAXI *xmax* (m | ft | cm)

The keyword is used along with the value *xmax* to indicate the maximum bound of the extending grid on the right side along the I direction. In this case, the size of each extending grid is automatically computed so that the last extending grid is at the location of *xmax*.

*GEXMINJ *ymin* (m | ft | cm)

Same as *GEXMINI above, but the grid size of extending grid applies in the J direction.

*GEXMAXJ *ymax* (m | ft | cm)

Same as *GEXMAXI above, but the grid size of extending grid applies in the J direction.

*GEXMINK *zmin* (m | ft | cm)

See *GEXMINI above, but the grid size of extending grid applies in the K direction (assumes *KDIR *DOWN).

*GEXMAXK *zmax* (m | ft | cm)

Same as *GEXMAXI above, but the grid size of extending grid applies in the K direction (assumes *KDIR *DOWN).

*GELEVATB

Allow the extending grid to follow the bottom configuration of the existing grid. In absence of this keyword the extending grids at the bottom are on the same plane (see the figure in the explanation below).

*GELEVATT

Allow the extending grid to follow the top configuration of the existing grid. In absence of this keyword the extending grids at the top are on the same plane (see the figure in the explanation below).

DEFAULTS:

If *GEOEXTD is absent then the geomechanics grid, there is no extending grid added.

If *GELEVATB is absent all the extending grids at the bottom are on the same horizontal plane.

If *GELEVATT is absent all the extending grids at the top are on the same horizontal plane.

CONDITIONS:

If *GEOEXTD is used, the minimum required keywords are *GEXDI or (*GEXMINI,*GEXMAXI), *GEXDJ or (*GEXMINJ, *GEXMAXJ) and *GDK or (GEXMINK, *GEXMAXK). When *GEXDI is used, *GEXMINI and *GEXMAXI should not be used and vice versa.

Keywords in J direction and K direction are treated in a similar manner.

If geomechanics and host grids are same and geomechanics grid is extended using this option, plain strain approach **can** be used. However if geomechanics grid is different from the host grid, (i.e. keyword GEOGRID is used) and the geomechanics grids are extended using the extended grid, plain strain approach **cannot** be used.

If the extending grid is combined with the amalgamating grid (*GAMALI etc.), the plane strain approach **cannot** be used.

The extending grid **cannot** be used with locally refined host grid.

If there is no extending grid added on one side of the grid boundary, a value of zero must be used. For instance, *GEOEXTD 0:5 0:0 5:5 would mean that there is no extending grid on the left side of the grid boundary in the I direction; and there is no extended grid on the either side of the grid boundary in the J direction.

EXPLANATION:

The extending grid is designed so that additional grid blocks could be added automatically to the geomechanics grid in either of the I, J, or K directions.

Just like GEOGRID when this option is used, two sets of Simulation Results files are created, i.e, one for the host grid and the other for the geomechanics grid.

The size of extending grid can be assigned by either keyword *GEXDI or keywords *GEXMINI and *GEXMAXI. Note that *GEXDI permits the extended grid to be specified as a contant value (via *GCON) or a varying value (via *GVAR) in I direction while keywords *GEXMINI and *GEXMAXI allow the size of extending grid to be based on the distances from the existing grid boundary. The distance is equally divided into number of extended grids specified by *GEOEXTD.

Specification of extended grid in other directions is treated in a similar manner.

When assigning the geomechanical properties, initial stresses and boundary conditions etc. the total of grids for geomechanics must include the extending grid. The number of grids in each direction must be:

I direction = ni_1+ NI + ni_2

J direction = nj_1 + NJ + nj_2

K direction = nk_1 + NK + nk_2

where,

ni_1, ni_2, nj_1, nj_2, nk_1, nk_2 are the numbers of extending grids.

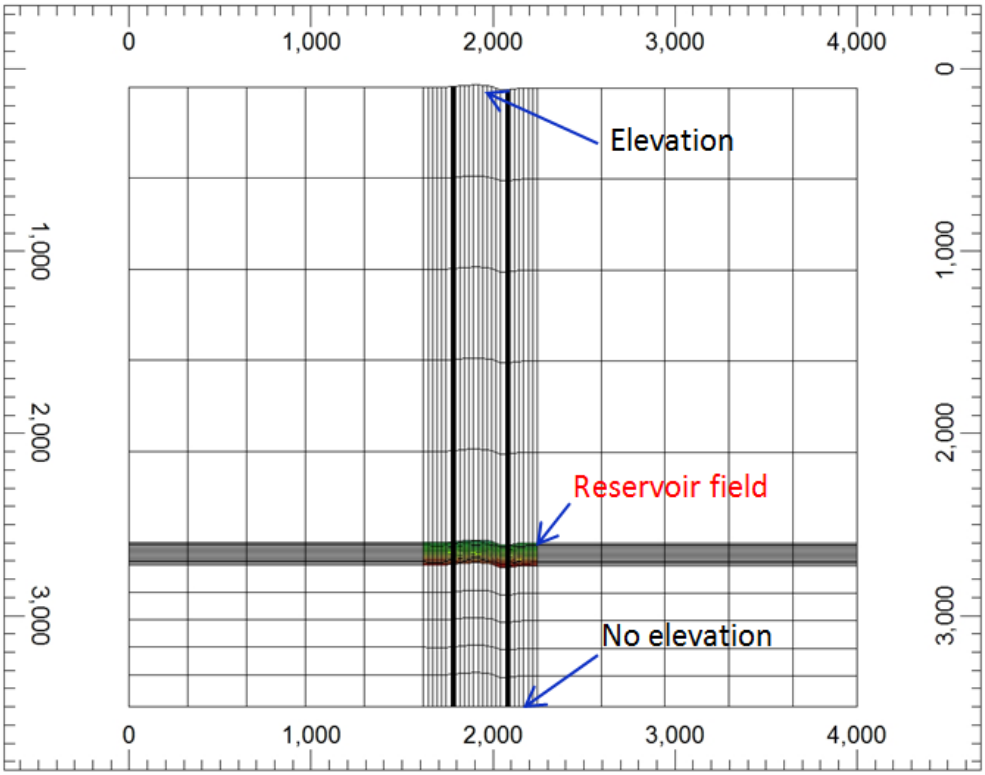
NI, NJ and NK are the existing number of grids in I, J, K directions.

Example:

A reservoir has 33×25×14 grids. The extending grid is added in the geomechanics section of the dataset as follows:

```
*GEOMECH                    ** geomechanics main keyword
** Extended grids
*GEOEXTD  5:5   5:5   5:5
*GEXMINI  0.0               ** Minimum bound in the I direction
*GEXMAXI  4000.0            ** Maximum bound in the I direction
*GEXMINJ  1000              ** Minimum bound in the J direction
*GEXMAXJ  4000.0            ** Maximum bound in the J direction
*GEXMINK  0.0               ** Minimum bound in the K direction
*GEXMAXK  3500.0            ** Maximum bound in the K direction
*GELEVATT                   ** Elevating the top of geomechanics region
*GEOROCK 1                  ** Rock Type 1
*ELASTMOD 5.0e5             ** Young's modulus
*POISSRATIO .35             ** Poisson's ratio
*COHESION 100000000         ** Cohesion
*GEOROCK 2                  ** Rock Type 2
*ELASTMOD 5.0e6             ** Young's modulus
*POISSRATIO .35             ** Poisson's ratio
*COHESION 100000000         ** Cohesion
*GEOTYPE   ijk
**         i      j       k    rocktype
           1:43   1:35   1:24      2
           6:38   6:30   6:19      1      ** rock type 1 for layers 6 to 19
*STRESS3D 2000 2000 4000 0.0 0.0 0.0
```

For geomechanics, the number of grids is (43x35x24) since there are 10 more grids are added in each direction of the reservoir. Therefore, when assigning the rock types, the geomechanics grid must be used instead of reservoir grid.



Based on the information of extending grid given above, a plot of the geomechanics grid is displayed. Since the keyword *GELEVATT is used, the top of geomechanics grid follows the top configuration of the existing grid (color), whereas, the bottom of the geomechanics grid is flat since keyword *GELEVATB has not been used. The bounding box has a minimum value of zero and a maximum value of 4000 in the horizontal direction. In the vertical direction, the box has minimum and maximum values of zero and 3500, respectively.

Plane Strain Option *PLSTRAINY

PURPOSE:

Specify that strains will be calculated by a series of planes normal to the Y (J) direction.

FORMAT:

*PLSTRAINY

DEFINITIONS:

*PLSTRAINY
Indicates that the 3D strains will be calculated as a series of 2D planes normal to the Y (J) direction.

DEFAULTS:

If *PLSTRAINY and *GEOM3D are absent, then the strains will be calculated by a series of planes normal to the X (I) direction.

CONDITIONS:

This keyword works only with Cartesian and corner-point grids. Radial grids are not allowed.

This keyword is not allowed together with *GEOM3D.

EXPLANATION:

Three-dimensional strain calculations are performed by a plane strain method, where strains are calculated in a series of 2D planes normal to one grid direction. Consequently, strains are assumed to be zero in that direction. The purpose of using this method is to improve the speed of stress-strain calculations while maintaining appropriate mechanical responses due to disturbances in the reservoir. In order to minimize estimating errors, it is recommended that the user select the direction in which the strains can be ignored. Normally, strains are smallest in the longest horizontal extent of a reservoir field.

Example:

```
*GEOMECH        ** keyword for coupling with the geomechanics module
*PLSTRAINY      ** keyword for plane strains along the Y direction
```

Deformation Rock Type

***GEOROCK, *GEOTYPE**

PURPOSE:

Assign multiple deformation rock types.

FORMAT:

*GEOROCK typ_num (*COPY old_typ_num)

ARRAY:

*GEOTYPE

DEFINITIONS:

typ_num
: Current rock type number to be applied to data which depends on rock type. See *GEOMECH. Data is read starting with type #1 as default. 'typ_num' can be any number from 1 to the maximum dimensioned value.

old_typ_num
: Previously defined rock type which is to be copied into current rock type. *COPY is optional. This option is useful when rock types differ in only a few properties.

*GEOTYPE
: Assigns rock type numbers to the grid blocks, using any of the array reading options. This array is initialized with rock type #1 for every block.

DEFAULTS:

*GEOROCK 1

*GEOTYPE *CON 1

CONDITIONS:

All the rock type numbers assigned using *GEOTYPE must be defined by *GEOROCK.

EXPLANATION:

For a list of the keywords which can vary with rock type, see section **Multiple Rock Types** in the summary at the beginning of this chapter.

Plastic Model Formation Properties *ELASTMOD, *POISSRATIO, *YLDSTRESS, *COHESION, *HARDEN, *FRICANGLE, *BIOTSCOEF, *DILANGLE

PURPOSE:

Define strength properties of the formation.

FORMAT:

*ELASTMOD	*elastmod*
*POISSRATIO	*poissratio*
*YLDSTRESS	*yldstress*
*COHESION	*cohesion*
*HARDEN	*harden*
*FRICANGLE	*fricangle*
*BIOTSCOEF	*biotscoef*
*DILANGLE	*dil_angle*

DEFINITIONS:

elastmod

Young's elastic modulus (kPa | psi | kPa).

poissratio

Poisson's ratio (dimensionless).

yldstress

Yield stress for Tresca and Von Mises materials (kPa | psi | kPa).

cohesion

Cohesion for Mohr-Coulomb and Drucker-Prager materials (kPa | psi | kPa).

harden

Hardening parameter for the linear strain hardening option (kPa | psi | kPa).

fricangle

Angle of internal friction for Mohr-Coulomb and Drucker-Prager materials (degrees).

biotscoef

Biot's coefficient.

dil_angle

Dilation angle (degrees) for Mohr-Coulomb and Drucker-Prager materials. The recommended range is $0 \leq dil_angle \leq fricangle$. Specifying $dil_angle > fricangle$ may lead to numerical instability. When $dil_angle = fricangle$ the results of non-associated flow and associated flow must be the same.

For some materials such as soil or sand, the use of friction angle alone (associated flow rule) may lead to a large discrepancy in computing the volumetric strain. For such materials, dilation angle *dil_angle* is used instead to obtain more accurate results which are comparable to lab data. Generally,

the non-associated flow rule is based on the plastic potential function whereas the associated flow rule is based on the yield function.

Since the stiffness matrix generated by the non-associated flow rule is unsymmetrical, an appropriated solver must be used (see *SOLVERG).

DEFAULTS:

If the following keywords are absent, the indicated action is taken.

Absent	**Action**
*YLDSTRESS	*yldstress* = 0
*COHESION	*cohesion* = 0
*HARDEN	*harden* = 0
*FRICANGLE	*fricangle* = 30
*BIOTSCOEF	*biotscoef* = 1
*DILANGLE	*dil_angle* = *fricangle*

CONDITIONS:

Both *ELASTMOD and *POISSRATIO are required if the plastic deformation option is used. See the *GEOMECH keyword. The other keywords are optional.

These keywords are rock-type dependent, and are applied to the rock type number in effect at the time they are read. See the *GEOROCK and *GEOTYPE keywords.

EXPLANATION:

When the Mohr-Coulomb or the Drucker-Prager yield criteria are used, the relevant keywords are:

*ELASTMOD *POISSRATIO *COHESION *FRICANGLE *HARDEN *BIOTSCOEF

Mohr-Coulomb and Drucker-Prager are popular yield criteria for geological materials. By default, the material is assumed to be cohesionless, with a friction angle of 30 degree, a good number to use for sand, and a perfectly plastic behavior upon yielding with no strain hardening. Biot's coefficient is one, representing full interaction between pore pressure and stress.

When the Von Mises or the Tresca yield criteria are used, the relevant keywords are:

*ELASTMOD *POISSRATIO *YLDSTRESS *HARDEN *BIOTSCOEF

Von Mises and Tresca are popular yield criteria for metal plasticity. Due to their simplicity, they are included here mainly for testing purposes.

If the elasticity is applied to a rock type, only three main keywords are needed such as: *ELASTMOD, *POISSRATIO, *YLDSTRESS. In order that the rock behaves elasticity through the course of loading, the yield stress (*YLDSTRESS) must be a very large value.

Temperature-dependent Properties *GRTEMTAB

PURPOSE:

Specify temperature dependence of geomechanical properties.

FORMAT:

*GRTEMTAB $key_1 \ldots key_n$
{ T $value_1 \ldots value_n$ }

DEFINITIONS:

*GRTEMTAB $key_1 \ldots key_n$
Specify which geomechanical properties will vary with temperature. Table columns are defined by one or more of the following sub-keywords, in any order. Each sub-keyword may be used at most once.

*ELASTMOD	Young's modulus (kPa | psi | kPa)
*POISSRATIO	Poisson's ratio (dimensionless)
*BIOTSCOEFF	Biot's coefficient (dimensionless)
*COHESION	Cohesion (kPa | psi | kPa)
*FRICANGLE	Friction angle (degrees)
*THEXPCOEF	Linear thermal expansion coefficient (1/C | 1/F | 1/C)

{ T $value_1 \ldots value_n$ }
Specify table consisting of a column of temperature values T plus a column of values for each quantity defined by the ordered sub-keyword list $key_1 \ldots key_n$. Temperatures T must be monotonically increasing down the table. The minimum allowed number of rows is 2.

DEFAULTS:

If *GRTEMTAB is absent for a rock type, formation geomechanical properties for that rock type do not depend on temperature.

If *GRTEMTAB appears but a sub-keyword is absent, the corresponding value is obtained from the primary keyword of the same name (or its default). For example, if *GRTEMTAB appears without sub-keyword *ELASTMOD, the constant Young's modulus is obtained from primary keyword *ELASTMOD.

CONDITIONS:

Keyword *GRTEMTAB is available on a rock-type basis (see *GEOROCK).

*GRTEMTAB is not available together with the Pseudo-Dilation Model.

EXPLANATION:

Significant temperature variations can be found in a number of oil recovery methods. In these cases geomaterials can experience changes in property values, which will affect geomechanical responses as well as fluid flow. The properties found in the lab to have significant temperature dependence are Young's modulus, Poisson's ratio, cohesion, Biot's coefficient, friction angle and thermal expansion coefficients. Use of *GRTEMTAB can help improve a history match when temperature plays an important role in the recovery process.

Example:

The following is an example using SI units:

```
*GRTEMTAB  *ELASTMOD  *COHESION  *POISSRATIO
** temp (C)   (kPa)      (kPa)
       50     1.0e8       200       0.295
      250     0.5e8       190       0.3
      450     0.1e8       185       0.303
```

Data Sources

Temperature dependent data of these properties can be obtained from handbooks or papers that deal with rock properties. However, a special or unique rock type may require a tailored lab test in order to determine physical properties related to a specific field study. The following references may help you understand the trends of geomaterial properties with changing temperature.

References:

Koca, M.Y et al., :Changes in the Engineering Properties of Marble in Fire-exposed Columns", Int. J. Rock Mechanics & Mining Sciences, Vol. 43, pp. 520-530 (2006).

Heuze, F.E., "High-temperature Mechanical, Physical and Thermal Properties of Granitic Rocks – A Review", Int. J. Rock Mech. Min. Sci. & Geomech. Abstract, Vol. 20, pp. 3-20 (1983)

Madland, M.V., Korsenes, R.I. and Risnes, R., "Temperature Effects in Brazilian, Uniaxial and Triaxial Compressive Tests with High Porosity Chalk", SPE 77761 (2002).

Porosity-Dependent Properties *GRPORTAB

PURPOSE:

Specify formation geomechanical properties that depend on porosity.

FORMAT:

*GRPORTAB $key_1 \ldots key_n$
{ φ_f $value_1 \ldots value_n$ }

DEFINITIONS:

*GRPORTAB $key_1 \ldots key_n$

Specify which geomechanical properties will vary with porosity. Table columns are defined by one or more of the following sub-keywords, in any order. Each sub-keyword may be used at most once.

*ELASTMOD	Young's modulus (kPa | psi | kPa)
*POISSRATIO	Poisson's ratio (dimensionless)
*BIOTSCOEFF	Biot's coefficient (dimensionless)
*COHESION	Cohesion (kPa | psi | kPa)
*FRICANGLE	Friction angle (degrees)
*THEXPCOEF	Linear thermal expansion coefficient (1/C | 1/F | 1/C)

{ φ_f $value_1 \ldots value_n$ }

Specify table consisting of a column of porosity values φ_f plus a column of values for each quantity defined by the ordered sub-keyword list $key_1 \ldots key_n$. Porosities φ_f must be monotonically increasing down the table. The minimum allowed number of rows is 2.

DEFAULTS:

If *GRPORTAB is absent for a rock type, formation geomechanical properties for that rock type do not depend on porosity.

If *GRPORTAB appears but a sub-keyword is absent, the corresponding value is obtained from the primary keyword of the same name (or its default). For example, if *GRPORTAB appears without sub-keyword *ELASTMOD, the constant Young's modulus is obtained from primary keyword *ELASTMOD.

CONDITIONS:

Keyword *GRPORTAB is available on a rock-type basis (see *GEOROCK).

*GRPORTAB is not available together with the Pseudo-Dilation Model.

EXPLANATION:

Keyword *GRPORTAB allows you to define geomechanical properties that change with porosity. The properties specified by primary keywords *ELASTMOD, etc., normally are measured in the laboratory at constant porosity. However, porosity changing due to injection or production in the field can affect geomechanical properties.

The porosity used by *GRPORTAB is void porosity, which is the fraction of gross volume not occupied by rock.

Example:

The following is an example using SI units:

```
*GRPORTAB  *ELASTMOD  *COHESION  *POISSRATIO  *FRICANGLE
** porosity
     0.01     1e8        200         0.29         25
     0.1     0.5e8       190         0.3          30
     0.2     0.1e8       185         0.32         35
```

References:

Li, L. and Aubertin, M., "A general relationship between porosity and uniaxial strength of engineering materials," *Can. J. Civ. Eng.*, Vol. 30, pp. 644-658, 2003.

Phani, K.K. and Sanyal, D., "Critical reevaluation of the prediction of effective Poisson's ratio for porous materials," *J. Materials Science*, Vol. 40, pp. 5685-5690, 2005.

Pressure-Dependent Properties *GRPRESTAB

PURPOSE:

Specify formation geomechanical properties that depend on fluid pore pressure.

FORMAT:

*GRPRESTAB $key_1 \ldots key_n$
{ *pres* $value_1 \ldots value_n$ }

DEFINITIONS:

*GRPRESTAB $key_1 \ldots key_n$

Specify which geomechanical properties will vary with pore pressure. Table columns are defined by one or more of the following sub-keywords, in any order. Each sub-keyword may be used at most once.

*ELASTMOD	Young's modulus (kPa | psi | kPa)
*POISSRATIO	Poisson's ratio (dimensionless)
*BIOTSCOEFF	Biot's coefficient (dimensionless)
*COHESION	Cohesion (kPa | psi | kPa)
*FRICANGLE	Friction angle (degrees)
*THEXPCOEF	Linear thermal expansion coefficient (1/C | 1/F | 1/C)

{ *pres* $value_1 \ldots value_n$ }

Specify table consisting of a column of pore pressure values plus a column of values for each quantity defined by the ordered sub-keyword list $key_1 \ldots key_n$. The pore pressure must be monotonically increasing down the table. The minimum allowed number of rows is 2.

DEFAULTS:

If *GRPRESTAB is absent for a rock type, formation geomechanical properties for that rock type do not depend on fluid pore pressure.

If *GRPRESTAB appears but a sub-keyword is absent, the corresponding value is obtained from the primary keyword of the same name (or its default). For example, if *GRPRESTAB appears without sub-keyword *ELASTMOD, the constant Young's modulus is obtained from primary keyword *ELASTMOD.

CONDITIONS:

Keyword *GRPRESTAB is available on a rock-type basis (see *GEOROCK).

*GRPRESTAB is not available together with the Pseudo-Dilation Model.

EXPLANATION:

Keyword *GRPRESTAB allows you to define geomechanical properties that change with fluid pore pressure. The properties specified by primary keywords *ELASTMOD, etc., normally are measured in the laboratory at constant pressure. However, fluid pressure changing due to injection or production in the field can affect geomechanical properties.

Example:

The following is an example using field units:

```
*GRPRESTAB  *ELASTMOD  *POISSRATIO  *COHESION  *FRICANGLE
**Pressure  Young       Poisson     Cohes     Angle (degrees)
0            3.50E+04     0.359          22         16
500          3.20E+04     0.358          20         17
1000         3.00E+04     0.357          18         18
1500         2.70E+04     0.356          17         19
2000         2.60E+04     0.355          16         20
2500         2.40E+04     0.354          15         21
3000         2.20E+04     0.353          14.5 22
3500         2.00E+04     0.352          14         24
4000         1.95E+04     0.35           13         25
4500         1.90E+04     0.345          12         26
5000         1.80E+04     0.34           11.5 27
5500         1.70E+04     0.336          11         28
6000         1.60E+04     0.332          10.5 29
7500         1.50E+04     0.32           10         30
8000         1.40E+04     0.315          9          32
8500         1.30E+04     0.31           8          34
9000         1.20E+04     0.305          7          36
9500         1.10E+04     0.3            6          38
10000        1.00E+04     0.3            5          40
```

Stress-Dependent Properties *GRSTRESTAB, *GRMPRSTAB

PURPOSE:

Specify formation geomechanical properties that depend on effective mean stress or minimum principal effective stress.

FORMAT:

*GRSTRESTAB $key_1 \ldots key_n$
{ *stres* $value_1 \ldots value_n$ }

*GRMPRSTAB $key_1 \ldots key_n$
{ *stres* $value_1 \ldots value_n$ }

DEFINITIONS:

*GRSTRESTAB
Specify that geomechanical properties will vary with effective mean stress.

*GRMPRSTAB
Specify that geomechanical properties will vary with minimum principal effective stress.

$key_1 \ldots key_n$
Specify which geomechanical properties will vary with stress. Table columns are defined by one or more of the following sub-keywords, in any order. Each sub-keyword may be used at most once.

*ELASTMOD	Young's modulus (kPa | psi | kPa)
*POISSRATIO	Poisson's ratio (dimensionless)
*BIOTSCOEFF	Biot's coefficient (dimensionless)
*COHESION	Cohesion (kPa | psi | kPa)
*FRICANGLE	Friction angle (degrees)
*THEXPCOEF	Linear thermal expansion coefficient (1/C | 1/F | 1/C)

{ *stres* $value_1 \ldots value_n$ }
Specify table consisting of a column of stress values plus a column of values for each quantity defined by the ordered sub-keyword list $key_1 \ldots key_n$. The effective mean stress must be monotonically increasing down the table. The minimum allowed number of rows is 2.

DEFAULTS:

If *GRSTRESTAB and *GRMPRSTAB are absent for a rock type, formation geomechanical properties for that rock type do not depend on stress.

If *GRSTRESTAB or *GRMPRSTAB appears but a sub-keyword is absent, the corresponding value is obtained from the primary keyword of the same name (or its default). For example, if *GRSTRESTAB or *GRMPRSTAB appears without sub-keyword *ELASTMOD, the constant Young's modulus is obtained from primary keyword *ELASTMOD.

CONDITIONS:

Keywords *GRSTRESTAB and *GRMPRSTAB are available on a rock-type basis (see *GEOROCK).

*GRSTRESTAB and *GRMPRSTAB are not available together with the Pseudo-Dilation Model.

Both *GRSTRESTAB and *GRMPRSTAB cannot be used together in the same rock type.

EXPLANATION:

Keyword *GRSTRESTAB or *GRMPRSTAB allows you to define geomechanical properties that change with effective mean stress or minimum principal effective, respectively. The properties specified by primary keywords *ELASTMOD, etc., normally are measured in the laboratory at constant stresses. However, stresses changing due to injection or production in the field can affect geomechanical properties.

Example:

The following is an example using field units for keyword *GRSTRESTAB:

```
*GRSTRESTAB  ELASTMOD  POISSRATIO  COHESION  FRICANGLE
** Stress   Young       Poisson     Cohes   Angle (degrees)
     0       1.0e4        0.3         10       40
   500       1.1e4  0.3         11       38
  1000       1.2e4        0.305       12       36
  1500       1.3e4  0.31        13       34
  2000       1.4e4        0.315       14       32
  2500       1.5e4  0.32        15       30
  3000       1.6e4        0.332       17       29
  3500       1.7e4        0.336       18       28
  4000       1.8e4        0.34        20       27
  4500       1.9e4        0.345       21       26
  5000 1.95e4 0.35         22       25
  5500 2.0e4        0.352       23       24
  6000 2.2e4        0.353       25       22
  7500 2.4e4        0.354       27       21
  8000 2.6e4        0.355       28       20
  8500 2.7e4        0.356       29       19
  9000 3.0e4        0.357       30       18
  9500 3.2e4        0.358       32       17
 10000       3.5e4  0.359       33       16
```

STARS: Solid Component Properties *GEOSOLID, *SDELASTMOD, *SDPOISSON, *SDCOHES, *SDFRICANG, *SDBIOTSCF, *SDTHEXPCF, *SDILANG

PURPOSE:

Assign geomechanical properties for a STARS solid component.

FORMAT:

*GEOSOLID	'*namec*'
*SDELASTMOD	*elastmod*
*SDPOISSON	*poissratio*
*SDCOHES	*cohesion*
*SDFRICANG	*fricangle*
*SDBIOTSCF	*biotscoef*
*SDTHEXPCF	*thexpcoef*
*SDILANG	*dil_angle*

DEFINITIONS:

*GEOSOLID '*namec*'

Name of solid component, in quotes. This component name must be one of the solid components specified via keyword *COMPNAME. This keyword must be the first of this keyword group and must appear at most once for each *namec*.

elastmod

Young's modulus of the solid component (kPa | psi | kPa).

poissratio

Poisson's ratio of the solid component (dimensionless).

cohesion

Cohesion of the solid component (kPa | psi | kPa). Use only with the Mohr-Coulomb or the Drucker-Prager yield criteria.

fricangle

Friction angle of the solid component (degrees).

biotscoef

Biot's coefficient of the solid component (dimensionless).

thexpcoef

Linear thermal expansion coefficient of the solid component (1/C | 1/F | 1/C).

dil_angle

Dilation angle of the solid component (degrees). See the description for keyword *DILANGLE.

DEFAULTS:

If *GEOSOLID appears for a component but the following keywords are absent, the indicated action is taken.

Absent	**Action**
*SDCOHES	*cohesion* = 0
*SDFRICANG	*fricangle* = 30
*SDBIOTSCF	*biotscoef* = 1
*SDTHEXPCF	*thexpcoef* = 0
*SDILANG	*dil_angle* = *fricangle*

CONDITIONS:

If *GEOSOLID appears for a component, then *SDELASTMOD and *SDPOISSON are required for that component.

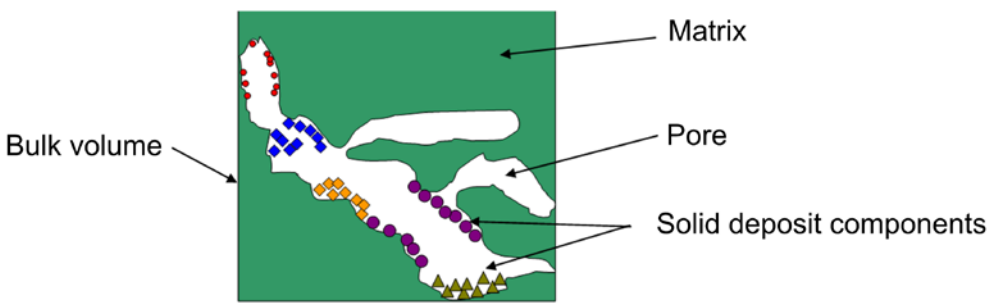
These keywords are independent of rock type defined by *GEOROCK or its default.

This keyword group must appear once for each solid component for which geomechanical properties are assigned.

EXPLANATION:

Solid component properties are defined in both the fluid-flow and geomechanical data sections. In the fluid-flow section (Component Properties), those properties relate to volume (density, blockage), energy (heat capacity) and chemical reactions. In the geomechanical data section, solid component properties involve mechanical deformation (Young's modulus, Poisson's ratio, cohesion, etc.). In this case the geomechanical properties of a grid block depend not only on the formation material properties but also on the amount of each solid component present.

Volume fractions of the solid components are obtained directly from the flow simulation, potentially different for each grid block. Depending on its volume fraction and its geomechanical properties, a solid component can have a strong or weak influence on the deformation response.



Bulk volume of a grid block can be considered as a sum of volumes of matrix (rock), fluid and solid components.

$$V_b^0 = V_m + V_f + \sum_{i=1}^{n} V_s^i \qquad (1)$$

Dividing both sides of (1) by V_b^0 gives:

$$1 = \phi_m + \phi_f + \sum_{i=1}^{n} \phi_s^i \qquad (2)$$

Let: $\phi_b = \phi_m + \phi_f$ (3)

Substituting (3) into (2) leads to:

$$\phi_b = 1 - \sum_{i=1}^{n} \phi_s^i \qquad (4)$$

Property χ of a grid block can be estimated as:

$$\chi = \left\{ \phi_b (\chi_b)^m + \left(\sum_{i=1}^{n} \phi_i^s (\chi_i)^m \right) \right\}^{1/m} \qquad (5)$$

where:

χ	Geomechanical property of a grid block
χ_i	Geomechanical property of a solid deposit component i
χ_b	Geomechanical property of matrix and fluid
V_m	Volume of matrix
V_f	Volume of fluid
V_s^i	Volume of solid deposit component i
$\phi_m = \frac{V_m}{V_b^0}$	Volume fraction of matrix
$\phi_f = \frac{V_f}{V_b^0}$	Volume fraction of fluid
$\phi_s^i = \frac{V_s^i}{V_b^0}$	Volume fraction of a solid deposit component i
ϕ_b	Volume fraction of matrix and fluid
V_b^0	Initial bulk volume of a grid block
n	Number of solid deposit components
m	Scaling power

In the above equation (5), the property χ_b can be constant or a function of volume fraction of fluid and it will be determined by porosity-dependent properties (*GRPORTAB).

Example:

Consider a sand-flow simulation where sand is mobilized, flows entrained in the fluid stream and then deposits again. Entrained sand "FREE_SAN" does not contribute to formation strength and is modelled with a fluid flow equation. The original sand is called "SAND1" and the settled sand is called "SAND2". These three sands have the same volumetric and thermal properties. Reactions change one kind of sand into another kind in response to local conditions. The Component Properties would be:

```
** Fluid component density data
*COMPNAME 'WATER' 'OIL' 'SOLN_GAS' 'FREE_SAN' 'SAND1' 'SAND2'
**         -----   ---   --------   --------   -----   -----
*CMM        0     0.508   0.016       1.150   1.150   1.150
*MASSDEN    0      985     320        2650
*CP         0     82E-8  1.45E-6     1.0E-7
*CT1        0      7E-4   810E-6     205E-6

** Solid component density data
*SOLID_DEN  'SAND1'   2650.0E-0   1.0E-7   205E-6
*SOLID_DEN  'SAND2'   2650.0E-0   1.0E-7   205E-6

** Reactions linking 'FREE_SAN', 'SAND1' and 'SAND2'
...
```

In the Geomechanics data section is found:

```
** For matrix (rock)
   *ELASTMOD    4.0E5   ** Young's modulus
   *POISSRATIO  0.3     ** Poisson's ratio
   *COHESION    15      ** Cohesion
   *FRICANGLE   40      ** Friction angle

** For original sand:
*GEOSOLID 'SAND1'
   *SDELASTMOD  4.0E6   ** Young's modulus
   *SDPOISSON   0.1     ** Poisson's ratio

** For settled-out sand:
*GEOSOLID 'SAND2'
   *SDELASTMOD  1.0E7   ** Young's modulus
  *SDPOISSON   *SDPOISSON   0.2     ** Poisson's ratio
```

Yield Criterion *TRESCA, *VONMISES, *MOHRCOUL, *DRUCKER

PURPOSE:

Assign yield criterion.

FORMAT:

*MOHRCOUL | *DRUCKER | *TRESCA | *VONMISES

DEFINITIONS:

*MOHRCOUL
Mohr-Coulomb yield criterion is used.

*DRUCKER
Drucker-Prager yield criterion is used.

*TRESCA
Tresca yield criterion is used.

*VONMISES
Von Mises yield criterion is used.

DEFAULTS:

*MOHRCOUL

CONDITIONS:

Only one of the yield criteria may be in use, i.e. they are mutually exclusive.

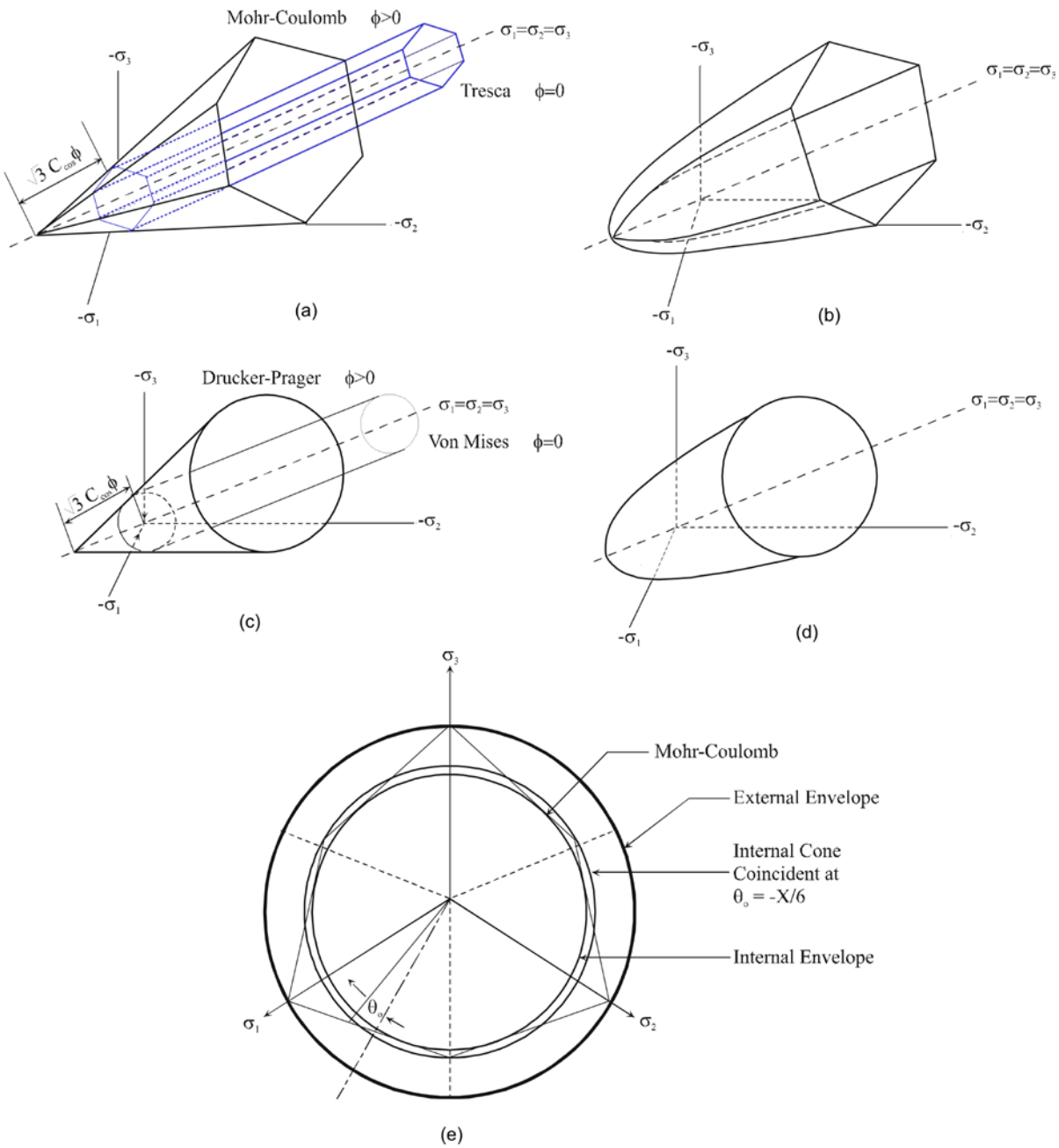
EXPLANATION:

A pictorial view of these yield criteria in the three dimensional stress space are shown in Figure below.

These yield surfaces prescribe the stress states below which the material will behave elastically. As the material strain hardens, the yield surface expands as a function of the accumulated plastic strains. Presently, the model assumes isotropic hardening where the expansion of the yield surface is symmetric about the axis of the yield cone.

The Mohr-Coulomb and the Drucker-Prager yield criteria are popular criteria for frictional porous material. They are typically used for analysis of soils and rocks.

The Tresca and the Von Mises criteria are popular for metal plasticity and are available here mainly for testing.



General and Linearized Yield Surfaces in Principal-Stress Space (from Desai & Christian, 1977)

Cap Model *GCAPMODEL

PURPOSE:

Using cap model for an elasto-plastic constitutive model.

FORMAT:

*GCAPMODEL *nmodel*

DEFINITION:

*GCAPMODEL
Specify a cap model for a material.

nmodel
Indicate a specific cap model used in computation.

CONDITIONS:

When the keyword *GCAPMODEL appears, others keywords related to a cap model must be accompanied with it.

EXPLANATION:

For soils, the Mohr-Coulomb criterion and Drucker-Prager criterion both suffer a deficiency which is the material can support an unlimited hydrostatic compression, This deficiency can be removed by adding a cap model which acts as a yield surface in the criteria. The cap model would allow consideration of compressive plastic volumetric strains and limit the amount of plastic dilation that occurred when loading on the Mohr-Coulomb or Drucker-Prager failure surface. The cap model is allowed to expand and contract along the hydrostatic axis. Details of a cap model will be given in the next section.

Cap Model 1 *GCAPLOC, *GCAPD, *GCAPW, *GCAPR, *GCAPTEN, *GCAPMAT, *ELASTMOD, *POISSRATIO, *COHESION, *FRICANGLE, *GCINCRMT, *GCAPMOD

PURPOSE:

Define coefficients for a cap model #1 as described in the keyword *GCAPMODEL.

FORMAT:

*GCAPLOC	κ
*GCAPD	*D*
*GCAPW	*W*
*GCAPR	*R*
*GCAPTEN	*tension*
*GCAPMAT	*type*
*ELASTMOD	*elastmod*
*POISSRATIO	*poissratio*
*COHESION	*cohesion*
*FRICANGLE	*fricangle*
*GCINCRMT	*segment*
*GCAPMOD	

DEFINITION:

κ
: Initial hardening parameter (kPa | psi | kPa)

D
: Cap material constant (1/kPa | 1/psi |1/kPa)

W
: Cap material constant (dimensionless)

R
: Aspect ratio (≥ 0) of the ellipse (ratio of horizontal to vertical axes)

tension
: Tension cut-off limit strength (kPa | psi | kPa)

type
: = 1 for soil whose hardening surface is allowed to move back toward the origin

: = 2 for rock whose hardening surface is not allowed to move back

elastmod
: Young's elastic modulus (kPa | psi | kPa)

poissratio
: Poisson's ratio

cohesion

Cohesion for Drucker-Prager model (kPa | psi | kPa)

fricangle

Friction angle for Drucker-Prager model (degrees)

segment

Number of divisions for strain

*GCAPMOD

Modify Young's modulus on the cap hardening surface.

DEFAULTS

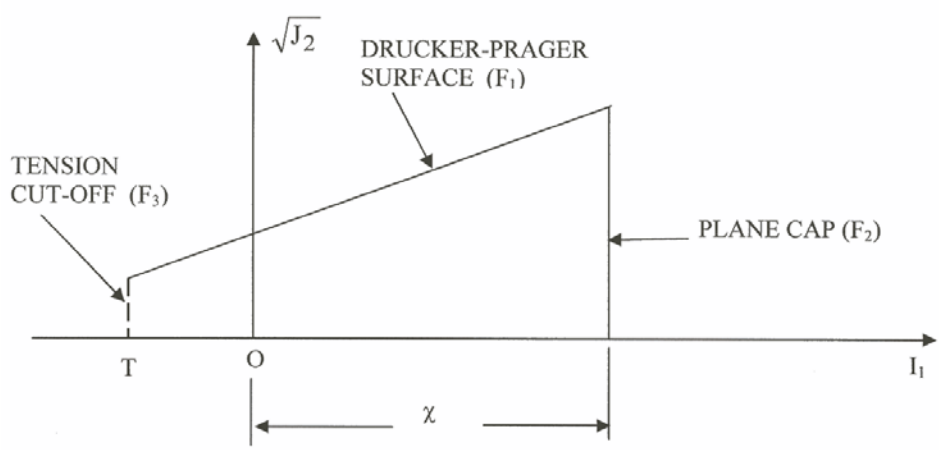
```
*COHESION      0
*FRICANGLE    30
*GCAPR         0
*GCAPTEN       0
*GCAPLOC       0
*GCAPMAT       2
*GCINCRMT      1
*GCAPMOD       No modification
```

CONDITION:

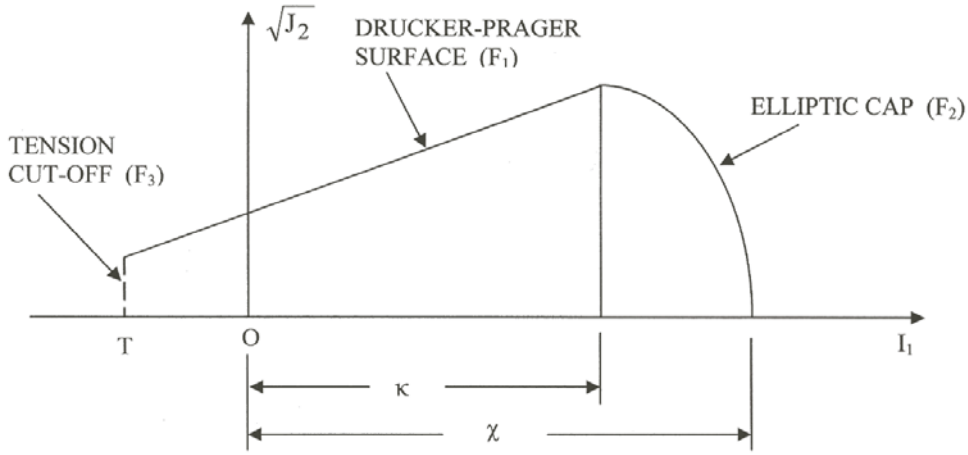
Accept the above default properties; other properties of the cap model must be given.

EXPLANATION:

This cap model # 1 is only used for a perfectly plastic material and is coupled with the Drucker-Prager model. The cap surface can be a plane cap (Bath et al. and Sandler et al.) or an elliptic cap (Chen and Mizuno) depending on the value of shape ratio R. If R equals zero, the plane cap is used otherwise the elliptic cap is used.



Compressive Plane Cap Models in I1 and $\sqrt{J_2}$ Plane



Compressive Elliptic Cap Models in I_1 and $\sqrt{J_2}$ Plane

where:

I_1: First invariance of effective stresses

J_2: Second invariance of effective deviatoric stresses

As seen in the above figures, the failure surfaces are consisted of three different surfaces such as Drucker-Prager failure surface expressed by function F_1, cap failure surface expressed by function F_2 and tension cut-off plane expressed by function F_3. It should be mentioned that this cap model is only applied to perfectly plastic materials and coupled to Drucker-Prager failure criterion. Also, compressive stress is positive and tensile stress is negative.

In the above figure, there are three failure surfaces as follows:

Drucker-Prager failure surface F_1 is defined by:

$$F_1 = \alpha I_1 + \sqrt{J_2} - k = 0$$

Where α and k are material parameters related to the cohesion and the friction angle.

Cap failure surface F_2 is introduced by:

Compression plane cap surface:

$$F_2 = I_1 - \chi = 0$$

Compression elliptic cap surface:

$$F_2 = \left[I_1 - L_{(\kappa)}\right]^2 + R^2 J_2 - \left[\chi - L_{(\kappa)}\right]^2 = 0$$

where:

$L_{(\kappa)} = \kappa$ for $\kappa > 0$ and $L_{(\kappa)} = 0$ for $\kappa \leq 0$

χ is the cap location dependent on the plastic volumetric strain ε_{kk}^{p} and is assumed to be:

$$\chi = \frac{1}{D}\mathbf{ln}\left(1 + \frac{\varepsilon_{kk}^{p}}{W}\right)$$

D and W are cap material constants which are defined by keywords *GCAPD and *GCAPW, respectively.

κ and R are and the hardening parameter of the cap and aspect ratio of the ellipse as defined by the keywords *GCAPLOC and *GCAPR, respectively.

Tension cut-off limit plane F_3 is given by:

$$F_3 = I_1 - T = 0$$

Details of this cap model # 1 and related equation can be reviewed in the references.

In addition, the keyword *GCINCRMT is used to improve the accuracy and convergence. When strain is large, it can be divided into many segments (strain/segment) so that the stress return algorithm converges easily.

When keyword *GCAPMOD appears, the Young's modulus on the cap hardening surface will be modified as:

$$E_{cap} = \frac{3(1-2\nu)}{\frac{3(1-2\nu)}{E} + 3D\left(\varepsilon_{kk}^{p} + W\right)}$$

where E is the initial Young's modulus (*ELASMOD) and ν is Poisson's ratio.

Example:

In the geomechanical section of a data set, this cap model is used.

```
*GCAPMODEL   1        ** Cap model type 1
*ELASTMOD    40.3E+3  ** Young modulus (kPa)
*POISSRATIO  0.2736   ** Poisson ratio
*FRICANGLE   49.093   ** Friction angle (degrees)
*COHESION    10.0     ** Cohesion (kPa)
*GCAPR       0.0      ** plane cap
*GCAPD       1.42E-6  ** D value (1/kPa)
*GCAPW       0.0075   ** W value
*GCAPTEN     -5.0     ** Tension cut-off (kPa)
*GCAPLOC     100      ** kPa, N/A when plane cap is used
*GCINCRMT    5        ** Number of strain divisions
*GCAPMOD              ** Modify Young's modulus on hardening
```

References:

Bath, K.J., Snyder, M.D., Cimento, A.P. and Rolph, W.D.: "*On Some Current Procedures and Difficulties in Finite Element Analysis of Elastic-Plastic Response*", Comput. Struct., Vol. 12, pp. 607-624, 1980.

Sandler, I.S., DiMaggio, F.L. and Baladi, G.Y.: "*Generalized Cap Model for Geological Materials*", J. Geotech. Eng. Dic., ASCE, Vol. 102 (GT7), pp. 683-699, 1976.

Chen, W.F. and Mizuno, E.: *Nonlinear Analysis in Soil Mechanics: Theory and Implementation*, Elsevier, 1990.

Nonlinear Constitutive Model *NLINEAR

PURPOSE:

Assign a nonlinear constitutive model for a type of rock.

FORMAT:

*NLINEAR *nmodel*

DEFINITION:

*NLINEAR
Specify a nonlinear constitutive model for a material.

nmodel
Indicate a specific constitutive model used in computation.

CONDITIONS:

When the keyword *NLINEAR appears, others keywords related to a constitutive model must be accompanied with it.

EXPLANATION:

The material may have different aspects of behavior such as linear elasticity, elasto-plasticity, nonlinear elasticity or nonlinear plasticity. There are so many constitutive models which are used to express such nonlinear behavior of a material. However, only some of the most used models are selected to implement in the geomechanics module. Depending on the value of '*nmodel*' after the keyword *NLINEAR, a type of constitutive model is defined. For instance,

*NLINEAR 1 → Nonlinear elastic constitutive model 1.

*NLINEAR 2 → Nonlinear elastic constitutive model 2.

Details of each nonlinear constitutive model will be explained in an appropriate section containing coefficients related to the model.

Nonlinear Elastic Constitutive Model 1 *ELASTMOD, *POISSRATIO, *GAMMA, *GEXPONENTN, *GULBULKMOD

PURPOSE:

Define coefficients for a nonlinear elastic constitutive model # 1 as described in the keyword *NLINEAR.

FORMAT:

*ELASTMOD	*elastmod*
*POISSRATIO	*poissratio*
*GAMMA	*gamma*
*GEXPONENTN	*n*
*GULBULKMOD	*bulkmod*

DEFINITION:

elastmod
: Young's elastic modulus (kPa | psi)

poissratio
: Poisson's ratio.

gamma
: Coefficient multiplier (kPa | psi)$^{(1-n)}$ used to determine the bulk modulus at a given mean effective stress.

n
: Exponential power ‘n’ used to define nonlinearity of the bulk modulus.

bulkmod
: Bulk modulus (kPa | psi) used for unloading behavior.

DEFAULT

If the keyword *GULBULKMOD disappears, its value is zero.

CONDITION:

Young's modulus *ELASTMOD and Poisson's ratio *POISSRATIO must be given.

EXPLANATION:

In this constitutive model, the bulk modulus is defined as a function of mean effective stress under the loading condition. The bulk modulus at a certain mean effective stress can be written as:

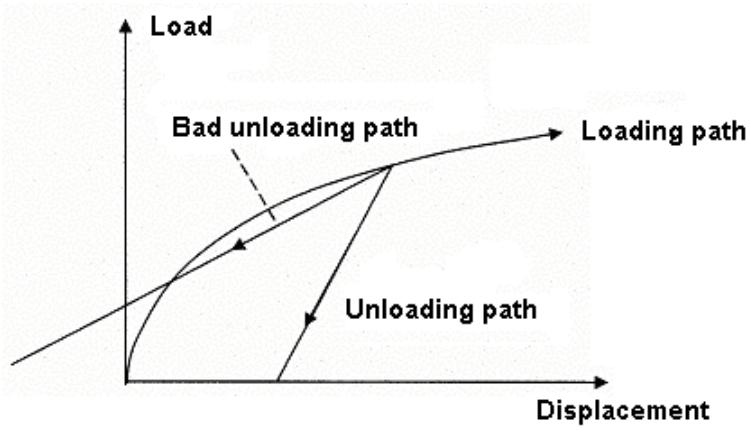
$$K = \text{gamma} * \left(\sigma'_m\right)^n$$

Where the coefficient ‘gamma’ and power ‘n’ are defined above, σ'_m is the mean effective stress which is given by:

$$\sigma'_m = \left(\sigma'_{11} + \sigma'_{22} + \sigma'_{33}\right)/3$$

In the above formula, $\sigma'_{11}, \sigma'_{22}, \sigma'_{33}$ are effective stresses in X, Y, and Z directions respectively.

When unloading occurs, the bulk modulus is assumed to be constant and has a value as described by the keyword *GULBULKMOD and its value (*bulkmod*). Care should be taken when assign a value of unloading bulk modulus so that the unloading path totally lies in the area bounded by the loading path and the displacement axis as shown. In other words when the unloading path is given, the value of *bulkmod* is selected carefully so that the unloading path will not cut the loading path. If the keyword *GULBULKMOD is not entered for a rock type, the unloading path is coincident to the loading path. The loading path and unloading path are distinguished by the work done criteria.



Example:

```
*GEOMECH
*NLINEAR 1           **Keyword and number for the hypoelastic model
*ELASTMOD   3.95E5   **(kPa | psi)
*POISSRATIO  0.3     **unitless
*GAMMA       6125.0  **(kPa | psi)**(1-n)
*GEXPONENTN  0.4     **unitless
*GULBULKMOD  2.5E+4  **(kPa | psi)
```

Nonlinear Elastic Constitutive Model 2 *ECOEF, *BCOEF, *GPATM, *FRATIO, *NE, *NB, *NTE, *NTB. *EXPN1, *EXPN2, *FRICANGMN, *FRICANGMX, *DFRICANGLE, *URECOEF, *URBCOEF, *URNE, *URNB, *URNTE, *URNTB, *UREXPN1, *UREXPN2, *COHESION, *FRICANGLE, *MCOEF, *FYOUNGAF, *POISSRATIO

PURPOSE:

Define coefficients for a hyperelastic constitutive model # 2 as described in the keyword *NLINEAR.

FORMAT:

*ECOEF	*Ke*
*BCOEF	*Kb*
*GPATM	*Patm*
*FRATIO	*Rf*
*NE	*ne*
*NB	*nb*
*NTE	*nte*
*NTB	*ntb*
*EXPN1	*n1*
*EXPN2	*n2*
*FRICANGMN	*phimin*
*FRICANGMX	*phimax*
*DFRICANGLE	*dphi*
*URECOEF	*Keur*
*URBCOEF	*Kbur*
*URNE	*neur*
*URNB	*nbur*
*URNTE	*nteur*
*URNTB	*ntbur*
*UREXPN1	*n1ur*
*UREXPN2	*n2ur*
*COHESION	*c*
*FRICANGLE	*phi*
*MCOEF	*m*
*FYOUNGAF	*factor*
*POISSRATIO	*poissratio*

DEFINITION:

Ke

Dimensionless loading modulus number

Kb

Dimensionless loading bulk modulus number

Patm

The atmospheric pressure (kPa | psi)

Rf

Failure ratio which lies between 0.01 and one

ne

Loading tangential modulus exponent for confining stress

nb

Loading bulk modulus exponent for confining stress

nte

Loading modulus exponent for temperature

ntb

Loading bulk modulus exponent for temperature

n1, n2

Loading tangential modulus exponents for failure ratio

phimin

Minimum friction angle (degrees)

phimax

Maximum friction angle (degrees)

dphi

Reduction in friction angle (degrees) for one log circle

Keur

Dimensionless unloading-reloading modulus number

Kbur

Dimensionless unloading-reloading bulk modulus number

neur

Unloading-reloading modulus exponent for confining stress

nbur

Unloading-reloading bulk modulus exponent for confining stress

nteur

Unloading-reloading modulus exponent for temperature

ntbur

Unloading-reloading bulk modulus exponent for temperature

n1ur, n2ur

Unloading-reloading modulus exponents for failure ratio

c

Cohesion for Mohr-Coulomb and Drucker-Prager materials (kPa | psi)

phi

Angle of internal friction for Mohr-Coulomb and Drucker-Prager materials (degrees) at a confining pressure of 1 atm.

m

Coefficient factor used to compute the reference stress. Its value is 0 or 1.

factor

Coefficient factor used to compute Young's modulus after failure. Its value is between 0.01 and 1.

poissratio

Force Poisson's ratio to be fixed at this value instead of recalculated.

Note: the confining stress used here corresponds to the minimum principle stress.

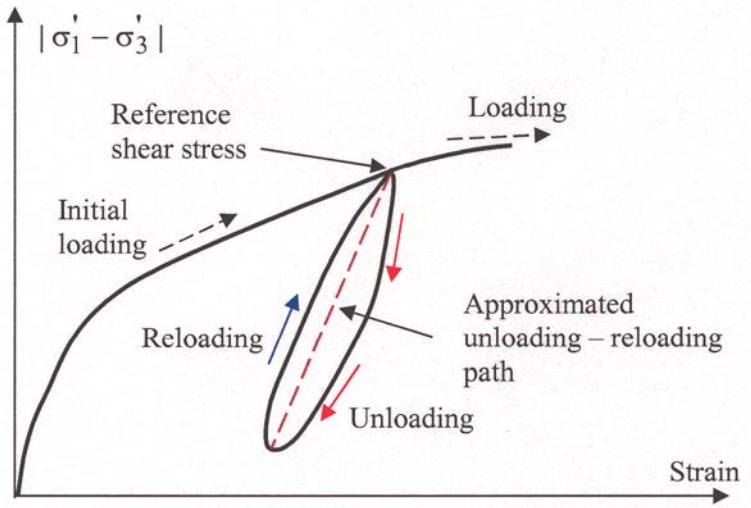
DEFAULT:

*GPATM	1 atm
*COHESION	0
*FRICANGLE	30
*NE, *NB	0.5
*URNE, *URNB	0.5
*EXPN1, *UREXPN1	2
*EXPN2, *UREXPN2	1
*NTE, *NTB	0
*URNTE, *URNTB	0
*FRICANGMN	0
*DFRICANGLE	0
*FRICANGLMX	30
*URECOEF	0
*URBCOEF	0
*MCOEF	1
*FYOUNGAF	0.5

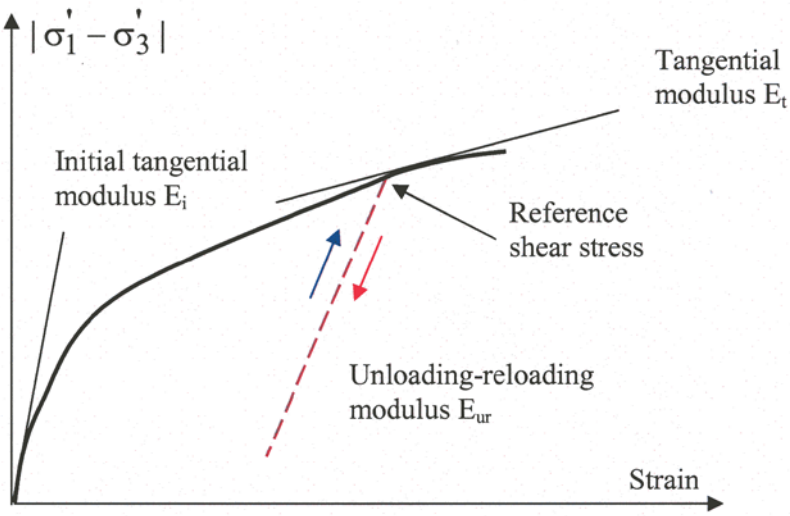
EXPLANATION:

The nonlinear elasticity constitutive model # 2 or hyperbolic model implemented in the module is based on the works of Duncan and Chang and Settari et al. In this model, tangential modulus as well as bulk modulus varies with minimum principle effective stress σ_3' and temperature. Poisson's ratio in this model is, thus, also varied. The model also has a loading path and an unloading-reloading path which are distinguished by the reference shear stress criteria. When keywords *URECOEF and *URBCOEF disappear, there is no loading path. In this case, the unloading path coincides with the loading path. Furthermore, in this model, the unloading and reloading paths are the same.

In this hyperbolic model, the Mohr-Coulomb failure criterion is used for the material failure due to shear stress to compute the stress level. The stress level is also limited by one so that the shear stress cannot exceed the shear failure of the Mohr-Coulomb model. More details on this model can be seen in related references.



The material follows a loading path when it is subjected to shear stress higher than it has previously experienced. Along this path, the constitutive model is governed by the tangential modulus. The material follows the unloading-reloading path when its shear stress is lower than the reference shear stress. In this case, the unloading-reloading modulus can either be affected by the shear stress level or not depending on the selection of the exponents related to the model.



In the above figure, there are three type of modulus:

Initial tangential modulus E_i: the slope of the tangential line passing the zero shear stress $|\sigma_1' - \sigma_3'| = 0$. The initial tangential modulus E_i is computed by:

$$E_i = K_e P_{atm} \left(\frac{\sigma_r}{P_{atm}}\right)^{ne} \left(\frac{T}{T_o}\right)^{nte}$$

$$\sigma_r = \sigma_3' + m \frac{c}{\tan(\text{phimax})}$$

where:

c, Ke, Patm, phimax, ne, nte, m are defined above by the related keywords.

T	Current temperature
T_o	Initial temperature
σ_r	Reference stress
$\sigma_3^{'}$	Minimum principle effective stress

When m = 0, the reference stress is the minimum principle effective stress which was proposed by Duncan and Chang.

Tangential modulus E_t:

$$E_t = E_i \left\{ 1 - \left[R_f \frac{\left(\sigma_1^{'} - \sigma_3^{'}\right)\left(1 - \sin(\varphi)\right)}{2c\cos(\varphi) + 2\sigma_3^{'}\sin(\varphi)} \right]^{n2} \right\}^{n1}$$

$$\varphi = phi - dphi * \log_{10}\left(\frac{\sigma_3^{'}}{Patm} \right)$$

where:

$E_i(\sigma_r)$: Initial tangential modulus defined in the above equation

R_f: Failure ratio defined by keyword *FRATIO

L_σ: Stress level which is limited to one when the deviatoric effective stress reaches to the failure envelope. Its value is determined by:

$$L_\sigma = \frac{\sigma_1^{'} - \sigma_3^{'}}{\left(\sigma_1^{'} - \sigma_3^{'}\right)_f} \le 1$$

Wherein, for sands, the failure envelope is normally defined by Mohr-Coulomb failure surface such as:

$$\left(\sigma_d^{'}\right)_f = \left(\sigma_1^{'} - \sigma_3^{'}\right)_f = \frac{2c\cos(\varphi) + 2\sigma_3^{'}\sin(\varphi)}{1 - \sin(\varphi)}$$

In the above equation, the friction angle φ for a given $\sigma_3^{'}$ is given by:

$$\varphi = phi - dphi * \log_{10}\left(\frac{\sigma_3^{'}}{Patm} \right)$$

and φ is limited within the range: phimax $\ge \varphi \ge$ phimin.

where:

R_f, c, n_1, n_2, phi, dphi, phimax, phimin are defined above.

If $n_1 = 2$ and $n_2 = 1$, the above equation has the same form as that of the Duncan and Chang model.

Since the stress level L_σ is limited by one, the hyperelastic model used in this module considers the Mohr-Coulomb failure envelope as its asymptote.

When the stress level L_σ is larger than 1, the material will fail. In such a case, the tangential Young's modulus E_f in the failure mode (*YLDSTATE = 1) may be expressed as:

$$E_f = K_e P_{atm} \left(\frac{\sigma_r}{P_{atm}} \right)^{ne} \left[factor \left(1 - \left(R_f \right)^{n2} \right) \right]^{n1}$$

Unloading-Reloading modulus E_{ur}

$$E_{uri} = K_{eur} P_{atm} \left(\frac{\sigma_r}{P_{atm}} \right)^{neur} \left(\frac{T}{T_o} \right)^{nteur}$$

$$E_{ur} = E_{uri} \left\{ 1 - \left[R_f \frac{\left(\sigma_1' - \sigma_3' \right)\left(1 - \sin(\varphi) \right)}{2c\cos(\varphi) + 2\sigma_3' \sin(\varphi)} \right]^{n2ur} \right\}^{n1ur}$$

where:

n1ur and *n2ur* are defined above.

If the value of *n2ur* is set to be zero, the unloading-reloading modulus is unaffected by the shear stress level.

If the material fails when reloading, the tangential Young's modulus is computed as:

$$E_{urf} = K_{eur} P_{atm} \left(\frac{\sigma_r}{P_{atm}} \right)^{neur} \left[factor \left(1 - \left(R_f \right)^{n2ur} \right) \right]^{n1ur}$$

Loading bulk modulus B_m

$$B_m = K_b P_{atm} \left(\frac{\sigma_r}{P_{atm}} \right)^{nb} \left(\frac{T}{T_o} \right)^{ntb}$$

Unloading bulk modulus B_{ur}

$$B_{ur} = K_{bur} P_{atm} \left(\frac{\sigma_r}{P_{atm}} \right)^{nbur} \left(\frac{T}{T_o} \right)^{ntbur}$$

Poisson ratio ν for loading or unloading-reloading case is computed by the equation:

$$\nu = \frac{3B - E}{6B}$$

where:

B	Bulk modulus
E	Tangential modulus

If you want Poisson's ratio to be fixed at a certain value instead of recalculated by the above formula, use keyword *POISSRATIO.

The computed Poisson ratio should lie between this range: $0.01 \le \nu \le 0.49$.

Example: For a rock type which has a nonlinear elastic constitutive model 2, its parameters are given as follows:

```
*GEOMECH
*NLINEAR       2     ** keyword and number for the
                        hyperelastic model
*ECOEF       485.0   ** unitless
*BCOEF       404.2   ** unitless
*GATM         14.7   ** ambient pressure (psi)
*FRATIO        0.85  ** fraction ratio (unitless)
*NE            0.3   ** unitless
*NB            0.3   ** unitless
*NTE           0.0   ** unitless
*NTB           0.0   ** unitless
*EXPN1         2.0   ** unitless
*EXPN2         1.0   ** unitless
*FRICANGMN     1.0   ** degrees
*FRICANGMX    60     ** degrees
*DFRICANGLE    0.1   ** degrees
*URECOEF     700     ** unit less
*URBCOEF     583.3   ** unit less
*URNE          0.6   ** unitless
*URNB          0.6   ** unitless
*URNTE         0.0   ** unitless
*URNTB         0.0   ** unitless
*UREXPN1       2.0   ** unitless
*UREXPN2       1.0   ** unitless
*COHESION     14.5   ** psi
*FRICANGLE    40     ** degrees
*MCOEF         0     ** unitless
```

References:

Settari, A., Ito, Y., Fukushima, N. and Vaziri, H.: "*Geotechnical Aspects of Recovery Processes in Oil Sands"*, Can. Geotech. J., Vol. 30, pp. 22-33, 1993.

Settari, A. and Walters, D.: "*Advances in Coupled Geomechanical and Reservoir Modeling With Application to Reservoir Compaction"*, SPE 51927, 1999.

Duncan, J.M. and Chang, C-Y.: "*Nonlinear Analysis of Stress and Strain in Soils"*, ASCE Journal of the Soil Mechanics and Foundations Division, Vol. 96, pp. 1629-1653, 1970.

Creep Model *GMCREEP

PURPOSE:

Assign a creep constitutive model for a type of rock. Creep is defined as the time-dependent strain that happens in a material at constant stress.

FORMAT:

*GMCREEP c*model*

DEFINITION:

*GMCREEP *cmodel*

Specify a creep constitutive model for a material, and indicate which plastic yield surface to use via *cmodel*.

*GMCREEP 1 → Drucker-Prager type

*GMCREEP 2 → Mises type

CONDITIONS:

Keyword *GMCREEP must be followed by others keywords related to the plastic yield surface indicated by *cmodel*. See manual page "Creep Model 1,2", below.

EXPLANATION:

There are many constitutive models used to express creep in materials. In CMG's geomechanics module, the elasto-viscoplastic constitutive model based on Perzyna's theory is selected. The total strain rate is decomposed into elastic strain rate and viscoplastic strain rate as follows:

$$\dot{\varepsilon} = \dot{\varepsilon}_e + \dot{\varepsilon}_{vp}$$

Here

$\dot{\varepsilon}_e$

Elastic strain rate estimated by the elastic constitutive relation.

$\dot{\varepsilon}_{vp}$

Viscoplastic strain rate based on Perzyna's theory wherein the viscous response is estimated by a time-rate flow rule. The viscoplastic strain rate is expressed by:

$$\dot{\varepsilon}_{vp} = \gamma \cdot \langle \Phi(F) \rangle \cdot \frac{\partial F}{\partial \sigma}$$

Here

γ

The fluidity parameter that defines the relative rate of viscoplastic strain.

$\langle \Phi(F) \rangle$

Flow function that determines the current magnitude of viscoplastic strain rate.

$$\langle \Phi(F) \rangle = \begin{cases} \Phi(F) & \text{For } F > 0 \\ 0 & \text{For } F \leq 0 \end{cases}$$

F

Plastic yield function.

Creep Model 1, 2 *VISPARA, *VISFLOWR, *VISPOWER, *VISSCHEME, *VISTIME, *VISSTEP, *VISINIT, *ELASTMOD, *POISSRATIO, *FRICANGLE, *COHESION, *YLDSTRESS

PURPOSE:

Define coefficients for creep models *GMCREEP 1 (Drucker-Prager yield surface) and *GMCREEP 2 (Mises yield surface).

FORMAT:

*VISPARA	*gamma*
*VISFLOWR	*function*
*VISPOWER	*delta*
*VISSCHEME	*scheme*
*VISTIME	*para*
*VISSTEP	*increment*
*VISINIT	*initime*
*ELASTMOD	*elastmod*
*POISSRATIO	*poissratio*
*FRICANGLE	*fricangle*
*COHESION	*cohesion*
*YLDSTRESS	*yldstress*

DEFINITION:

gamma

Fluidity parameter γ (1/days | 1/days | 1/mins).

function

Viscoplastic flow function type.

delta

Parameter δ used in viscoplastic flow function.

scheme

Scheme implicitness.

scheme = 1	explicit
scheme = 2	semi-implicit
scheme = 3	implicit

para

Time step increment parameter.

For explicit scheme	0.01 < *para* < 0.15
For implicit scheme	0.01 < *para* < 10

increment

Increase the value of time step from the previous time step.

$\Delta t_{n+1} = increment * \Delta t_n$

initime

Initial time-step length (day | day | min).

elastmod

Young's elastic modulus (kPa | psi | kPa).

poissratio

Poisson's ratio.

cohesion

Cohesion for **Drucker-Prager** materials (kPa | psi | kPa).

fricangle

Friction angle for **Drucker-Prager** materials (degrees).

ylstress

Yield stress for **von Mises** materials (kPa | psi).

DEFAULTS:

*COHESION	0
*FRICANGLE	30
*VISFLOWR	2
*VISPOWER	1.0
*VISSCHEME	3
*VISTIME	0.01
*VISSTEP	0.1
*VISINIT	1.0

CONDITIONS:

Young's modulus *ELASTMOD, Poisson's ratio *POISSRATIO and fluidity parameter *VISCOPARA must be given.

For a von Mises material, the yield stress (*YLDSTRESS) is used, whereas for a Drucker-Prager material cohesion (*COHESION) and friction angle (*FRICANGLE) must be used.

EXPLANATION:

This type of viscoplastic constitutive model has the following assumptions:

1. Viscoplastic flow occurs only when the yield function F is greater than a uniaxial yield stress F_0.
2. The viscoplastic strain rate depends upon the current stresses only.

Therefore the viscoplastic flow rule can be written as:

$$\dot{\varepsilon}_{vp} = \gamma \cdot \langle \Phi(F) \rangle \cdot \frac{\partial F}{\partial \sigma} \qquad (1)$$

where:

γ

Fluidity parameter controlling the plastic flow rate (*VISPARA).

The flow function Φ can be one of the forms:

function = 1:

$$\Phi(F) = \exp\left\{\delta\left(\frac{F - F_0}{F_0}\right)\right\} - 1 \qquad (2)$$

function = 2:

$$\Phi(F) = \left(\frac{F - F_0}{F_0}\right)^{\delta} \qquad (3)$$

In a time interval Δt equation (1) can be expressed as:

$$\Delta\varepsilon_{vp} = \Delta t_n \left\{(1-\eta)\dot{\varepsilon}_{vp}{}^{n} + \eta\dot{\varepsilon}_{vp}{}^{n+1}\right\} \qquad (4)$$

For the explicit scheme (*scheme* = 1) $\eta = 0$. For the semi-implicit scheme (*scheme* = 2) $\eta = 0.5$. For the implicit scheme (*scheme* = 3) $\eta = 1$.

When a value of η is greater than or equal to 0.5, the time integration scheme (4) is unconditionally stable; whereas the scheme (4) is only conditionally stable when the value of η is less than 0.5. For the latter case, the time-step length must be controlled. The time-step length can be computed as:

$$\Delta t_n \le \text{para} * \sqrt{\frac{\varepsilon_v{}^{n}}{\left(\dot{\varepsilon}_v{}^{n}\right)_{vp}}}$$

where

para:

Time increment parameter *para* as defined by the keyword *VISTIME.

ε_v:

Volumetric strain

$(\dot{\varepsilon}_v)_{vp}$:

Volumetric strain rate of viscoplasticity.

The change in time-step length between two consecutive calculations is also limited by:

$$\Delta t_{n+1} \le \text{Increment} * \Delta t_n$$

where

Increment:

the *increment* parameter as defined by the keyword *VISSTEP.

Note that the time-step length computed in the viscoplastic model is completely different from the time-step size used for fluid flow in the reservoir simulator. However, the sum of time-step length in the geomechanics module for one loading cannot exceed the flow time step. If the sum exceeds the flow time step, the steady state is not reached within the allowed time. A warning message in the file *.geo will appear such as "*Steady state time is larger than the flow time step"*. If such a case occurs, some parameters should be changed; for instance, fluidity parameter which mostly influents the plastic flow rate.

Example:

Creep model for a Drucker-Prager material in field units:

```
*GMCREEP          1           ** For Drucker-Prager material
*ELASTMOD         1.5E5       ** Young’s modulus (psi)
*POISSRATIO       0.3         ** Poisson’s ratio
*FRICANGLE        15.0        ** Friction angle (degrees)
*COHESION         14.5        ** Cohesion (psi)
*VISPARA          1.0E-5      ** Fluidity parameter (1/day)
*VISPOWER         0.01        ** Parameter used in flow function
*VISFLOWR         2           ** Flow function type
*VISSCHEME        1           ** Explicit scheme
*VISTIME          0.01        ** Time increment parameter
*VISSTEP          0.1         ** Time-step increment
*VISINIT          0.5         ** Initial time-step length (day)
```

References:

Perzyna, P.: "*Fundamental Problems in Viscoplasticity"*, Advances in Applied Mechanics, Academic, New York, Vol. 9, pp. 244-368, 1966.

Zienkiewicz, O.C. and Cormeau, I.C.:" *Visco-plasticity, Plasticity and Creep in Elastic Solids – A Unified Numerical Solution Approach"*, Int. Journal for Numerical Methods in Engineering, Vol. 8, pp. 821-845, 1974.

STARS: Pseudo Dilation Model *PGDILA, *PGPACT, *PGPDMAX

PURPOSE:

Use a pseudo dilation model to mimic the STARS *DILATION option.

FORMAT:

*PGDILA	*pgdila*	*young1*
*PGPACT	*pgpact*	*young2*
*PGPDMAX	*pgpdmax*	*young3*

DEFINITIONS:

pgdila
: Threshold pressure beyond which the material behavior changes from an elastic to a dilation state (kPa | psi).

young1
: Young's modulus in a dilation state (kPa | psi).

pgpact
: Threshold pressure below which the material behavior changes from elastic state to a recompaction state (kPa | psi).

young2
: Young's modulus in a recompaction state (kPa | psi).

pgpdmax
: Threshold pressure above which the material behavior changes from dilation state to an elastic state (kPa | psi).

young3
: Young's modulus in an elastic state beyond the dilation state (kPa | psi).

CONDITIONS:

These keywords are assigned to the current rock type specified via *GEOROCK, or to rock type #1 if *GEOROCK is absent.

To use the pseudo dilation model the minimum required data is *PGDILA and *PGPACT. Also required are *ELASTMOD (Young's modulus), *POISSRATIO (Poisson's ratio) and a very large value for *YLDSTRESS (yield stress) so the material is always in elastic mode.

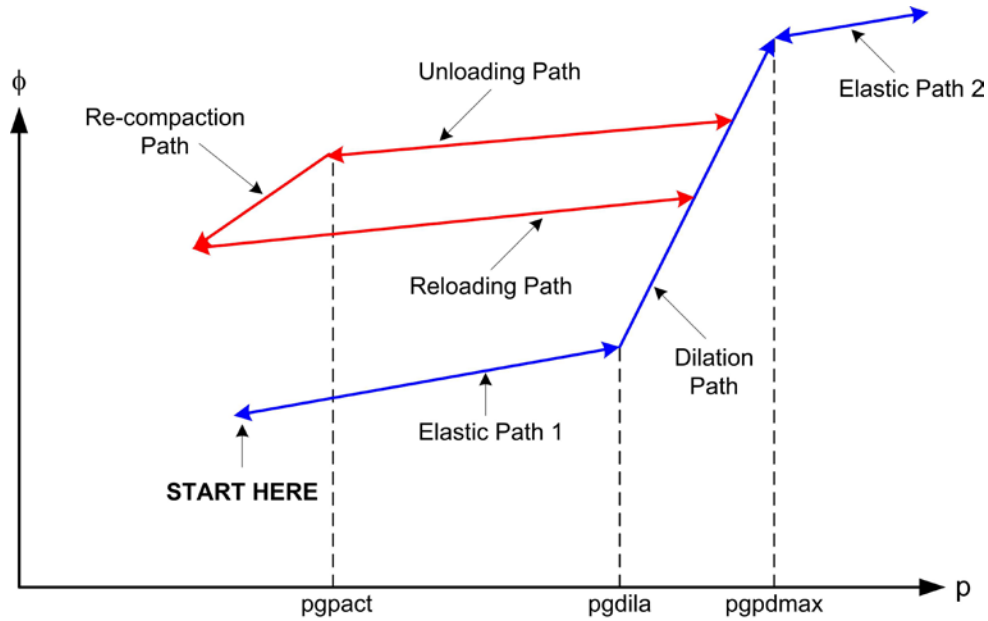
pgpdmax must be greater than *pgdila* which must be greater than *pgpact*.

DEFAULTS:

If keyword *PGPDMAX does not appear, *pgpdmax* has a very high value.

EXPLANATION:

The pseudo dilation model is implemented in CMG's geomechanics module to produce a pressure-porosity curve similar to that of the STARS *DILATION empirical model. In the pseudo dilation model, the change of Young's modulus is controlled by threshold pressures specified by *PGDILA, *PGPACT and *PGPDMAX as shown in the below figure.



In the above figure, the six paths can be grouped as follows:

1. Elastic Path 1, Unloading Path, Reloading Path, Elastic Path 2: The material behavior is elastic so the load and deformation is reversible. A single Young's modulus specified via keyword *ELASTMOD is used for all these paths except Elastic Path 2.
2. Dilation Path: Young's modulus will change to *young1* when the pressure is larger than *pgdila*. On this path the load and deformation cannot be reversed; if pressure decreases at some point, the material behavior starts down an Unloading Path instead.
3. Recompaction Path: If while on the unloading path the pressure continues decreasing until it is less than *pgpact*, the material behavior changes to a recompaction state with Young's modulus *young2*. On the Recompaction Path the load and deformation is not reversible. If the pressure increases then the material behavior will go along the Reloading Path instead.

Porosity Coupling

With *PGDILA there are two porosity coupling options available. In each case keywords *PGDILA, *PGPACT and *PGPDMAX control the Young's modulus for each path. See the EXPLANATION for keyword *GCOUPLING.

***GCOUPLING 0**

Use this coupling option with *PGDILA to obtain more accurate deformation and stress estimates while still getting the fluid-flow pressure-porosity behavior from the STARS *DILATION model. In this coupling, porosity is a function of pressure and temperature only.

***GCOUPLING 2**

This type of coupling is more compact than the previous one since the porosity is not only a function of pressure and temperature but also a function of mean total stress.

Example:

```
*GEOROCK 1     ** Rock type # 1
*ELASTMOD 5e4  ** Young's modulus for elastic path 1
*POISSRATIO 0.3  ** Poisson's ratio
*YLDSTRESS 1e10  ** Very large yield stress
** dilation keyword   threshold p   Young's mod
*PGDILA               1536.0        1e4
*PGPACT               500.0         2e4
```

Generalized Plasticity Model *GENPLAST, *ELASTMOD, *POISSRATIO, *PEAKCOHES, *RESDUCOHES, *EPSCUTOFF, *COHESEXPNT, *FRICHARDEN, *INITFANGLE, *PEAKFANGLE, *CVFANGLE, *COHESHARD, *FRANGSOFT

PURPOSE:

Use the Generalized Plasticty constitutive model which allows cohesion, friction angle and dilation angle to change with effective plastic strain.

FORMAT:

*GENPLAST	*option*
*ELASTMOD	*elastmod*
*POISSRATIO	*poissratio*
*PEAKCOHES	c_p
*RESDUCOHES	c_{res}
*EPSCUTOFF	ε_c
*COHESEXPNT	n
*FRICHARDEN	a
*INITFANGLE	φ_0
*PEAKFANGLE	φ_p
*CVFANGLE	φ_{cv}
*COHESHARD	
*FRANGSOFT	

DEFINITION:

*GENPLAST *option*

Specify that the Generalized Plasticity model along with the sub-option. Keywords *ELASTMOD and *POISSRATIO will be required.

option	Sub-option
1	Associated Drucker-Prager
2	Non-associated Drucker-Prager
3	Associated Mohr-Coulomb
4	Non-associated Mohr-Coulomb

*ELASTMOD *elastmod*

Young's modulus (kPa | psi | kPa).

*POISSRATIO *poissratio*

Poisson's ratio.

*PEAKCOHES c_p

Peak cohesion (kPa | psi | kPa).

*RESDUCOHES c_{res}

Residual value of cohesion (kPa | psi | kPa).

*EPSCUTOFF ε_c
Effective plastic strain cut-off for cohesive softening.

*COHESEXPNT n
Exponent number for cohesive softening.

*FRICHARDEN a
Coefficient for friction hardening.

*INITFANGLE φ_0
Initial friction angle (degrees).

*PEAKFANGLE φ_p
Maximum or peak friction angle (degrees).

*CVFANGLE φ_{cv}
Friction angle at constant volume (degrees).

*COHESHARD
Condition for hardening the cohesion.

*FRANGSOFT
Condition for softening the friction angle.

DEFAULT:

If *PEAKCOHES is absent then $c_p = 0$ is assumed.

If *RESDUCOHES is absent then $c_{res} = 0$ is assumed.

If *EPSCUTOFF is absent then $\varepsilon_c = 100$ is assumed.

If *COHESEXPNT is absent then $n = 0$ is assumed.

If *FRICHARDEN is absent then $a = 0$ is assumed.

If *INITFANGLE is absent then $\varphi_0 = 30$ is assumed.

If *PEAKFANGLE is absent then $\varphi_p = 30$ is assumed.

If *CVFANGLE is absent the $\varphi_{cv} = 30$ is assumed.

If *COHESHARD is absent, softening the cohesion is assumed.

If *FRANGSOFT is absent, hardening the friction angle is assumed.

CONDITION:

If *GENPLAST appears then *ELASTMOD and *POISSRATIO are required.

EXPLANATION:

Geo-materials normally exhibit pressure sensitivity and dilation behavior. These behavioral aspects can be described within the framework of non-associated isotropic hardening and softening and some convenient yield function which encapsulates all the basic ingredients of a granular material. The generalized plasticity model is based on either Mohr-Coulomb or Drucker-Prager yield surface. With this model, cohesion and friction angle are allowed changing with effective plastic strain according to the rules:

1. By default, cohesion decays exponentially with effective plastic strains until it reaches a residual value. In order to keep a constant cohesion, the values n of COHESEXPNT keyword should be zero. When the keyword *COHESHARD is used, the cohesion increases with the effective plastic strain until it reaches its maximum value via the keyword *PEAKCOHES. In practice, the value of peak cohesion should not be zero.
2. By default, friction angle is increasing with effective plastic strains until it reaches a peak value. However, when the keyword *FRANGSOFT is applied to a rock type, the friction angle starts decaying along with increasing the effective plastic strain from the peak angle to the value of φ_0.
3. Dilation angle derived by Row is a function of friction angle. Therefore, when a non-associated plastic flow rule is used, the dilation angle also varies along with the change of effective plastic strain.

The physics behind the trend of decreasing cohesion with increasing effective plastic strains is based on the fact that there is usually a gradual loss of cementation as the material structure starts microcracking due to shearing. On the other hand, with increasing effective plastic strain, more friction is mobilized thus causing hardening towards a peak angle or an ultimate friction angle value. Hence, the dilation angle increases along with the friction angle.

Softening and hardening cohesion may not be simultaneously combined in one material.

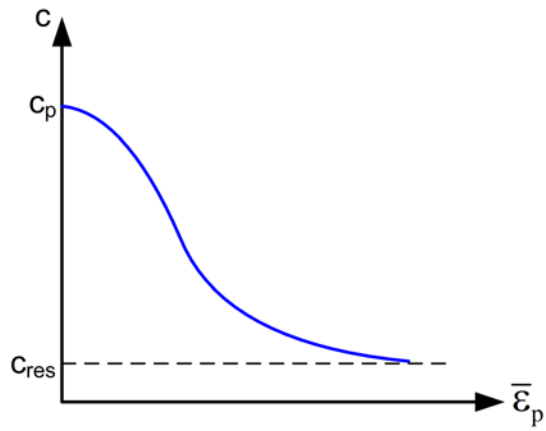
By default, cohesion, friction angle and dilation angle are expressed as:

$$c = \left(c_p - c_{res}\right)\exp\left[-\left(\frac{\bar{\varepsilon}_p}{\varepsilon_c}\right)^n\right] + c_{res}$$

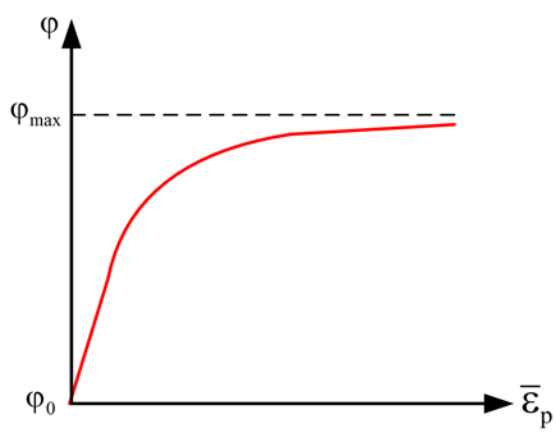
$$\varphi = \sin^{-1}\left(2\sin\left(\varphi_p - \varphi_0\right)\frac{\sqrt{a\bar{\varepsilon}_p}}{a + \bar{\varepsilon}_p}\right) + \varphi_0$$

$$\sin\psi = \frac{\sin\varphi - \sin\varphi_{cv}}{1 - \sin\varphi\sin\varphi_{cv}}$$

In this constitutive model, the function of friction angle φ must be monotonic. Therefore, the coefficient for friction hardening (*FRICHARDEN *a*) should exceed the largest value expected for the computed effective plastic strain $\bar{\varepsilon}_p$.



Nonlinear softening cohesion vs. effective plastic strain



Hardening friction angle φ versus effective plastic strain

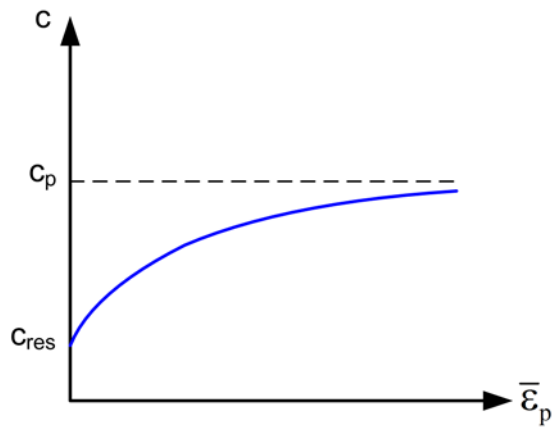
Example 1:

```
** Non-linear Cohesion Softening & Hardening Friction Angle
*GENPLAST 1  ** Generalized plasticity
             ** Drucker-Prager associated flow
*ELASTMOD 9.83e5     ** Young's modulus
*POISSRATIO 0.21     ** Poisson's ratio
*PEAKCOHES  500.38   ** Peak cohesion
*RESDUCOHES 0.0      ** Residual cohesion
*EPSCUTOFF  100.0    ** Effective plastic strain cut-off
*COHESEXPNT 0.0      ** Exponent of cohesion softening
*FRICHARDEN 0.0      ** Coefficient for friction hardening
*INITFANGLE 30       ** Initial friction angle
*PEAKFANGLE 30       ** Peak friction angle
*CVFANGLE   0.0      ** Friction angle at constant volume
```

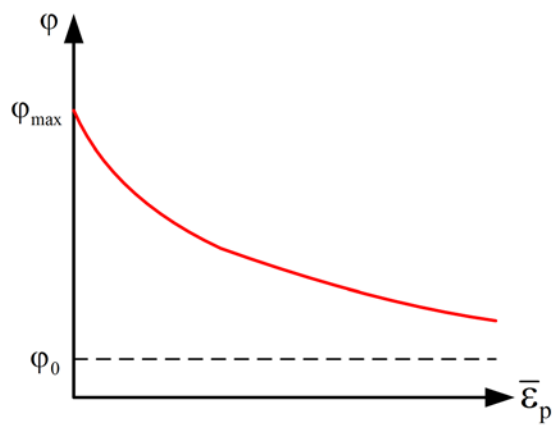
When keywords *COHESHARD and *FRANGSOFT appear, the cohesion and friction angle can be expressed as:

$$c = -\left(c_p - c_{res}\right)\exp\left[-\left(\frac{\bar{\varepsilon}_p}{\varepsilon_c}\right)^n\right] + c_p$$

$$\varphi = -\sin^{-1}\left(2\sin\left(\varphi_p - \varphi_0\right)\frac{\sqrt{a\bar{\varepsilon}_p}}{a + \bar{\varepsilon}_p}\right) + \varphi_p$$



Nonlinear hardening cohesion versus effective plastic strain



Softening friction angle φ versus effective plastic strain

Example 2:

```
** Non-linear Cohesion Softening & Hardening Friction Angle
*GENPLAST 1  ** Generalized plasticity
             ** Drucker-Prager associated flow
*ELASTMOD 1.00e5     ** Young's modulus (psi)
*POISSRATIO 0.30     ** Poisson's ratio
*PEAKCOHES  14.5     ** Peak cohesion (psi)
*RESDUCOHES 0.0      ** Residual cohesion (psi)
*EPSCUTOFF  1E-02    ** Effective plastic strain cut-off
*COHESEXPNT 0.2      ** Exponent of cohesion softening
*FRICHARDEN 5E-02    ** Coefficient for friction hardening
*INITFANGLE 20       ** Initial friction angle (deg)
*PEAKFANGLE 40       ** Peak friction angle (deg)
*CVFANGLE   20.      ** Friction angle at constant volume (deg)
*COHESHARD           ** Hardening cohesion
*FRANGSOFT           ** Softening friction angle
```

Reference:

Vermeer, P.A. and R. De Borst: "Non-Associated Plasticity for Soils, Concrete and Rock," Heron, Vol. 29(3), pp. 5-64, 1984.

Single Surface Failure Model *SSMODEL, *ELASTMOD, *POISSRATIO, *SGAMMA, *SBETA, *SM, *SN, *SNG, *SA1, *SETHA1, *SKAPA1, *SKAPA2, *STEN, *SPATM

PURPOSE:

The Single Surface Failure constitutive model captures the main behaviors of geomaterials under complex states of stress.

FORMAT:

*SSMODEL	*option*
*ELASTMOD	*elastmod*
*POISSRATIO	*poissratio*
*SGAMMA	γ
*SBETA	β
*SM	m
*SN	n
*SNG	g
*SA1	a_1
*SETHA1	η_1
*SKAPA1	κ_1
*SKAPA2	κ_2
*STEN	T
*SPATM	p_a

DEFINITION:

*SSMODEL *option*

Specify the Single Surface failure model and sub-option. Keywords *ELASTMOD and *POISSRATIO will be required.

option	Sub-option
1	Associated plastic flow
2	Non-associated plastic flow

*ELASTMOD *elastmod*

Young's modulus (kPa | psi | kPa).

*POISSRATIO *poissratio*

Poisson's ratio.

*SGAMMA γ

Material parameter related to the yield surface.

*SBETA β

Material parameter used to control the shape of failure surface on the octahedral plane.

*SM *m*

Power of stress ratio

*SN n

Parameter used to control the phase change when the volume changes from compaction to dilation or vanishes.

*SNG g

Power used to control the shape of failure surface in a meridian plane.

*SA1 a_1

Parameter used to compute a hardening value

*SETHA1 η_1

Exponential parameter for effective plastic strain.

*SKAPA1 κ_1

Material constant used to compute a hardening value for a non-associated flow

*SKAPA2 κ_2

Material constant used to compute a hardening value for a non-associated flow

*STEN T

Tensile strength of rock (kPa | psi | kPa).

*SPATM p_a

Atmospheric pressure (kPa | psi | kPa).

DEFAULT:

If *SGAMMA is absent then $\gamma = 1$ is assumed.

If *SBETA is absent then $\beta = 0$ is assumed.

If *SM is absent then $m = 0$ is assumed.

If *SN is absent then $n = 0$ is assumed.

If *SNG is absent then $g = 2$ is assumed.

If *SA1 is absent then $a_1 = 0$ is assumed.

If *SETHA1 is absent then $\eta_1 = 0$ is assumed.

If *SKAPA1 is absent then $\kappa_1 = 0$ is assumed.

If *SKAPA2 is absent then $\kappa_2 = 0$ is assumed.

If *STEN is absent then $T = 0$ is assumed.

If *SPATM is absent then p_a = 14.7 psi (101.3 kPa) is assumed.

CONDITION:

If *SSMODEL appears then *ELASTMOD and *POISSRATIO are required.

EXPLANATION:

The Single Surface Failure model discussed here is based on the work of Desai[1]. This model, also called the hierarchical model, captures the main features of geomechanical behavior: isotropic and dynamic hardening, softening, and associated and non-associated plasticity under complex states of stress. In comparison with multi-surface models, the Single Surface model has many advantages, including:

1. No numerical instabilities caused by operation at the non-smooth intersection of two surfaces in a multi-surface model.
2. The model allows different strengths under different stress paths.
3. The model allows change in the shape and size of failure envelope, tension and cohesion of a material.

The yield function F of the model[2] is:

$$F = \frac{J_2}{p_a^2} - \left[-\alpha \left(\frac{I_1 + T}{p_a} \right)^n + \gamma \left(\frac{I_1 + T}{p_a} \right)^g \right] (1 - \beta S_r)^m \qquad (1)$$

The potential function Q is:

$$Q = \frac{J_2}{p_a^2} - \left[-\alpha_Q \left(\frac{I_1 + T}{p_a} \right)^n + \gamma \left(\frac{I_1 + T}{p_a} \right)^g \right] (1 - \beta S_r)^m \qquad (2)$$

where:

Compressive stress is positive.

J_2: The second invariant of deviatoric stress tensor

I_1 : The first invariant of stress tensor

P_a: The atmospheric pressure

S_r : Stress ratio

$$S_r = \frac{\sqrt{27}}{2} \frac{J_3}{(J_2)^{1.5}} \qquad (3)$$

J_3: The third invariant of deviatoric stress tensor

α: The hardening function in F

$$\alpha = \frac{a_1}{\xi^{\eta_1}} \qquad (4)$$

The increments of ξ and ξ_w are computed by:

$$d\xi = \lambda \left(\frac{\partial Q}{\partial \sigma_{ij} \partial \sigma_{ij}} \right)^{1/2} \qquad (5)$$

$$d\xi_v = \lambda \left(\frac{\partial Q}{\partial \sigma_{ii}} \right) / \sqrt{3} \qquad (6)$$

α_Q: The hardening function in Q

$$\alpha_Q = \alpha + \kappa(\alpha_0 - \alpha)(1 - r_v) \qquad (7)$$

$$\kappa = \kappa_1 + \kappa_2 S_r \qquad (8)$$

$$r_v = \frac{\xi_v}{\xi} \qquad (9)$$

λ: A scalar also called plastic multiplier

If we let T = 0, m = 0, g = 2 and α = 0, Equation (1) becomes:

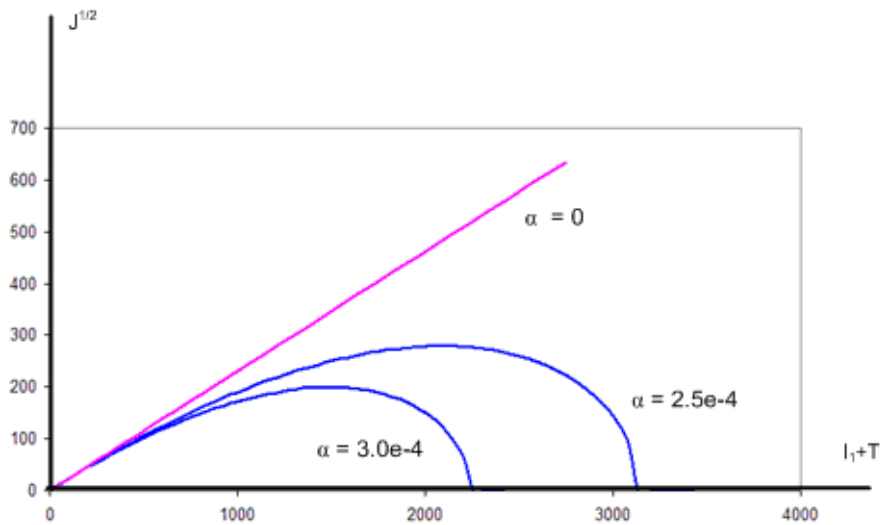
$$F = \frac{J_2}{p_a^2} - \gamma\left(\frac{I_1}{p_a}\right)^2 = 0 \qquad (10)$$

Equation (10) is an approximation to the Drucker-Prager constitutive model when:

$$\sqrt{\gamma} \approx \frac{2\sin\phi}{\sqrt{3}(3-\sin\phi)} \qquad (11)$$

Where ϕ is friction angle. For some problems, the value of γ should be slightly adjusted to obtain equivalent results.

The following figure is an example to illustrate how the parameter α changes the shape of failure envelope. Data used for plotting the curves are: g = 2, γ = 0.0533, n = 3, m = 0, T = 25.1 psi and p_a = 14.7 psi.



Other parameters such as g, n and β can also change the shape of failure envelope. More details are discussed in the references 1 and 2.

When α = 0, the failure envelope reaches the ultimate state.

Example 1:

This is an associated flow case in field units.

```
*SSMODEL 1          ** Associated flow
*ELASTMOD  1.5E4    ** Young's modulus (psi)
*POISSRATIO  0.3    ** Poisson's ratio
*SGAMMA  0.0533     ** Gamma
*SN    3.0          ** Power n
*SNG  2.0           ** Power g
*STEN  -25.1        ** Tensile strength (psi)
*SPATM  14.7        ** Ambient pressure (psi)
```

Example 2:

This is a non-associated flow case in field units.

```
*SSMODEL 2          ** Non-associated flow
*ELASTMOD  1.5E4    ** Young's modulus (psi)
*POISSRATIO  0.3    ** Poisson's ratio
*SGAMMA  0.0533     ** Gamma
*SN    3.0          ** Power n
*SNG  2.0           ** Power g
*SKAPA1  0.05       ** Hardening parameter 1
*SKAPA2  -0.01      ** Hardening parameter 2
*STEN   -25.1       ** Tensile strength (psi)
*SPATM  14.7        ** Ambient pressure (psi)
```

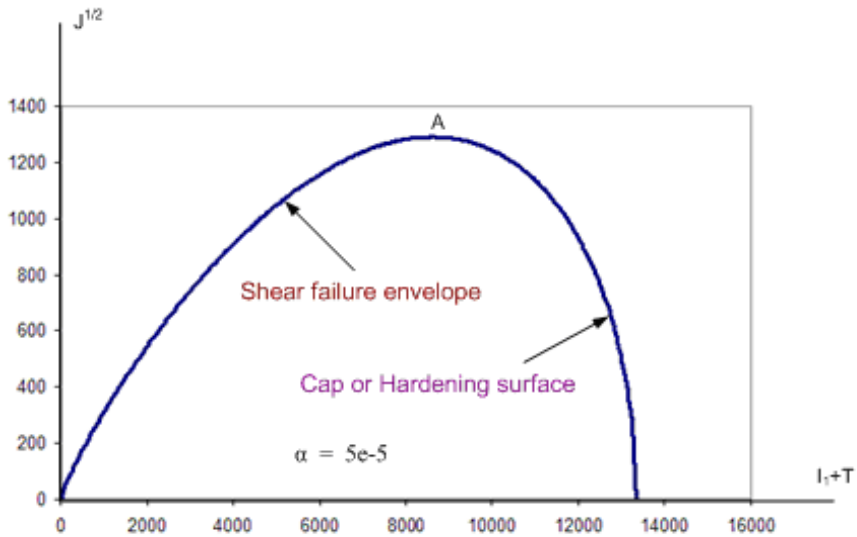
Example 3:

This is a hardening surface case in field units. This case is equivalent to a cap model because there is a dynamically hardening surface which encloses the shear failure envelope on the compressive stress axis.

```
*SSMODEL 1          ** Associated flow
*ELASTMOD  1.5E4    ** Young's modulus (psi)
*POISSRATIO  0.3    ** Poisson's ratio
*SGAMMA  0.35       ** Gamma
*SN    3.0          ** Power n
*SNG  1.7           ** Power g
*SA1   5E-5         ** Hardening parameter
*SETHA1  4E-1       ** Plastic strain parameter
*STEN   -25.1       ** Tensile strength (psi)
*SPATM  14.7        ** Ambient pressure (psi)
```

For the purpose of representing a cap model situation, the key data of the Single Surface model is the plastic strain parameter *SETHA1 which must be non-zero.

Using the above data with α = 5e-5 or η_1 = 0, the following single failure surface is plotted.



It should be noted that on the left of the maximum point A on the curve is the shear failure envelope and on the right of it is the cap or hardening surface.

When using the keyword *YLDSTATE to illustrate failure modes of a material on RESULTS®, there are 3 states for the single surface failure model:

1. If its value is 0, the material is in an elastic state lying inside the curve.
2. If its value is 1, the material is in a plastic state lying on the shear failure envelope.
3. If its value is 2, the material is in a plastic state lying on the cap surface.

References:

Desai, S.C.: **Mechanics of Materials and Interfaces**: *The Disturbed State Concept*, CRC Press, 2001.

Liu, X. et al.: "Numerical modelling of nonlinear response of soil. Part 1: Constitutive model", **Int. J, Solids and Structures**, Vol. 42. pp. 1849-1881, 2005.

Modified Cam Clay Model *MCCMODEL, *SHEARMOD, *POISSRATIO, *FRICANGLE, *MCSWINDX, *MCOMINDX, *MCRCSL, *MCPREHD, *MCETHA, *MCOCR

PURPOSE:

The Modified Cam Clay (MCC) constitutive model captures the behavior of soft geomaterials such as soil or sand.

FORMAT:

*MCCMODEL	*option*
*SHEARMOD	*shearmod*
*POISSRATIO	*poissratio*
*FRICANGLE	*angle*
*MCSWINDX	*swellindx*
*MCOMINDX	*comindx*
*MCRCSL	*ratio*
*MCPREHD	*hard*
*MCETHA	*etha*
*MCOCR	*ocr*

DEFINITION:

*MCCMODEL *option*
Specify the Modified Cam Clay failure model and sub-option.

option	Sub-option
1	Associated plastic flow
2	Non-associated plastic flow

*SHEARMOD *shearmod*
Shear modulus (kPa | psi | kPa).

*POISSRATIO *poissratio*
Poisson's ratio.

*FRICANGLE *angle*
Friction angle (degrees)

*MCSWINDX *swellindx*
Swelling index or gradient of swelling line

*MCOMINDX *comindx*
Compression index or gradient of virgin consolidation line (VCL)

*MCRCSL *ratio*
Ratio of critical state lines in compression and tension

*MCPREHD *hard*
Initial hardening or preconsolidated pressure (kPa | psi | kPa)

*MCETHA η
Factor for potential failure surface function – Nonassociated plastic flow

*MCOCR *ocr*

Over consolidated ratio

DEFAULT:

If *SHEARMOD is absent, shear modulus will be calculated at every time step.

If *FRICANGLE is absent then *angle* = 30 degrees is assumed.

If *MCRCSL is absent then *ratio* = 1 is assumed.

If *MCETHA is absent then η = 0 is assumed.

If *MCOCR is absent then *ocr* = 0 is assumed.

CONDITIONS:

If *MCCMODEL appears then * MCSWINDX and * MCOMINDX are required. Besides, other requirements are also needed for the model as follows:

If *SHEARMOD appears, the Poisson's ratio is computed at every time step.

If *POISSRATIO appears, the shear modulus is computed at every time step.

If *MCPREHD appears, the initial hardening is maximum of the value of *hard* and the initial mean effective stress σ'_m.

If *MCOCR appears, the value of *ocr* must be equal or greater than unity. The initial hardening is the maximum of (*hard* and *ocr**σ_0); σ_0 is computed in the next section.

If *MCOCR appears, the keyword *MCPREHD may be absent. In this case, the initial hardening is equal to *ocr**σ_0.

The value of swelling index (*swellindx*) is always less than the value of compression index (*comindx*).

If *MCETHA appears, the value of η must be equal or greater than zero.

EXPLANATION:

For soft geomaterials such as soil or sand, this modified cam clay model can be used for simulation. In this model, Young's modulus continues changing via the bulk modulus. Poisson's ratio can be changed or kept constant depending on the user's choice. If the Poisson's ratio is kept constant, shear modulus will change with time. In this case, the keyword *SHEARMOD must be absent in the data. If the shear modulus is kept constant, Poisson's ratio is allowed to change with time. In this case, the keyword *SHEARMOD must appear in the data and the keyword POISSRATIO can be absent. Even though the keyword POSSRATIO appears in the data, the keyword *SHEARMOD will dominate.

The failure surface function of this model has a form as:

$$F(I, J, \alpha, \sigma_0) = I^2 + \left(\frac{J}{N}\right)^2 - 3\sigma_0 I = 0$$

where:

I : First invariant of effective stress tensor

J : Square root of second invariant of effective stress tensor

$3\sigma_0$: Intersection between the failure surface and I axis.

N : $= \dfrac{M}{3\sqrt{3}}$

M : $= g(\text{ratio},\theta)\ Mc = \dfrac{2\text{ratio}}{1+\text{ratio}-(1-\text{ratio})\sin 3\theta}\ Mc$

θ : Lode's angle defined as: $-\dfrac{\pi}{6} \le \theta \le \dfrac{1}{3}\arcsin\left[\dfrac{3\sqrt{3}}{2}\dfrac{J_3}{J^{3/2}}\right] \le \dfrac{\pi}{6}$

J_3 : Third invariant of effective stress tensor = $\dfrac{1}{3}s_{ij}s_{jk}s_{ki}$

s_{ij} : Deviatoric stress = $\sigma'_{ij} - \dfrac{1}{3}I\delta_{ij}$

σ'_{ij} : Effective stress

δ_{ij} : Kronecker delta = 1 if $i = j$ and =0 if $i \neq j$

M_c : Slope of critical state line (CSL) in compression = $\dfrac{6\sin\varphi}{2-\sin\varphi}$

φ : Friction angle

The value of σ_0 which is defined above can also be called as hardening. At the initial state, it has a value as

If $I \neq 0$.

$$\sigma_0 = \max(\text{hard}, \frac{1}{3I}\left(I^2 + \left(\frac{J}{N}\right)^2\right)) \quad \text{if *MCOCR is absent}$$

$$\sigma_0 = \max(\text{hard}, \frac{\text{ocr}}{3I}\left(I^2 + \left(\frac{J}{N}\right)^2\right)) \quad \text{if *MCOCR appears}$$

If I = 0,

$\sigma_0 = \textit{hard}$

Mobilized value of σ_0 can be calculated as:

$$\sigma_0^{n+1} = \sigma_0^n \exp\left(C_1\left(\Delta\varepsilon_v - \frac{\Delta\sigma'_m}{K_n}\right)\right)$$

where:

$\Delta\sigma'_m$: Change in mean effective stress between time step n+1 and n

$\Delta\varepsilon_v$: Change in volumetric strain between time step n+1 and n

K_n : Bulk modulus at time step n = $\dfrac{(1+e_0)}{\text{swellindx}}\dfrac{I}{3}$

e_0 : Initial void ratio = $\dfrac{\phi_0}{1-\phi_0}$

ϕ_0 : Initial porosity

Shear modulus G_n at time n can be computed if Poisson's ratio ν_0 is given initially as follows:

$$G_n = \frac{K_n(3-6\nu_0)}{2(1+\nu_0)}$$

If shear modulus G_0 is given initially, Poisson's ratio at time n is computed by:

$$\nu_n = \frac{3K_n - 2G_0}{2G_0 + 6K_n}$$

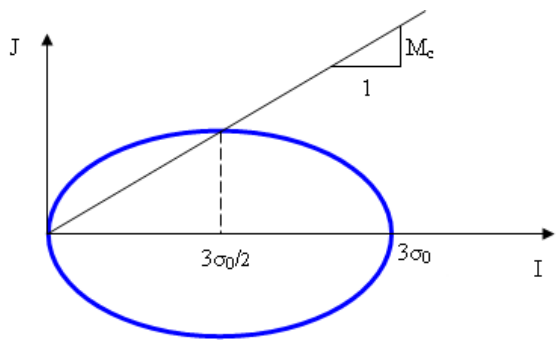
Young's modulus at time n is computed by:

$$E_n = 3K_n(1-\nu_0) \quad \text{or} \quad E_n = 3K_n(1-\nu_n)$$

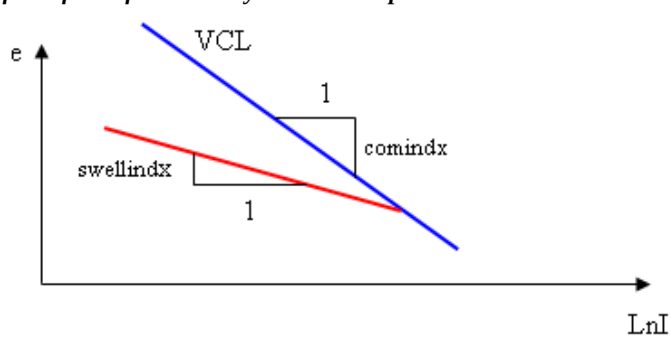
For non-associated plastic flow, the potential function has a form:

$$Q(I, J, \alpha, \sigma_0) = I^2 + \left(\frac{J}{N}\right)^2 - 3\eta\sigma_0 I = 0$$

Where η via the keyword MCETHA is defined above.



Yield surface of modified cam clay model in I-J plane



In the void ratio e and LnI plane

The input of material properties for a modified cam clay model is given as follows:

Example 1: No shear modulus is used

```
*MCCMODEL 1        ** Associated flow
*POISSRATIO  0.3   ** Poisson's ratio
*MCSWINDX  0.018   ** Swelling index
*MCOMINDX    0.13  ** Compression index
*FRICANGLE  26.0   ** Friction angle (degrees)
*MCRCSL   1.0      ** Ratio between tension and compression indices
*MCPREHD  20.0     ** Initial hardening (psi)
```

Example 2: With shear modulus and OCR are used. Non-associated plastic flow is applied

```
*MCCMODEL 2        ** Non-associated flow
*SHEARMOD  2.0E5   ** Shear modulus (psi)
*MCSWINDX  0.018   ** Swelling index
*MCOMINDX    0.13  ** Compression index
*FRICANGLE  26.0   ** Friction angle (degrees)
*MCRCSL   1.0      ** Ratio between tension and compression indices
*MCPREHD  20.0     ** Initial hardening (psi)
*MCOCR    3.0      ** Overconsolidated ratio
*MCETHA   0.0      ** Factor in non-associated flow potential
```

References:

Hashash, Y.M.A. and Whittle, A.J. "Integration of the Modified Cam-Clay Model in Non-Linear Finite Element Analysis", *Computers and Geotechnics*, Vol. 14, pp. 59-83, 1992.

Borja, R.I. and Tamagnini, C. "Cam-Caly Plasticity. Part III: extension of the Infinitesimal Model to Include Finite Strains", Comput. Methods. Appl. Mech. Eng., Vol. 155, pp. 73-95, 1998.

Jointed Rock *ROCKJOINT, *IKPJ, *IJPJ, *CKNJ, *CKSJ, *CKSPACING, *DILFACTOR, *CRAKSET, *CRAKTYPE

PURPOSE:

A mass of jointed rock is treated as anisotropy, as opposed to the isotropy of an intact rock.

FORMAT:

*ROCKJOINT *joint_type*
*IKPJ *angle1*
*IJPJ *angle2*
*JKPJ *angle3*
*CKNJ *stiffn*
*CKSJ *stiffs*
*CKSPACING *space*
*DILFACTOR *dilfac*
*CRAKSET *crack_num*
{ *JOINT *joint_type* }

ARRAY:

*CRAKTYPE

DEFINITION:

*ROCKJOINT *joint_type*
Indicate a new joint type, with positive integer identifier *joint_type*. Use *joint*_type after *CRAKSET sub-keyword *JOINT to assign a defined joint type to a crack set. These sub-keywords assign properties to this joint type:

*IKPJ, *IJPJ, *JKPJ, *CKNJ, *CKSJ, *CKSPACING, *DILFACTOR

For intact rock, specify only zero joint spacing via *CKSPACING.

*IKPJ *angle1*
Specify joint angle *angle1* (degrees) with respect to the X axis in the IK plane. This is also called dip angle. The allowed range of *angle1* is [0,180].

*IJPJ *angle2*
Specify joint angle *angle2* (degrees) with respect to the X axis in the IJ plane. This is also called strike angle. The allowed range of *angle2* is [0,180].

*JKPJ *angle3*
Specify joint angle *angle3* (degrees) with respect to Y axis in the JK plane. The allowed range of *angle3* is [0,180].

*CKNJ *stiffn*
Specify normal stiffness *stiffn* (kPa/m | psi/ft) of a joint. The value must be greater than zero.

*CKSJ *stiffs*

Specify shear stiffness *stiffs* (kPa/m | psi/ft) of a joint. The value must be greater than zero.

*CKSPACING *space*

Specify spacing *space* (m | ft) to indicate the distance between two joints in each jointed type. When *space*=0, the rock is assumed to be intact and no other *ROCKJOINT sub-keywords are required.

*DILFACTOR *dilfac*

Specify dilation factor *dilfac*, a real positive number between 0 and 1.

*CRAKSET *crack_num*

Specify a crack set with identifier *crack_num* by combining joint types defined bykeyword *ROCKJOINT.

{ *JOINT *joint_type* }

Table of joint types contained in crack set *crack_num*. Each *joint_type* must have been defined previously via keyword *ROCKJOINT.

*CRAKTYPE

Assign crack sets defined via *CRAKSET to grid blocks, using any of the array reading options.

DEFAULTS:

If keyword *DILFACTOR is absent from a *ROCKJOINT definition, that joint type has no dilation effect.

CONDITIONS:

If keyword *CRAKTYPE is used, a valid *crack_num* must be assigned to each grid block.

This option is available only with elastic materials.

All the crack set numbers assigned using *CRAKTYPE must have been defined via keywords *ROCKJOINT and *CRAKSET.

For 2-D plane strain, only one angle is needed. For 3-D cases, only two angles are needed. Three angles should not be used simultaneously.

EXPLANATION:

Rock can be characterized as either intact rock or jointed rock mass. A rock mass is called intact when there is no crack, so it can be considered as a continuum and hence isotropic. In contrast, a jointed rock mass (discontinuum rock) is a compound material including intact rocks and jointed rocks containing numerous joints or small fractures in different directions (Fig. 1). Because of this complex geometry, the treatment becomes more difficult when separately computing matrix (intact rock) and fracture (joint) in a rock; therefore, a continuum approach is used. In this method, an equivalent constitutive model can be constructed to include all joint characteristics, thereby mimicking solutions obtained by a discrete element method. The main advantage of the continuum method is its simplicity, and it can be applied to any complicated rock mass. When a dilation factor (*DILFACTOR) is applied, the compliance tensor of equivalent constitutive model will be affected.

This model is available only with elastic materials; elasto-plastic materials will be studied in the future.

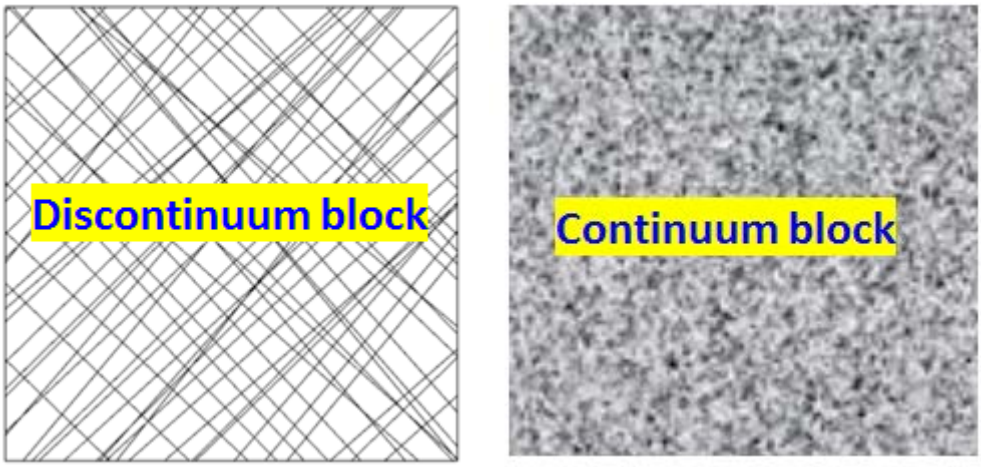


Fig. 1: Discontinuum block and continuum block

Example 1:

In this 2-D plane strain example, there are two rock joints and their properties are specified via keywords *ROCKJOINT. Since the plane strain is in the IK plane, only one angle in the IK plane is given (*IKPJ). The crack set (*CRAKSET) is defined with two rock joints before it is assigned grid blocks via keyword *CRAKTYPE.

Note that all grid blocks in Geomechanics must have a crack set number. The crack set may contain rock joints, or it may contain only intact rock (joint spacing is zero).

```
*ROCKJOINT 1
    *IKPJ  135.0            ** (degrees)
    *CKNJ  13.9E6           ** ( kPa/m)
    *CKSJ  6.95E6           ** ( kPa/m)
    *CKSPACING  0.1         ** ( m)

*ROCKJOINT 2
    *IKPJ  45               **  (degrees)
    *CKNJ  13.9E6           ** ( kPa/m)
    *CKSJ  6.95E6           ** ( kPa/m)
    *CKSPACING  0.1         ** (m)

*CRAKSET 1
    JOINT 1
    JOINT 2

*CRAKTYPE  *IJK             ** Crack set #
    1:20     1:1     1:10           1
```

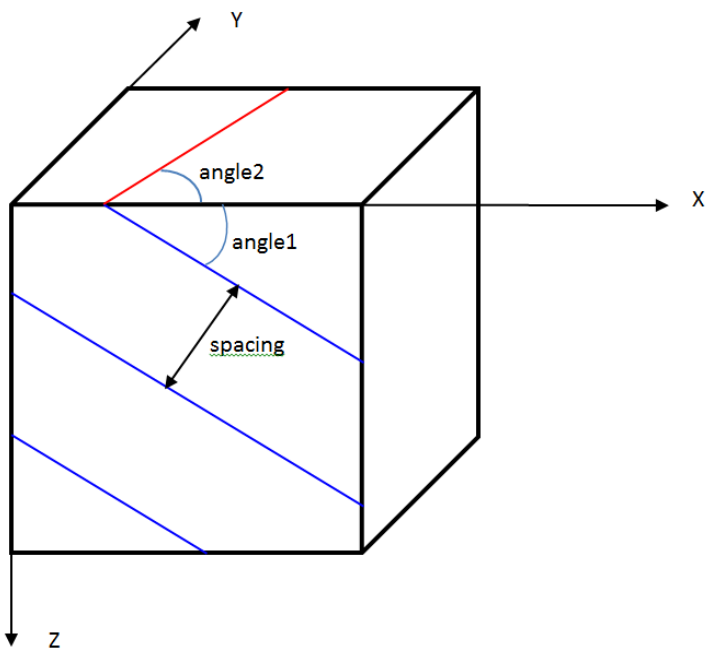
Example #2: This 3D case

```
*ROCKJOINT 1
  *IKPJ  135.0      ** (degrees) in IK plane
  *IJPJ     90.0    ** (degrees) in IJ plane
  *CKNJ  13.9E6     ** ( kPa/m)
  *CKSJ  6.95E6     ** ( kPa/m)
  *CKSPACING  0.0   ** ( m)   Treat as intact rock since spacing is zero

*ROCKJOINT 2
  *IKPJ  45 .0      ** (degrees) in IK plane
  *IJPJ   90.0      ** (degrees) in IJ plane
  *CKNJ  13.9E6     ** ( kPa/m)
  *CKSJ  6.95E6     ** ( kPa/m)
  *CKSPACING  0.1   ** (m)
  *DILFACTOR  0.2   ** Dilation factor

*CRAKSET 1
  JOINT 1
*CRAKSET 2
  JOINT 2

*CRAKTYPE  *IJK    ** Crack set #
  1:20   1:20   1:10  1  ** Whole Geomechanics domain is intact rock
  6:15   6:15   3:8   2  ** Part of Geomechanics domain is jointed rock mass
```



References:

Pande, G.N., Beer, G. and Williams, J.R., Numerical Methods in Rock Mechanics, John Wiley & Sons Ltd., Chichester, 1990.

Huang, T.H., Chang, C.S. and Yang, Z.Y., "Elastic Moduli for Fractured Rock Mass", Rock Mech. Rock Eng., Vol. 28 (3), pp. 135-144, 1995.

Thermal Expansion Coefficient *THEXPCOEF

PURPOSE:

Specify the thermal expansion coefficient for a rock type due to the thermal effect.

FORMAT:

*THEXPCOEF *thexpcoef*

DEFINITION:

thexpcoef
Linear thermal expansion coefficient (1/C | 1/F | 1/C).

CONDITION:

The keyword *THEXPCOEF and its value must be entered when the thermal effect is taken into account in the geomechanical model. When the keyword is absent, changes in thermal do not affect to the rock at all.

EXPLANATION:

When the temperature changes, the deformation as well as stress of rock also changes. Such changes strongly depend on the thermal expansion coefficient of a rock type. When the keyword *THEXPCOEF is absent in the geomechanical section, the thermal expansion coefficient is zero by default; therefore, the source term due to the change in temperature is zero in the equilibrium equation. That means the temperature does not play a part in calculation of stresses via the rock deformation. The value of thermal expansion coefficient can be measured or taken from a material handbook.

Example:

```
*GEOROCK 1           ** rock type # 1
*ELASTMOD  1.0e6   *POISSRATIO .3   *YLDSTRESS 1000000
*THEXPCOEF   1.0e-6
```

In this example, the value of linear thermal expansion coefficient is 10^{-6} (unit used) for rock type # 1.

Non-Uniform Formation Properties *YOUNGMAP, *POISSONMAP, *COHESIONMAP, *FRICANGMAP, *THEXPMAP

PURPOSE:

Specify properties for each grid cell separately, when the degree of non-uniformity would cause the number of rock types to be inconvenient.

ARRAY:

*YOUNGMAP
*POISSONMAP
*COHESIONMAP
*FRICANGMAP
*THEXPMAP

DEFINITION:

Each of these grid arrays specifies the same quantity as the corresponding per-rock-type keyword shown in this table.

Grid Array	Rock Type Keyword
*YOUNGMAP	*ELASTMOD *elastmod*
*POISSONMAP	*POISSRATIO *poissratio*
*COHESIONMAP	*COHESION *cohesion*
*FRICANGMAP	*FRICANGLE *fricangle*
*THEXPMAP	*THEXPCOEF *thexpcoef*

For the syntax of grid-array data, see section **Input of Grid Property Arrays** in the "Keyword Data Entry System" chapter. The only sub-keywords available are *CON, *ALL, *IVAR, *JVAR, *KVAR and *IJK. The grid indices correspond to the grid used for geomechanical calculations. See section **Host Grid versus Independent Grid** at the top of this chapter.

DEFAULTS:

Each cell that is not assigned the property via the grid-array keyword uses the property value assigned via the rock-type keyword or its default. Note that *ELASTMOD and *POISSRATIO are mandatory and so have no default.

CONDITION:

If a property is assigned to a cell via the grid-array keyword as well as the rock-type keyword, the last assignment (in top-down sequence) is the value that will be used. The usual sequence is to place the grid-array keyword below the rock-type keyword so that the property value of specified cells may be overwritten.

EXPLANATION:

Use these grid-array keywords to assign these properties when they vary over the entire grid, but only one or a few types of constitutive models are required.

These grid-array keywords may be used together with multiple rock types (keywords *GEOROCK and *GEOTYPE). Use rock types to assign all other non-uniform properties, including the type of constitutive model.

Examples:

The following examples assume a 5x5x2 grid geomechanics grid.

Assign the same value of Young's modulus to all blocks.

```
*YOUNGMAP *CON 1.2E6  ** Same value for all blocks
```

Assign a different value of Young's modulus to each block.

```
*YOUNGMAP *ALL  ** Different value for each block
1.0E6 1.2E6 1.3E6 1.4E6 1.5E6 1.6E6 1.5E6 1.4E6 1.3E6 1.2E6
1.1E6 1.2E6 1.3E6 1.4E6 1.5E6 1.6E6 1.5E6 1.4E6 1.3E6 1.1E6
1.0E6 1.2E6 1.3E6 1.4E6 1.5E6 1.6E6 1.5E6 1.4E6 1.3E6 1.2E6
1.1E6 1.2E6 1.3E6 1.4E6 1.5E6 1.6E6 1.5E6 1.4E6 1.3E6 1.1E6
1.0E6 1.2E6 1.3E6 1.4E6 1.5E6 1.6E6 1.5E6 1.4E6 1.3E6 1.2E6
```

Assign Young's modulus to irregular regions and Poisson's ratio to K layers.

```
*YOUNGMAP *IJK
  1:5 1:5 1  1.3E6  ** Layer #1
   1  1:3 1  1.4E6  ** Modify part of layer #1
  1:5 1:5 2  1.2E6  ** Layer #2
*POISSONMAP *KVAR  0.2  0.3
```

Use rock types to assign properties on a per-layer basis, then override in a disturbed zone. Note that Young's modulus for block (3,3,1) is first assigned 1E6 via *YOUNGMAP but is later overwritten with 1e5 via *GEOTYPE and rock type #3.

```
*GEOROCK 1     ** Lower layer
  *COHESION    100
  *THEXPCOEF   1E-5
*GEOROCK 2     ** Upper layer
  *COHESION    1E6
  *THEXPCOEF   1E-6
*GEOROCK 3     ** Disturbed zone
  *ELASTMOD    1.E5
  *POISSRATIO  0.35
  *COHESION    200
  *THEXPCOEF   1E-6
  *FRICANGLE   20
  *DILANGLE    0.0
  *DRUCKER
*YOUNGMAP   *KVAR  1E6  2E6
*POISSONMAP *KVAR  0.2  0.3
*GEOTYPE *IJK
   1:5 1:5  1   1  ** Lower layer
   1:5 1:5  2   2  ** Upper layer
    3   3  1:2  3  ** Disturbed layer
```

Matrix Permeability Multiplier *GPERMLC, *GPERMES, *GPERMTS, *GPERMVL, *GPERMNEX_MT, *GPERMNTX_MT, *GPERMNEY_MT, *GPERMNTY_MT, *GPERMNEZ_MT, *GPERMNTZ_MT, *USTRESCUR, *GULOGINT

PURPOSE:

Specify matrix-block permeability multiplier depending upon geomechanical response.

FORMAT:

*GPERMLC C_{n1}
or

*GPERMES
{ $\Delta\sigma'_E$ k_x/k_x^0 k_y/k_y^0 k_z/k_z^0 } or { σ'_E k_x/k_x^0 k_y/k_y^0 k_z/k_z^0 }
or

*GPERMTS
{ $\Delta\sigma_T$ k_x/k_x^0 k_y/k_y^0 k_z/k_z^0 } or { σ_T k_x/k_x^0 k_y/k_y^0 k_z/k_z^0 }
or

*GPERMVL
{ ε_v k_x/k_x^0 k_y/k_y^0 k_z/k_z^0 }
or

*GPERMNEX_MT
{ $\Delta\sigma'_{EX}$ k_x/k_x^0 } or { σ'_{EX} k_x/k_x^0 }

*GPERMNEY_MT
{ $\Delta\sigma'_{EY}$ k_y/k_y^0 } or { σ'_{EY} k_y/k_y^0 }

*GPERMNEZ_MT
{ $\Delta\sigma'_{EZ}$ k_z/k_z^0 } or { σ'_{EZ} k_z/k_z^0 }
or

*GPERMNTX_MT
{ $\Delta\sigma_{TX}$ k_x/k_x^0 } or { σ_{TX} k_x/k_x^0 }

*GPERMNTY_MT
{ $\Delta\sigma_{TY}$ k_y/k_y^0 } or { σ_{TY} k_y/k_y^0 }

*GPERMNTZ_MT
{ $\Delta\sigma_{TZ}$ k_z/k_z^0 } or { σ_{TZ} k_z/k_z^0 }

*USTRESCUR
*GULOGINT

DEFINITIONS:

*GPERMLC C_{n1}
Use the empirical model of Touhidi-Baghini to specify a matrix permeability multiplier, where C_{n1} (dimensionless) is an experimental parameter.

Multiplier $k/k^0 = \exp[\ C_{n1} \cdot \varepsilon_v\ ]$ is applied to a block's permeability in all three directions, where ε_v is the volumetric strain.

*GPERMES

Use a table to specify matrix permeability multipliers in all directions as a function of mean **effective** stress (kPa | psi). The stress is either $\Delta\sigma'_E$ (difference from initial value), or σ'_E (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMTS

Use a table to specify matrix permeability multipliers in all directions as a function of mean **total** stress (kPa | psi). The stress is either $\Delta\sigma_T$ (difference from initial value), or σ_T (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMVL

Use a table to specify matrix permeability multiplier as a function of volumetric strain ε_v (dimensionless). The strain entries must be monotonically increasing.

*GPERMNEX_MT

Use a table to specify matrix I-direction permeability multiplier as a function of normal **effective** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma'_{EX}$ (difference from initial value), or σ'_{EX} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNEY_MT

Use a table to specify matrix J-direction permeability multiplier as a function of normal **effective** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma'_{EY}$ (difference from initial value), or σ'_{EY} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNEZ_MT

Use a table to specify matrix K-direction permeability multiplier as a function of normal **effective** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma'_{EZ}$ (difference from initial value), or σ'_{EZ} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNTX_MT

Use a table to specify matrix I-direction permeability multiplier as a function of normal **total** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma_{TX}$ (difference from initial value), or σ_{TX} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNTY_MT

Use a table to specify matrix J-direction permeability multiplier as a function of normal **total** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma_{TY}$ (difference from initial value), or σ_{TY} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNTZ_MT

Use a table to specify matrix K-direction permeability multiplier as a function of normal **total** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma_{TZ}$ (difference from initial value), or σ_{TZ} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

k_x/k_x^0 k_y/k_y^0 k_z/k_z^0

Permeability multipliers in the X, Y and Z (I, J and K) directions, respectively (dimensionless). Multipliers can be different in the three grid directions.

*USTRESCUR

If this keyword is present for a rock type, the stress value (first column) in the above tables is "current value". If this keyword is absent for a rock type, the stress value in those tables is "difference from initial value". A single instance of this keyword per rock type, applies to both matrix and fracture multipliers (*GPERMNEX, etc., and GPERMNTX, etc.).

*GULOGINT

If this keyword is present for a rock type, use logarithmic interpolation for table look-up of a permeability-ratio column. If this keyword is absent for a rock type, use linear interpolation. A single instance of this keyword per rock type applies to both matrix and fracture multipliers (*GPERMNEX, etc., and GPERMNTX, etc.).

Let x be the independent variable (ε_v, $\Delta\sigma_E$ or $\Delta\sigma_T$) and let y be the dependent variable (k_x/k_x^0, k_y/k_y^0 or k_z/k_z^0); let subscripts 1 and 2 denote two adjacent table rows and let * denote values between the table rows. The logarithmic interpolation algorithm is

$$[\log(y^*)-\log(y_1)]/[\log(y_2)-\log(y_1)] = [x^*-x_1]/[x_2-x_1]$$

whereas the linear interpolation algorithm is

$$[y^*-y_1]/[y_2-y_1] = [x^*-x_1]/[x_2-x_1]$$

DEFAULTS:

For each separate rock type, if none of these matrix permeability multiplier options is specified then blocks will experience no matrix permeability change due to geomechanical effects.

For each separate rock type and grid direction, if that direction's table keyword is absent then there is no matrix permeability multiplier applied to the matrix permeability in that direction.

For each separate rock type, if keyword *USTRESCUR is absent then the first column in the above tables is interpreted as the difference in stress between the current value and the initial value.

For each separate rock type, if keyword *GULOGINT is absent then table look-up uses linear interpolation.

CONDITIONS:

These keywords are assigned to the current rock type specified via *GEOROCK, or to rock type #1 if *GEOROCK is absent.

These keywords are effective only if keyword *NOCOUPERM is absent.

For each separate rock type, at most one of the following options may be used:

1. *GPERMLC
2. *GPERMES
3. *GPERMTS
4. *GPERMVL
5. *GPERMNEX_MT, *GPERMNEY_MT, *GPERMNEZ_MT
6. *GPERMNTX_MT, *GPERMNTY_MT, *GPERMNTZ_MT

For each separate rock type, matrix and fracture permeability multiplier tables must not use different stress-type options. For example, if matrix uses *GPERMNEX_MT (effective stress), then fracture may not use *GPERMNTX (total stress).

EXPLANATION:

In order to include geomechanical responses in reservoir fluid flow, the permeability multiplier of matrix blocks can depend on geomechanical information such as volumetric strain, mean stress or normal stress, with stress being effective or total. Two different approaches are available: empirical formula (*GPERMLC) and table look-up options.

Table Look-up

This approach is more flexible than the empirical formula since the permeability multipliers may differ in the three directions. The first column of the table can be either volumetric strain, mean effective stress or mean total stress. The values in the first column must be monotonic, either increasing or decreasing. The columns of multipliers (k_x/k_x^0), (k_y/k_y^0) and (k_z/k_z^0) can be linear or nonlinear. To ensure that permeabilities entered as data are used as initial values, the table must contain a row with $k_x/k_x^0 = k_y/k_y^0 = k_z/k_z^0 = 1$ at ε_v, $\Delta\sigma_E$ or $\Delta\sigma_T = 0$ (*USTRESCUR absent).

The table look-up options are divided into two approaches.

1. *GPERMES, *GPERMTS, *GPERMVL: Multipliers in all three directions depend upon a stress value which is the mean of the normal stresses.
2. *GPERMNEX_MT, etc.: In each direction separately, a multiplier depends upon the normal stress in that direction.

In addition, the stress value can be either effective stress (e.g., *GPERMES) or total stress (e.g., *GPERMTS).

Finally, the interpretation of the stress column (current value or difference from initial value) is determined by keyword *USTRESCUR.

Example:

```
*GEOROCK 1
   *GPERMLC  100 ** Empirical formula is used
*GEOROCK 2
   *GULOGINT     ** Logarithmic interpolation
   *GPERMES      ** Mean effective stress change
       **  mean eff
       ** stress dif   (kx/kx0) (ky/ky0) (kz/kz0)
            -600.        1.5      1.5      2.0
            -500.        1.4      1.4      1.8
            -200.        1.1      1.1      1.2
               0.        1.0      1.0      1.0  ** Initial
             300.        0.8      0.8      0.7
             500.        0.75     0.75     0.6
             800.        0.7      0.7      0.5
*GEOROCK 3
   *GPERMVL      ** Volumetric strain
      ** Vol strain     (kx/kx0) (ky/ky0) (kz/kz0)
          -0.005         1.5      1.5      1.6
          -0.001         1.4      1.4      1.5
          -0.0005        1.3      1.3      1.4
          -0.0001        1.2      1.2      1.3
          -0.00005       1.1      1.1      1.2
          -0.00001       1.05     1.05     1.02
           0.0           1.0      1.0      1.0  ** Initial
           0.0001        0.8      0.8      0.7
           0.0005        0.7      0.7      0.6
*GEOROCK 4
   *USTRESCUR    ** Using current stress
   *GPERMNEY_MT  ** Normal effective stress in J direction
       **Stress (ky/ky0)
       -100.0    10E4
          0.0    10E3
         50.0    10E2
        100.0    10E1
        200.0    1.0
        300.0    0.8
        400.0    0.6
        500.0    0.5
        600.0    0.4
   *GPERMNEZ_MT  ** Normal effective stress in Z direction
       **Stress (kz/kz0)
       -100.0    10E4
          0.0    10E3
         50.0    10E2
        100.0    10E1
        200.0    1.0
        300.0    0.9
        400.0    0.8
        500.0    0.6
        600.0    0.5
```

Note that positive stress means compression and negative volumetric strain means expansion.

Empirical Formula

Based on the work of Touhidi-Baghini (Li, P. and Chalaturnyk, R.J., "*Permeability Variations Associated with Shearing and Isotropic Unloading during the SAGD Process*",

CIPC Paper 2004-240), the following multiplier is applied to a block's permeability in all three directions

$$k/k^0 = \exp[\, C_{n1} \cdot \varepsilon_v \,]$$

where

C_{n1}	experimental parameter for the block's rock type, and
ε_v	volumetric strain of the block

Fracture Permeability Multiplier *GPERMNEX, *GPERMNTX, *GPERMNEY, *GPERMNTY, *GPERMNEZ, *GPERMNTZ

PURPOSE:

Specify fracture-block permeability multiplier depending upon geomechanical response.

FORMAT:

*GPERMNEX
{ $\Delta\sigma'_{EX}$ k_x/k_x^0 } or { σ'_{EX} k_x/k_x^0 }

*GPERMNEY
{ $\Delta\sigma'_{EY}$ k_y/k_y^0 } or { σ'_{EY} k_y/k_y^0 }

*GPERMNEZ
{ $\Delta\sigma'_{EZ}$ k_z/k_z^0 } or { σ'_{EZ} k_z/k_z^0 }

or

*GPERMNTX
{ $\Delta\sigma_{TX}$ k_x/k_x^0 } or { σ_{TX} k_x/k_x^0 }

*GPERMNTY
{ $\Delta\sigma_{TY}$ k_y/k_y^0 } or { σ_{TY} k_y/k_y^0 }

*GPERMNTZ
{ $\Delta\sigma_{TZ}$ k_z/k_z^0 } or { σ_{TZ} k_z/k_z^0 }

DEFINITIONS:

*GPERMNEX

Use a table to specify fracture I-direction permeability multiplier as a function of normal **effective** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma'_{EX}$ (difference from initial value), or σ'_{EX} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNEY

Use a table to specify fracture J-direction permeability multiplier as a function of normal **effective** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma'_{EY}$ (difference from initial value), or σ'_{EY} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNEZ

Use a table to specify fracture K-direction permeability multiplier as a function of normal **effective** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma'_{EZ}$ (difference from initial value), or σ'_{EZ} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNTX

Use a table to specify fracture I-direction permeability multiplier as a function of normal **total** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma_{TX}$ (difference from initial value), or σ_{TX} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNTY

Use a table to specify fracture J-direction permeability multiplier as a function of normal **total** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma_{TY}$ (difference from initial value), or σ_{TY} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

*GPERMNTZ

Use a table to specify fracture K-direction permeability multiplier as a function of normal **total** stress (kPa | psi) in that direction. The stress is either $\Delta\sigma_{TZ}$ (difference from initial value), or σ_{TZ} (current value) when *USTRESCUR is present. Stress entries must be monotonic.

k_x/k_x^0 k_y/k_y^0 k_z/k_z^0

Permeability multipliers in the I, J and K directions, respectively (dimensionless). Multipliers can be different in the three grid directions.

*USTRESCUR

This keyword is described in the manual page for "Matrix Permeability Multiplier". A single instance of this keyword applies to both matrix and fracture multipliers.

DEFAULTS:

For each separate rock type and grid direction, if that direction's table keyword is absent then there is no fracture permeability multiplier applied to the fracture permeability in that direction.

CONDITIONS:

These keywords are assigned to the current rock type via *GEOTYPE *FRACTURE.

These keywords are used only by fracture reservoir blocks created by grid options *DUALPOR and *DUALPERM.

These keywords are effective only if keyword *****NOCOUPERM is absent.

For each separate rock type, at most one of the following options may be used:

1. *GPERMNEX, *GPERMNEY, *GPERMNEZ
2. *GPERMNTX, *GPERMNTY, *GPERMNTZ

For each separate rock type, none of the keywords above may be used together with keyword *GPERMBB (Barton-Bandis model).

These fracture-block tables use settings controlled by keywords *USTRESCUR and *GULOGINT, which are described in the manual page for "Matrix Permeability Multiplier".

For each separate rock type, matrix and fracture permeability multiplier tables must not use different stress-type options. For example, if matrix uses *GPERMNEX_MT (effective stress), then fracture may not use *GPERMNTX (total stress).

EXPLANATION:

Fracture permeability can be affected also by a change in geomechanics. The permeability multiplier for fracture may be a function of normal effective stress or normal total stress. Each table contains the stress in one direction and the fracture permeability multiplier in that

direction. The first column of each table must be monotonic, either increasing or decreasing. The stress values in the table should cover the range expected during the simulation. Note that positive stress means compression.

Example:

In this example there is no effect on the fracture permeability in the J direction since the keyword *GPERMNEY is missing. Only rock type #2 (fracture) contains tables *GPERMNEX and *GPERMNEZ.

```
*GEOROCK 1                  ** For matrix
   *ELASTMOD   3.95E5       ** Young's modulus (psi)
   *POISSRATIO 0.3          ** Poisson's ratio
   *COHESION   1E5          ** Cohesion (psi)

*GEOROCK 2                  ** For fracture
   *ELASTMOD   3.95E5       ** Young's modulus (psi)
   *POISSRATIO 0.3          ** Poisson's ratio
   *COHESION   1E5          ** Cohesion (psi)

   *USTRESCUR               ** Use normal stress value
   *GPERMNEX         ** Normal effective stress, I direction
   ** stress   (kx/kx0)
    -100.        10E4
       0.        10E3
      50.        10E2
     100.        10E1
     200.        1.0
     300.        0.8
     400.        0.6
     500.        0.5
     600.        0.4
   *GPERMNEZ         ** Normal effective stress, K direction
   ** stress    (kz/kz0)
    -100.        10E4
       0.        10E3
      50.        10E2
     100.        10E1
     200.        1.0
     300.        0.9
     400.        0.8
     500.        0.6
     600.        0.5

*GEOTYPE *MATRIX   *IJK I1:I2 J1:J2 K1:K2  1
*GEOTYPE *FRACTURE *IJK I1:I2 J1:J2 K1:K2  2
```

Barton-Bandis Fracture Permeability *GPERMBB, *UNFIXFDIR, *GFRACBLK

PURPOSE:

Use Barton-Bandis fracture permeability model.

FORMAT:

*GPERMBB e_0 *kni frs khf kccf krcf*
*UNFIXFDIR
*GFRACBLK *IJK
{ *i1*:*i2* *j1*:*j2* *k1*:*k2* }

DEFINITIONS:

*GPERMBB
Enable the Barton-Bandis fracture permeability model, for the current rock type specified by *GEOROCK or its default.

e_0
Initial fracture aperture (m | ft).

kni
Initial normal fracture stiffness (kPa/m | psi/ft).

frs
Fracture opening stress (kPa | psi).

khf
Hydraulic fracture permeability (md).

kccf
Fracture closure permeability (md).

krcf
Residual value of fracture closure permeability (md).

*UNFIXFDIR
Allow the fracture direction to vary with time, for the current rock type specified by *GEOROCK or its default.

*GFRACBLK *IJK
For the specified blocks, start fracture permeability on the Barton-Bandis model Path FG (see Figure below) instead of Path AB. This allows fracture permeability to use the "open fracture" model from the beginning without passing through the fracture opening path.

i1:*i2* *j1*:*j2* *k1*:*k2*
Range of grid indices in the I, J and K directions. Each index ranges from 1 to the maximum number of blocks in that direction as specified by keyword *GRID. Each lower index must not exceed the corresponding upper index. Enter a single index as a range, for example, 5:5.

DEFAULTS:

This option is disabled for each rock type for which keyword *GPERMBB is not defined.

If *UNFIXFDIR is absent, the fracture direction is fixed when the fracture opening occurs.

For each block not specified by *GFRACBLK but using the *GPERMBB option, fracture permeability starts on Path AB in below Figure.

CONDITIONS:

After *GPERMBB all six values must be entered and each value must be positive.

*GPERMBB may be used only with natural fracture grid options *DUALPERM and *DUALPOR.

*GPERMBB model parameters can be modified in recurrent data.

Keyword *GPERMBB is effective only if keyword *NOCOUPERM is absent.

EXPLANATION:

A natural-fracture grid option for fluid flow consists of the usual grid system for the porous rock matrix couple together with a second grid system consisting of fracture blocks that coexist with the matrix blocks on a one-to-one basis. Geomechanics calculations are coupled only to the matrix blocks. However, fracture opening and closing can depend upon stresses in the matrix blocks. Keyword *GPERMBB allows for calculation of the fracture block permeability from normal fracture effective stress via the Barton-Bandis model.

Fracture closure permeability is computed by:

$$k_f = kccf\left(\frac{e}{e_0}\right)^4 \geq krcf \qquad (1)$$

where:

k_f: Fracture closure permeability (m | md)

$$e = e_0 - V_j \qquad (2)$$

$$V_j = \frac{\sigma'_n}{kni + \sigma'_n / V_m} \qquad (3)$$ Joint closure under a normal fracture effective stress σ'_n

$$V_m = e_0\left[1 - \left(\frac{krcf}{kccf}\right)^{1/4}\right] \qquad (4)$$ Maximum fracture closure $(m|ft)$

Fracture permeability depends on the value and history of normal fracture effective stress σ'_n as illustrated in below Figure. It should be noted that the normal fracture effective stress σ'_n is equivalent to the minimum principle effective stress.

Path AB: Initially σ'_n is greater than opening fracture stress *frs*. On this path initial fracture permeability is very small and behavior is reversible.

Path BC: When σ'_n becomes less than *frs*, the fracture opens suddenly and fracture permeability jumps from the initial value to hydraulic fracture permeability *khf*.

Path DCE: As long as σ'_n is less than zero, fracture permeability remains at *khf*.

Paths EF and FG: When σ'_n becomes greater than zero and greater than the opening fracture stress *frs*, fracture permeability jumps from *khf* to fracture closure permeability *kccf* and then follows the Barton-Bandis model (curve FG) specified by equation (1). Path FG has an asymptotic value at *krcf* as shown by the dotted line. Thereafter, fracture permeability varies reversibly with σ'_n on path GFED.

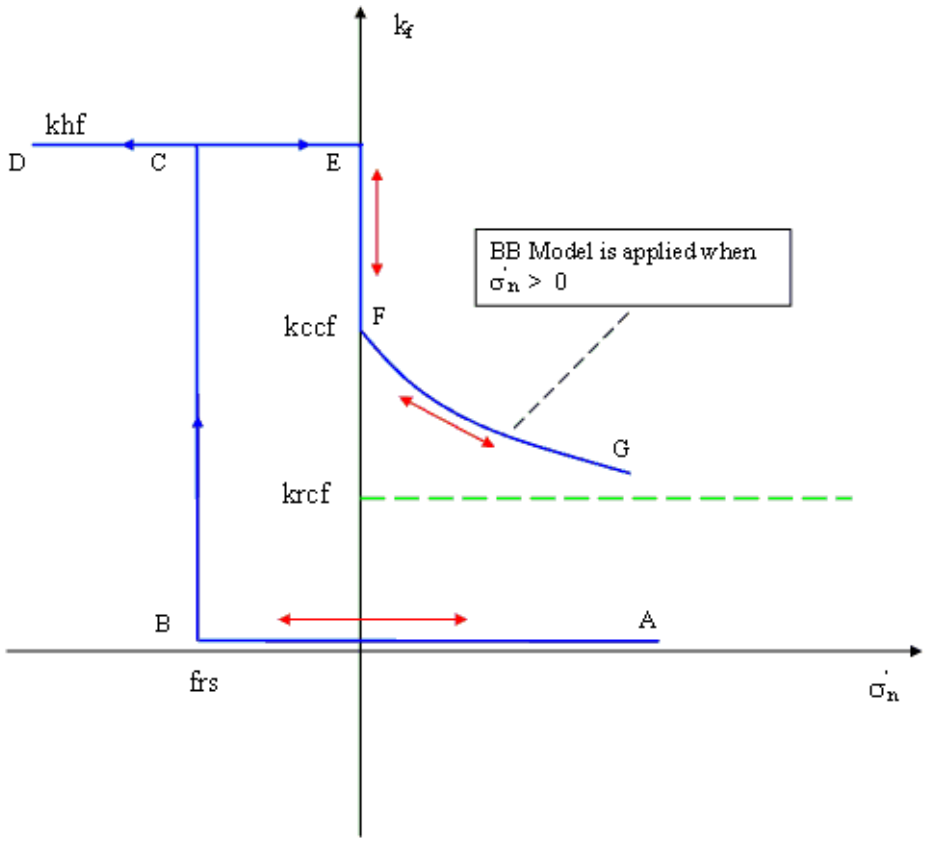


Diagram of fracture permeability

Examples:

```
*GEOROCK 1

. . .
** keyword   e0     kni    frs    khf   kccf  krcf
*GPERMBB   6.5e-5   3e6   -100.   100.   50.   5
. . .
*GEOROCK 2

. . .
** keyword   e0     kni    frs    khf   kccf  krcf
*GPERMBB   7.5e-5   4e6   -110.   120.   60.   3
*UNFIXFDIR
. . .
*GEOTYPE *KVAR 6*1 4*2  ** 10 layers
*GFRACBLK *IJK
   1:2 1:1 1:6  ** Fracture open from beginning
   1:2 1:2 7:10
```

Fracture Direction *FRACANGLE

PURPOSE:

Specify direction of fracture when the fracture permeability option *GPERMBB is used.

FORMAT:

*FRACANGLE
{ *IJK (*i1*(:*i2*) *j1*(:*j2*) *k1*(:*k2*) | *ALL) θ_x θ_y }

DEFINITIONS:

i1(:*i2*) *j1*(:*j2*) *k1*(:*k2*)
I-J-K address range specification.

*ALL
Specify all blocks.

θ_x
Angle in degrees between fracture and I (X) coordinate direction. The allowed range is from 0 to 90 degrees.

θ_y
Angle in degrees between fracture and J (Y) coordinate direction. The allowed range is from 0 to 90 degrees.

DEFAULTS:

For each grid block that is not explicitly assigned a fracture direction via *FRACANGLE, its fracture direction will be determined by the stress response on it.

CONDITIONS:

Each pair of angles θ_x and θ_y must satisfy the condition $\cos^2(\theta_x) + \cos^2(\theta_y) \leq 1$.

EXPLANATION:

When rock cracks the fracture direction depends strongly upon the direction of maximum principle effective stress acting on it. Use keyword *FRACANGLE to assign explicitly the fracture direction independent of stress response in the block. When this option is used, the fracture direction will remain as specified, until it is replaced with a new fracture direction via keyword *FRACANGLE in the recurrent data.

Multiple *IJK lines can be used after *FRACANGLE to specify a non-rectangular region or different angles for different regions.

Examples:

```
*FRACANGLE
   *IJK 1:10 5 1:2 60 40 ** Valid (sum cos2 = 0.837 < 1)
   *IJK 1:10 5 3:5 60 25 ** Invalid (sum cos2 = 1.071 > 1)
*FRACANGLE
   *IJK *ALL 60 60
```

Dilation Relative Permeabilities *RPWTABD, *RPLTABD

PURPOSE:

Define relative permeabilities for the dilated region.

FORMAT:

```
*RPWTABD
            sw      krw     krow
             :       :       :
*RPLTABD
            sl      krg     krog
             :       :       :
```

DEFINITIONS:

sw
: Water saturation. Table entries must increase, and must be separated by at least 0.001.

kr
: Relative permeability to water in the water/oil system. Table entries must increase.

krow
: Relative permeability to oil in the water/oil system. Table entries must decrease.

sl
: Liquid saturation. Table entries must increase, and must be separated by at least 0.001.

krg
: Relative permeability to gas in the liquid/gas system. Table entries must decrease.

krog
: Relative permeability to liquid in the liquid/gas system. Table entries must increase. The last entry must be equal to krow at critical water saturation.

DEFAULTS:

If the keywords *RPWTABD and *RPLTABD do not appear, the effect of remoulding on the relative permeability will not be taken into account.

CONDITIONS:

Both tables are optional if the plastic deformation option is used. See the *GEOMECH keyword.

These keywords are rock-type dependent, and are applied to the rock type number in effect at the time they are read. See the *GEOROCK and *GEOTYPE keywords.

EXPLANATION:

The relative permeability curves for the dilation zone may be used to account for the remoulding of the sand matrix as a result of plastic deformation. A grid block which has been subjected to plastic yielding will be considered to consist of the remoulded zone and the original rock matrix with a relatively low level of disturbances. It may be reasonable to assume that the remoulded zone tends to have a more linear relative permeability relationship. However, to avoid stability problems the relative permeability increase should not exceed about 20 times that of the original rock matrix.

By using a set of relative permeability curves identical to that of the original rock, the effect of remoulding on relative permeability will be removed.

Other Dilation Properties *FPVOLM, *TDMAX, *TDMIN

PURPOSE:

Assign limits of pore volume and transmissibility changes due to volumetric dilatation which includes the shear dilation component.

FORMAT:

*FPVOLM	*fpvolm*
*TDMAX	*tdmax*
*TDMIN	*tdmin*

DEFINITIONS:

fpvolm
: Maximum fractional change in the pore volume due to volumetric dilatation.

tdmax
: Maximum transmissibility increases due to volumetric dilatation.

tdmin
: Minimum transmissibility increases due to volumetric dilatation.

DEFAULTS:

*FPVOLM 0.05 *TDMAX 1 *TDMIN 1

CONDITIONS:

tdmin must not exceed *tdmax*.

EXPLANATION:

The volumetric dilatation of each grid block is calculated numerically by the finite-element stress-deformation analysis at each time step. Keyword *FPVOLM allows the user to limit the amount of volumetric dilatation.

Increase in transmissibility due to increase in the dilated zone is modelled via permeability multipliers of which *tdmin* and *tdmax* are the upper and lower bounds. Specifying *TDMAX 1 and *TDMIN 1 will result in no transmissibility increase. The relationship between transmissibility increase and pore-volume increase is as follows.

Let R be a block's fractional change in fluid pore volume due to volumetric dilatation, that is, $R = (V-V_i)/V_i$ where V_i is the initial pore volume and V is the current pore volume; *fpvolm* is the maximum allowed value of R. Let x = R/*fpvolm* be the block's scaled fractional change in pore volume, limited to the range [0,1]. The block's permeability multiplier is

$$tdmin + (tdmax - tdmin)(3x^2 - 2x^3)$$

which gives *tdmin* at x = 0, *tdmax* at x = 1 and zero slopes at both points. The multiplier of each block pair is applied to their respective permeabilities which are combined to give the transmissibility of the inter-block connection.

Note that no transmissibility variation is applied to diagonal directions if the *NINEPOINT grid option is used.

Well Radius *WRADIUS

PURPOSE:

Assign well radius where boundary conditions are to be applied for the axisymmetric radial grid. The desired model can be either the plasticity model or plasticity with well boundary unloading.

FORMAT:

*WRADIUS wrad

DEFINITIONS:

wrad
Well radius (m | ft).

DEFAULTS:

*WRADIUS 0

CONDITIONS:

Required only if the radial grid option is used.

EXPLANATION:

The plastic deformation model may be used with axisymmetric radial grid. In this case the well boundary is considered to be a rigid boundary, for example, a cased wellbore. When the well boundary unloading model is selected by using the *UNLOADSTR keyword, the boundary stress is assumed to be unloaded at the well radius by the amount specified using the *UNLOADSTR keyword.

Stiffness Matrix Calculation Option *STIFFINIT, *STIFFTANG, *STIFFCOM1, *STIFFCOM2

PURPOSE:

Assign stiffness matrix calculation option.

FORMAT:

*STIFFINIT | *STIFFTANG | *STIFFCOM1 | *STIFFCOM2

DEFINITIONS:

*STIFFINIT

Calculate the elasto-plastic stiffness matrix only once at the beginning of the time step.

*STIFFTANG

Calculate the elasto-plastic stiffness matrix at every iteration of every load increment of every time step.

*STIFFCOM1

Calculate the elasto-plastic stiffness matrix once at the beginning of every load increment of every time step and also whenever unloading is detected.

*STIFFCOM2

Calculate the elasto-plastic stiffness matrix at the beginning of the second iteration of each load increment of every time step, except for the first increment when the stiffness matrix is also calculated at the first iteration.

Unloading will also trigger the recalculation of the stiffness matrix.

DEFAULTS:

*STIFFTANG

CONDITIONS:

Only one of the stiffness matrix calculation options may be in use, i.e. they are mutually exclusive.

Deformation Solution Control *PRINTGEO, *NODE4, *NODE8, *NODE9, *GAUSSPNT, *NINCS, *FORCETOL, *DISPLACTOL, *NITERGEO, *STRESS_RETURN, *MAGNIFYDI, *MAGNIFYDJ, *MAGNIFYDK

PURPOSE:

Override default control parameters for plastic deformation option.

FORMAT:

*NODE4 | *NODE8 | *NODE9	
*GAUSSPNT	*ngaus*
*NINCS	*nincs*
*FORCETOL	*tol1*
*DISPLACTOL	*tol2*
*NITERGEO	*niter*
*PRINTGEO	*n*
*STRESS_RETURN	(*SUBSTEP | *GCPA)
*MAGNIFYDI	*magdi*
*MAGNIFYDJ	*magdj*
*MAGNIFYDK	*magdk*

DEFINITIONS:

*NODE4
: Use 4-node finite elements for 2D plane strain.

*NODE8
: Use 8-node finite elements for 2D plane strain.

*NODE9
: Use 9-node finite elements for 2D plane strain.

*GAUSSPNT *ngaus*
: Specify the number of Gaussian quadrature points for each local coordinate direction. The allowed range is 2 to 4.

*NINCS *nincs*
: Subdivide the load increment for each time step into *nincs* equal sub-increments. This will increase the accuracy of the plastic analysis at an increased computational cost. The allowed range for *nincs* is 1 to 10.

*FORCETOL *tol1*
: Relative convergence tolerance (expressed as a percentage) for force balance equations. The range is 10^{-10} to 1. *tol1* is the rms residual of force balance at all nodes acceptable as converged to an equilibrium condition. The unbalanced force is carried forward into the next time step. A value of 1 represents a rather loose tolerance whereas 10^{-10} is very tight.

*DISPLACTOL *tol2*

Convergence tolerance (m | ft) for the nodal displacement vector. The allowed range is 10^{-20} m to 10^{-2} m. The choice should be representative of the dimension of the model and should not be bigger than one percent of the length of the smallest element. A large value effectively turns off the displacement convergence criterion.

*NITERGEO *niter*

Maximum number of iterations allowed for solving the force balance equations. The minimum is 30.

*PRINTGEO *n*

Control the amount of output to the geomechanics output (.geo) file. The minimum that is written is a summary of input data and finite element information. *n* may be any number from 0 to 13, to produce the indicated additional output. *n* = 0 - 3 produce the same output as previous versions. The frequency of this output is determined by *WPRN *GRID.

n	***Additional output***
0	None
1, 2, 3, 4, 9, 11	Principle stresses
1, 2, 3, 7, 8, 10	Stress components, yield state; see also **Safety Factor** under *GOUTSRF
2, 3, 5, 8, 9, 12	Displacements
3, 6, 10, 11, 12	Reactions at constrained nodes
13	Mapping reservoir grids and geomechanical grids

*STRESS_RETURN

Specify the method used to compute stress return to the yield surface.

*SUBSTEP	This option is applicable to all elastoplastic constitutive models.
*GCPA	Use the Cutting Plane Algorithm (CPA) approach. This option is available only with Mohr-Coulomb (*MOHRCOUL) and Drucker-Prager (*DRUCKER) models.

*MAGNIFYDI *magdi*

Magnify the deformation in I direction: The real and positive value of *magdi* will magnify the deformation in this direction only.

*MAGNIFYDJ *magdj*

Magnify the deformation in J direction. The value of *magdj* is not necessary to be the same as that of *magdi*.

*MAGNIFYDK *magdk*

Magnify the deformation in K direction. The value of *magdk* is not necessary to be the same as that of *magdi* or *magdj*.

DEFAULTS:

*NODE4

*GAUSSPNT 2

*NINCS 1

*FORCETOL 0.1

*DISPLACTOL 0.0001

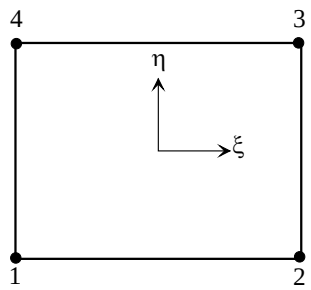
*NITERGEO 30

*PRINTGEO 0

*STRESS_RETURN *SUBSTEP

EXPLANATION:

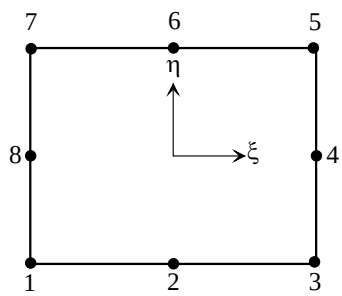
The nodal locations for each of the element types *NODE4, *NODE8 and *NODE9 and the associated element shape functions are shown in Figure below. The deformation can be magnified by using the keywords *MAGNIFYDI, *MAGNIFYDJ and *MAGNIFYDK in the corresponding directions. These keywords will not affect the computation but only help to visualize a deformed reservoir more clearly; especially for any reservoir with very small deformation. Without using these keywords, the deformation is not magnified. It should also be noted that the magnifying value must be positive.



4-node Isoparametric element

(a) 4-node element

$$N_i(\xi,\eta) = \frac{1}{4}(1+\xi\xi_i)\,(1+\eta\eta_i)$$



8-node Serendipity element

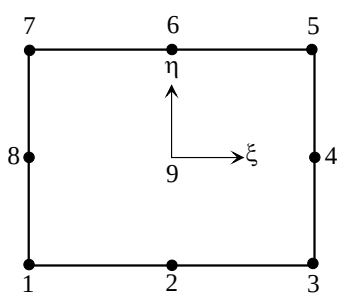
(b) 8-node element

for corner nodes

$$N_i^{(e)} = \frac{1}{4}(1+\xi\xi_i)\,(1+\eta\eta_i)(\xi\xi_i+\eta\eta_i-1),\ \ i = 1,3,5,7$$

for midside nodes

$$N_i^{(e)} = \frac{\xi_i^2}{2}(1+\xi\xi_i)\,(1-\eta^2)+\frac{n_i^2}{2}(1+\eta\eta_i)(1-\xi^2),\ i = 2,4,6,8$$



9-node Lagrangian element

(c) 9-node element

for corner nodes

$$N_i^{(e)} = \frac{1}{4}(\xi^2+\xi\xi_i)(\eta^2+\eta\eta_i),\ i = 1,3,5,7$$

for midside nodes

$$N_i^{(e)} = \frac{1}{2}\eta_i^2(\eta^2-\eta\eta_i)(1-\xi^2)+\frac{1}{2}\xi_i^2(\xi^2-\xi\xi_i)(1-\eta^2),\ i = 2,4,6,8$$

for central nodes

$$N_i^{(e)} = (1-\xi^2)\,(1-\eta^2)$$

Element types and shape functions available for geomechanical analysis

Geomechanics AIMSOL Control *SOLVERG, *PRECCG, *PRECABG, *NORTHG, *SORDERG, *SDEGREEG, *PIVOTG, *ITERMAXG, *ORTHOGG, *JDUMPG, *SITERPG

PURPOSE:

Indicate and control AIMSOL as the matrix solver for the geomechanics equations.

FORMAT:

*SOLVERG	(*AIMSOL | *AIMSOLN | *PGSOLV | *PGSOLN)
*PRECCG	*preccg*
*PRECABG	*precabs*
*NORTHG	*north*
*SORDERG	(*NATURAL | *REDBLACK | *RCM | *RCMRB)
*SDEGREEG	(*GAUSS | *ideg*)
*PIVOTG	(*ON | *OFF)
*ITERMAXG	*nitmax*
*ORTHOGG	(*CGS | *MGS) (*SINGLE | *DOUBLE)
*JDUMPG	
*SITERPG	

DEFINITIONS:

*SOLVERG (*AIMSOL | *AIMSOLN | *PGSOLV | *PGSOLN)

Specify which matrix solver to use on the linearized geomechanics equations.

*AIMSOL and *AIMSOLN use the AIMSOL matrix solution package. This option is required for *GEOM3D and recommended for large grids not using *GEOM3D. The AIMSOL package implemented here has control keywords that are separate but similar (with "G" appended) to the corresponding fluid-flow keywords found in the Numerical Control section. This allows you to specify AIMSOL control parameters for solving geomechanics equations that differ from the corresponding fluid-flow values.

*PGSOLV and *PGSOLN use a frontal solution technique in which the system is solved non-iteratively in a way which is equivalent to Gaussian elimination.

When a non-associated flow rule is used (see *DILANGLE) then *AIMSOLN must be used instead of *AIMSOL and *PGSOLN must be used instead of *PGSOLV. See **Non-Associated Flow** below.

*PRECCG *preccg*

Specify relative convergence tolerance for GMRES iterations (AIMSOL only).

*PRECABG *precabs*

Specify absolute convergence tolerance for the GMRES iterations (AIMSOL only).

*NORTHG *north*

Specify maximum number of GMRES orthogonalizations before iteration restart (AIMSOL only).

*SORDERG (*NATURAL | *REDBLACK | *RCM | *RCMRB)

Specify equation ordering strategy (AIMSOL only). Natural, red-black, reverse Cuthill-McKee, and reverse Cuthill-McKee followed by Red-Black are supported.

*SDEGREEG (*GAUSS | *ideg*)

Specify degree of LU factorization (AIMSOL only). *GAUSS indicates Gaussian elimination; otherwise, off-diagonal sub-matrices of degree greater than *ideg* are ignored. The initial non-zero off-diagonal sub-matrices are designated as degree 1.

*PIVOTG (*ON | *OFF)

Specify that pivoting is allowed in the inversion of diagonal sub-matrices (AIMSOL only).

*ITERMAXG *nitmax*

Specify maximum number of GMRES iterations allowed before returning to calling routine (AIMSOL only).

*ORTHOGG (*CGS | *MGS) (*SINGLE | *DOUBLE)

Specify orthogonalization method used in GMRES iterations: Classical (*CGS) or modified Gram-Schmidt (*MGS); and single or double orthogonalization (AIMSOL only).

*JDUMPG

The geomechanical linear system matrix is dumped to the ".geo" geomechanical output file (AIMSOL only).

*SITERPG

Information about GMRES dimensions and iterations is printed to the ".geo" output file (AIMSOL only; not for *CHECKONLY).

STARS: See the tutorial **Optimizing Memory Requirements**.

DEFAULTS:

Absent	Action	
*SOLVERG	*AIMSOL	(*GEOM3D present, *DILANGLE absent);
	*AIMSOLN	(*GEOM3D present, *DILANGLE present);
	*PGSOLV	(*GEOM3D absent, *DILANGLE absent);
	*PGSOLN	(*GEOM3D absent, *DILANGLE present).
*PRECCG	*PRECCG	10^{-6}
*PRECABG	*PRECABG	$5 \cdot 10^{-9}$
*NORTHG	*NORTHG	30

*SORDERG	*SORDERG	*NATURAL
*SDEGREEG	*SDEGREEG	2
*PIVOTG	*PIVOTG	*OFF
*ITERMAXG	*ITERMAXG	30
*ORTHOGG	*ORTHOGG	*CGS *SINGLE
*JDUMPG	(no dumping)	
*SITERPG	(no printing)	

CONDITIONS:

All keywords except *SOLVERG are valid only when *AIMSOL or *AIMSOLN is used.

*SOLVERG sub-keywords *PGSOLV and *PGSOLN may not be used with *GEOM3D.

*SOLVERG sub-keyword *AIMSOLN or *PGSOLN must be used when *DILANGLE is present.

EXPLANATION:

When a non-associated flow rule is used for an elasto-plastic material the stiffness matrix is unsymmetrical and requires an appropriate matrix solver. *SOLVERG options *PGSOLN and *AIMSOLN handle these unsymmetrical matrices. Use of another matrix solver option for unsymmetrical matrices may not give the correct result. Like *PGSOLV, *PGSOLN is available only for the 2D plane strain option (*GEOM3D absent). Like *AIMSOL, *AIMSOLN can handle 2D plane strain or 3D strain (*GEOM3D).

*PGSOLN and *AIMSOLN can solve associated-flow problems as well as the other matrix solution options but the storage requirement is larger. Therefore, these matrix solution options should be used only for cases with non-associated flow, as indicated by the default.

Dimension Over-Rides (Optional) *MPLNE, *MCONNG, *MDICLU_PG

PURPOSE:

Over-ride default dimension estimates obtained from the preliminary data scan.

FORMAT:

*MPLNE	*mplne*
*MCONNG	*mconng*
*MDICLU_PG	*mdiclu_pg*

DEFINITIONS:

mplne
: Maximum dimension for geomechanics array IPFRE.

mconng
: Maximum number of finite element connections for AIMSOL in Geomechanics.

mdiclu_pg
: Maximum number of AIMSOL LU submatrices in Geomechanics.

DEFAULTS:

Each of these keywords defaults to the value obtained from the data scan.

EXPLANATION:

Run-time dimensioning is designed to obtain all its needed information for storage allocation from a preliminary scan of the data. However, it is possible that several dimensioning parameters may be insufficient after this scan, in which case the user may enter values directly.

STARS: See also **Optimizing Memory Requirements** in the TUTORIAL chapter.

GEM: See also **Run-Time Dimensioning** in the I/O CONTROL chapter.

Initial Stress Distribution (2D) *STRESS, *STRESSGRAD, *STRESSALL, *STRESI, *STRESJ, *STRESK, *STRESSH

PURPOSE:

Assign the initial stress distribution for 2D plane strain approach.

FORMAT:

*STRESS	*sigma_y sigma_z sigma_yz sigma_x*
*STRESSGRAD	*strgrd_y strgrd_z strgrd_yz strgrd_x*
or	
*STRESS *IJK	
	{ *il*:*i2* *j1*:*j2* *k1*:*k2*
	sigma_y sigma_z sigma_yz sigma_x }
or	
*STRESSALL	
	{ *sigma_y sigma_z sigma_yz sigma_x* } (N lines)
or	
*STRESI *ALL	*sigma_x* ... (N values)
*STRESJ *ALL	*sigma_y* ... (N values)
*STRESK *ALL	*sigma_z* ... (N values)
*STRESSH *ALL	*sigma_yz* ... (N values)

DEFINITIONS:

sigma_y
: Y direction effective stress (kPa | psi).

sigma_z
: Z direction or vertical effective stress (kPa | psi).

sigma_yz
: YZ plane shear stress (kPa | psi).

sigma_x
: X direction effective stress (kPa | psi).

strgrd_y
: Y direction effective stress gradient (kPa/m | psi/ft).

strgrd_z
: Z direction effective stress gradient (kPa/m | psi/ft).

strgrd_yz
: YZ plane shear stress gradient (kPa/m | psi/ft).

strgrd_x
: X direction effective stress gradient (kPa/m | psi/ft).

il:*i2* *j1*:*j2* *k1*:*k2*
: Specify a group of fundamental blocks with I-J-K ranges.

DEFAULTS:

There is no default for initial stress.

CONDITIONS:

Initial stress is required if the plastic deformation option is used. See *GEOMECH.

EXPLANATION:

There are a number of methods for specifying initial stress distributions in the reservoir.

Uniform Distribution

Use *STRESS without *IJK to specify an initially uniform stress state to the entire reservoir.

Linear Variation with Depth

Use *STRESS without *IJK together with *STRESSGRAD to specify a linear variation of the individual stress components with depth. The values entered via *STRESS are assumed to be located at the centre of the stress reference block. This reference block is (1,1,1) by default but another block may be specified via keyword *GEORBLOCK. The initial Y-direction stress of a block is *sigma_y* + (*strgrd_y*) · Δh, where Δh is the depth of the block minus the depth of the reference block. Other stress components are computed in the same manner.

Uniform by Group

Use *STRESS *IJK to assign stresses by one or more groups of blocks. The syntax is similar to the *IJK array input option described in chapter Keyword Data Entry System, except that each I-J-K range is followed by four values. Each grid block that is not referenced by the *IJK ranges will have zero stress values.

Values for All Blocks, Stress Components Together

Use *STRESSALL to assign all four stress component values together on a line. There must be one line for each of N = ni · nj · nk fundamental grid blocks as specified via keyword *GRID *CART or *GRID *RADIAL. The order of blocks is natural order, that is, starting at (1,1,1), increasing the I index most frequently and the K index least frequently. For example: (1,1,1), (2,1,1), … (ni-1,nj,nk), (ni,nj,nk).

Values for All Blocks, Stress Components Separate

Use individual keywords *STRESI (I-direction), *STRESJ (J-direction), *STRESK (K-direction) and *STRESSH (shear) with *ALL to assign the corresponding stress component to each grid block. The number and ordering of blocks are the same as described for *STRESSALL. The syntax is similar to the *ALL array input option.

Radial Grid

The following associations exist between the radial and Cartesian stress components.

Radial	Y axis	sigma_y	*STRESJ *ALL
Axial (Z)	Z axis	sigma_z	*STRESK *ALL
Tangential	X axis	sigma_x	*STRESI *ALL
RZ plane	YZ plane	sigma_yz	*STRESSH *ALL

Initial Stress Distribution (3D) *STRESS3D, *STRESSGRAD3D, *STRESI, *STRESJ, *STRESK, *STRESSHIJ, *STRESSHJK, *STRESSHIK

PURPOSE:

Assign the initial stress distribution for 3D finite elements.

FORMAT:

*STRESS3D
  sigma_x sigma_y sigma_z sigma_xy sigma_yz sigma_xz
*STRESSGRAD3D
  strgrd_x strgrd_y strgrd_z strgrd_xy strgrd_yz strgrd_xz
or
*STRESS3D
  { *IJK *i1:i2 j1:j2 k1:k2*
    sigma_x sigma_y sigma_z sigma_xy sigma_yz sigma_xz }
or
*STRESS3D *ALL
  { *sigma_x sigma_y sigma_z sigma_xy sigma_yz sigma_xz* } (N lines)
or
*STRESS3D
*STRESI *ALL *sigma_xy* ... (N values)
*STRESJ *ALL *sigma_xy* ... (N values)
*STRESK *ALL *sigma_xy* ... (N values)
*STRESSHIJ *ALL *sigma_xy* ... (N values)
*STRESSHJK *ALL *sigma_yz* ... (N values)
*STRESSHIK *ALL *sigma_xz* ... (N values)

DEFINITIONS:

sigma_x
: X direction effective stress (kPa | psi).

sigma_y
: Y direction effective stress (kPa | psi).

sigma_z
: Z direction or vertical effective stress (kPa | psi).

sigma_xy
: XY plane shear stress (kPa | psi).

sigma_yz
: YZ plane shear stress (kPa | psi).

sigma_xz
: XZ plane shear stress (kPa | psi).

strgrd_x
: X direction effective stress gradient (kPa/m | psi/ft).

strgrd_y

Y direction effective stress gradient (kPa/m | psi/ft).

strgrd_z

Z direction effective stress gradient (kPa/m | psi/ft).

strgrd_xy

XY plane shear stress gradient (kPa/m | psi/ft).

strgrd_yz

YZ plane shear stress gradient (kPa/m | psi/ft).

strgrd_xz

XZ plane shear stress gradient (kPa/m | psi/ft).

i1:i2 j1:j2 k1:k2

Specify a group of fundamental blocks with I-J-K ranges.

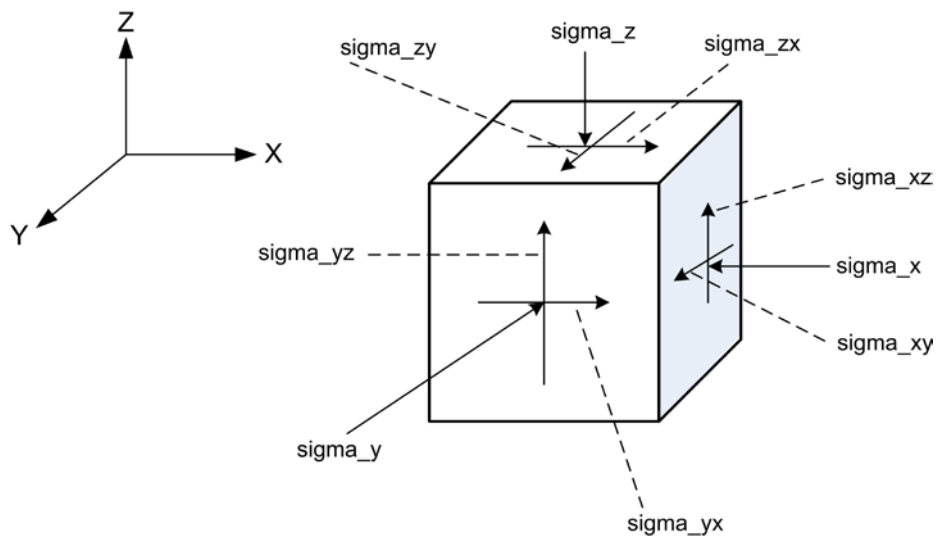
DEFAULTS:

There is no default for initialize stress.

CONDITIONS:

Initial stress is required if the plastic deformation option is used. See *GEOMECH.

Keyword *STRESS3D and subsequent values which indicate initial values of stresses on finite elements must be entered in a coupled data set. The convention of stresses on a block is shown in the figure.



In the above figure, sigma_x means normal stress on a surface perpendicular to the X axis, sigma_xy means that shear stress on a surface is perpendicular to the X axis and its traction direction is parallel to the Y axis. In addition, the shear stresses are symmetry, therefore:

sigma_yx = sigma_xy

sigma_zx = sigma_xz

sigma_zy = sigma_yz

EXPLANATION:

There are a number of methods for specifying initial stress distributions in the reservoir.

Uniform Distribution

Use *STRESS3D without *IJK to specify an initially uniform stress state to the entire reservoir.

Linear Variation with Depth

Use *STRESS3D without *IJK but with *STRESSGRAD3D to specify a linear variation of the individual stress components with depth. The values entered via *STRESS3D are assumed to be located at the centre of the stress reference block. This reference block is (1,1,1) by default but another block may be specified via keyword *GEORBLOCK. The initial Y-direction stress of a block is *sigma_y* + (*strgrd_y*) · Δh, where Δh is the depth of the block minus the depth of the reference block. Other stress components are computed in the same manner.

Uniform by Group

Use *STRESS3D *IJK to assign stresses by one or more groups of blocks. The syntax is similar to the *IJK array input option described in chapter Keyword Data Entry System, except that each I-J-K range is followed by six values. Each grid block that is not referenced by the *IJK ranges will have zero stress values.

Values for All Blocks, Stress Components Together

Use *STRESS3D *ALL to assign all six stress component values together on a line. There must be one line for each of N = ni · nj · nk fundamental grid blocks as specified via keyword *GRID *CART or *GRID *RADIAL. The order of blocks is natural order, that is, starting at (1,1,1), increasing the I index most frequently and the K index least frequently. For example: (1,1,1), (2,1,1), … (ni-1,nj,nk), (ni,nj,nk).

Values for All Blocks, Stress Components Separate

Use *STRESS3D followed by individual keywords *STRESI (I-direction), *STRESJ (J-direction), *STRESK (K-direction), *STRESSHIJ (XY plane shear), *STRESSHJK (YZ plane shear) and *STRESSHIK (XZ plane shear) with *ALL to assign the corresponding stress component to each grid block. The number and ordering of blocks are the same as described for *STRESS3D *ALL. The syntax is similar to the *ALL array input option.

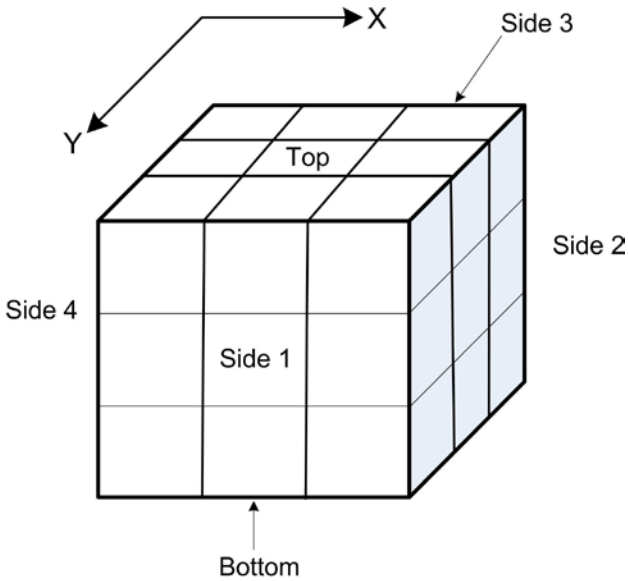
Sign of Stress Gradient

When the grid is defined with *KDIR UP, stress gradients must be positive; for *KDIR *DOWN they must be negative.

Example 1:

Consider a reservoir having 3x3x3 grids as shown in the below figure. Assuming that normal effective stresses in three directions on the first layer of the reservoir are 5000 psi, stresses on the second layer are 5500 psi and stresses on the third layer are 6000 psi. The keywords entered in a data set are expressed as:

```
*STRESS3D
*IJK   1:3     1:3     1            ** layer 1
 5000       5000      5000      0    0    0
*IJK   1:3    1:3      2            ** layer 2
 5500       5500      5500      0    0    0
*IJK   1:3     1:3     3            ** layer 3
 6000       6000      6000      0    0    0
```



The above stresses can also be expressed as:

```
*STRESS3D *ALL
           5000   5000   5000   0   0   0      **Start first layer
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0
           5000   5000   5000   0   0   0       **End first layer
           5500   5500   5500   0   0   0       **Start second layer
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0
           5500   5500   5500   0   0   0       **End second layer
```

```
6000   6000   6000   0   0   0       **Start third layer
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0
6000   6000   6000   0   0   0       **End third layer
```

The above example can also be expressible as:

```
*STRESS3D
*STRESI *ALL
5000  5000  5000  5000  5000  5000  5000  5000  5000
5500  5500  5500  5500  5500  5500  5500  5500  5500
6000  6000  6000  6000  6000  6000  6000  6000  6000
*STRESJ *ALL
5000  5000  5000  5000  5000  5000  5000  5000  5000
5500  5500  5500  5500  5500  5500  5500  5500  5500
6000  6000  6000  6000  6000  6000  6000  6000  6000
*STRESK *ALL
5000  5000  5000  5000  5000  5000  5000  5000  5000
5500  5500  5500  5500  5500  5500  5500  5500  5500
6000  6000  6000  6000  6000  6000  6000  6000  6000
```

Example 2:

If all the grid blocks have the same normal stresses, for instance 5000 psi in each direction. The keyword is entered in the data set as:

```
*STRESS3D    5000    5000    5000    0   0    0
```

Example 3:

Assuming the stress gradient is 5 psi/ft in the normal directions. The keyword is entered as:

```
*KDIR *UP …
*STRESSGRAD3D      5    5    5     0    0    0
** OR
*KDIR *DOWN …
*STRESSGRAD3D     -5   -5   -5     0    0    0
```

Since shear stresses are zero the corresponding keywords are absent.

Geomechanical Reference Block *GEORBLOCK

PURPOSE:

Specify the reference block for stress gradient option.

FORMAT:

*GEORBLOCK *i1* *j1* *k1*

DEFINITION:

i1
Index of the reference block in X direction.

j1
Index of the reference block in Y direction.

k1
Index of the reference block in Z direction.

DEFAULT:

If keyword *GEORBLOCK is absent, the reference block is the first non-null block in natural ordering (I-direction fastest, K-direction slowest).

EXPLANATION:

Keyword *GEORBLOCK is designed so that the stress gradient can be determined when a reference grid block is selected among the reservoir grid blocks. Each direction index must lie within the fundamental grid indices defined by keyword *GRID in the Reservoir Description chapter.

Example:

```
*STRESSGRAD  5  5  0  5
*GEORBLOCK   1  5  1     **Reference block i=1, j=5, k=1
```

Prescribed Boundary Conditions (2D) *PRESCBC

PURPOSE:

Prescribe displacement-type boundary conditions on nodal points of a finite element.

FORMAT:

*PRESCBC
{ node1(:node2) direction displacement }
or
*PRESCBC
{ *PLANE *p1(:p2)*
{ node1(:node2) direction displacement } }
or
*PRESCBC
{ *IJK *i1:i2 j1:j2 k1:k2*
{local node1(:local node2) direction displacement}}

DEFINITIONS:

*PRESCBC

Enable prescribed displacement boundary conditions. This keyword removes the default "essential" boundary conditions where nodes are constrained on the sides and bottom of the reservoir, and prepares for the user's prescribed displacement-type boundary conditions table.

node1(:*node2*)

Single nodal point (or range of nodal points) of a finite element to which the boundary condition is assigned. For the case of *KDIR *DOWN (as shown in the figure below) the nodal points on finite elements are numbered from **top to bottom** and from left to right. In the case of *KDIR *UP, the nodal points are numbered from **bottom to top** and from left to right.

Note that an element node that is not associated with any active element (grid block) is not included in the node ordering. This occurs when a node lies in the interior of a region of null blocks. Since the distribution of null blocks can differ from plane to plane, node ordering and total nodes may differ also between planes.

*IJK

Keyword for reservoir grid blocks

i1:*i2*

Starting block index *i1* to ending block index *i2* in X direction

j1:*j2*

Starting block index *j1* to ending block index *j2* in Y direction

k1:*k2*

Starting block index *k1* to ending block index *k2* in Z direction

local node1(:*local node2*)

Local nodal point (or a range of local nodal points) of a 2D finite element on which the prescribed boundary conditions are given. These local nodal points are strongly dependent on keywords *KDIR UP, *KDIR *DOWN and *PLSTRAINY. The convention of nodal points is given in the explanation.

direction

Direction of the displacement at a nodal point, given as follows:

01	displacement along the horizontal direction
02	displacement along the vertical direction
01:02	displacements in both directions

displacement

Value of the displacement (m | ft) at the indicated boundary nodal point(s) in the indicated direction(s).

*PLANE *p1*(:*p2*)

Specify to which plane the following set of boundary conditions are to be applied. When this keyword is absent, the prescribed boundary conditions are applied to all planes.

Plane number *p1* (or each number *p1* to *p2*) corresponds to the grid index in the direction normal to the computed strain planes. There are two cases:

1. Keyword *PLSTRAINY is absent: the computed strain planes are normal to the I (X) direction; *p1* is the I index and ranges from 1 to *ni* (see keyword *GRID).

2. Keyword *PLSTRAINY is present: the computed strain planes are normal to the J (Y) direction; *p1* is the J index and ranges from 1 to *nj* (see keyword *GRID).

CONDITIONS:

If the keyword *PRESCBC is entered in a data set, prescribed boundary conditions must be given.

DEFAULTS:

If keyword *PRESCBC is absent, constraints on the left, right and bottom of a plane are used.

If sub-keyword *PLANE is absent, all the planes have the same prescribed boundary conditions.

EXPLANATION:

Keyword *PRESCBC allows you to assign prescribed boundary conditions at any node on any finite element and on any plane which is cut from the reservoir either along the I or along the J direction.

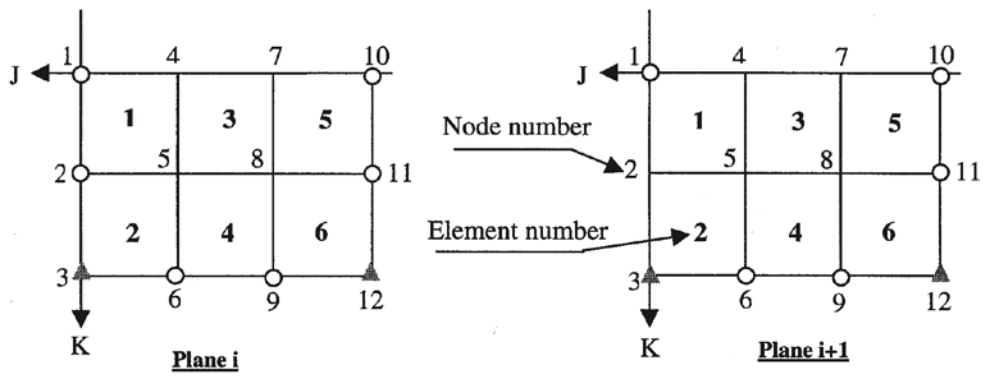
If keyword *PRESCBC is present, prescribed boundary conditions must be supplied for appropriate nodes on each plane; otherwise the boundary nodes of finite elements are not constrained by any conditions. There are two approaches:

First approach: Using global nodal points.

In this approach, finite elements as well as their nodal points on a plane must be determined in advance before the prescribed boundary conditions can be assigned on those nodal points. This approach is quite uncomfortable when counting elements and node numbers for a reservoir having complicated geometry. This approach is used for 2D finite elements only.

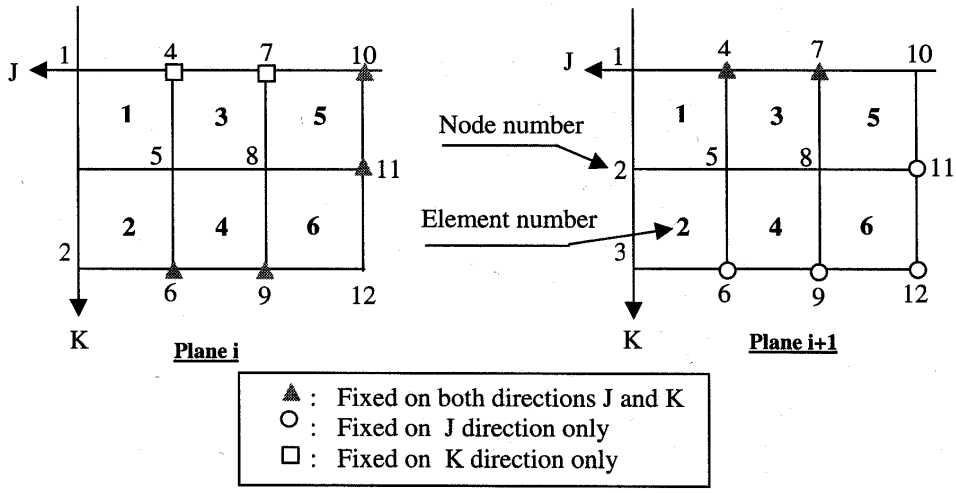
Example:

Consider two planes i and i+1. Without keyword *PRESCBC, default boundary conditions are shown in the following figure.



With the following *PRESCBC data, prescribed boundary conditions are given on plane i and plane i+1 as shown in the following figure.

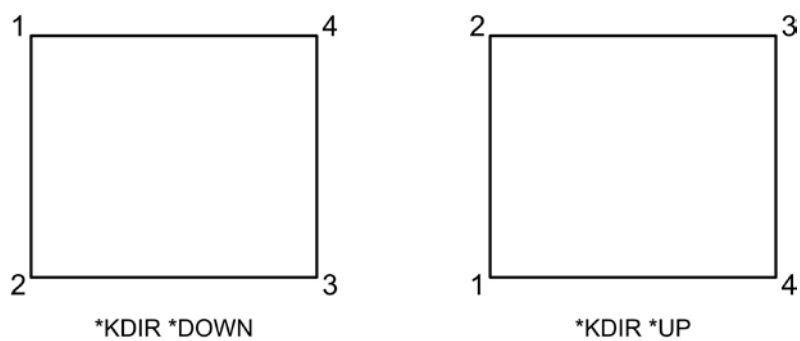
```
*PRESCBC
  *PLANE   i
  ** Node      Direction    Displacement
      4           02            0.0
      6         01:02           0.0
      7           02            0.0
     9:11       01:02           0.0
  *PLANE   i+1
  ** Node      Direction    Displacement
      4         01:02           0.0
      6           01            0.0
      7         01:02           0.0
      9           01            0.0
    11:12         01            0.0
```



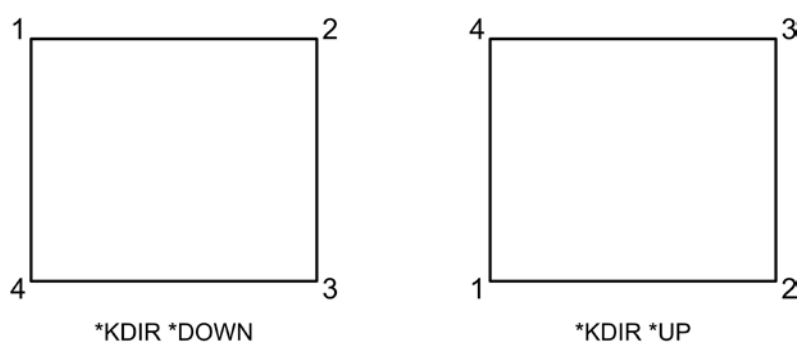
Second approach: Using local nodal points of a finite element.

This approach uses the reservoir block address in terms of I, J, K to assign the boundary conditions on local nodal points of a finite element. The approach is an easy and straightforward way compared to the first approach. In this approach, the user does not have to count the node and element number in advance. As mentioned above, numbering the local nodal points on a 2D finite element are strongly dependent on keywords *KDIR *UP, *KDIR DOWN and *PLSTRAINY. The nodal points for each case are illustrated in following figures.

Without keyword *PLSTRAINY

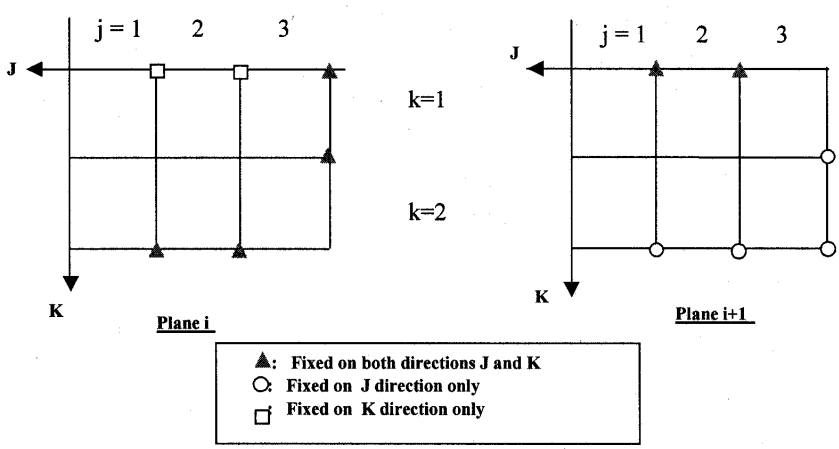


With keyword *PLSTRAINY



It should be noted that when *GRID *RAD is used, the keyword *PLSTRAINY is automatically applied. Therefore, the notation of local nodal points is shown above must be taken into account.

Taking the above example, on plane i and plane i+1, NJ = 3 and NK = 2. Keyword *KDIR DOWN is used and keyword *PLSTRAINY is absent that means the plane is perpendicular to the I direction. The prescribed boundary conditions are expressed as:




```
*PRESCBC
** For plane i
*IJK    i    1    1    ** j = 1, k = 1
            **local node         direction       displacement
                 4                  02               0.0
*IJK    i    2    1    ** j = 2, k = 1
            **local node         direction       displacement
                 1                  02               0.0
                 4                  02               0.0
*IJK    I    3    1    ** j = 3, k = 1
            **local node         direction       displacement
                 1                  02               0.0
                 3:4               01:02             0.0
*IJK    I    1    2    ** j = 1, k = 2
            **local node         direction       displacement
                 3                 01:02             0.0
*IJK    I    2    2    ** j = 2, k = 2
            **local node         direction       displacement
                 2:3               01:02             0.0
*IJK    I    3    2    ** j = 3, k = 2
            **local node         direction       displacement
                 2                 01:02             0.0
                 4                 01:02             0.0
** For plane i+1
*IJK    i+1    1    1    ** j = 1, k = 1
            **local node         direction       displacement
                 4                 01:02             0.0
*IJK    i+1    2    1    ** j = 2, k = 1
            **local node         direction       displacement
                 1                 01:02             0.0
                 4                 01:02             0.0
```

```
*IJK    i+1    3    1    ** j = 3, k = 1
            **local node         direction       displacement
                  1                01:02              0.0
                  3                 01                0.0
*IJK    i+1    1    2    ** j = 1, k = 2
            **local node         direction       displacement
                  3                 01                 0.0
*IJK    i+1    2    2    ** j = 2, k = 2
            **local node         direction       displacement
                 2:3                01                 0.0
*IJK    i+1    3    2    ** j = 3, k = 2
            **local node         direction       displacement
                 2:4                01                 0.0
```

Casing Treatments with Radial Grids

See "Casing Treatments with Radial Grids" in section "Summary of Geomechanic Model" at the beginning of this chapter.

Prescribed Boundary Conditions (3D) *PRESCBC3D

PURPOSE:

Prescribed displacement-type boundary conditions on a nodal point of a 3D finite element.

FORMAT:

*PRESCBC3D
*IJK *i1:i2* *j1:j2* *k1:k2*
node1(:node2) *direction* *displacement*

DEFAULT:

If keyword *PRESCBC3D is absent, the default ‘essential’ boundary condition is applied.

DEFINITION:

*PRESCBC3D
: Similar to the keyword *PRESCBC, the keyword *PRESCBC3D is applied for a case of 3D finite elements. The keyword removes all the essential boundary constraints on a reservoir and prepares for the input of prescribed displacement-type boundary constraints on finite element nodes.

*IJK
: Keyword for reservoir grid blocks.

i1:i2
: Starting block index *i1* to ending block index *i2* in X direction

j1:j2
: Starting block index *j1* to ending block index *j2* in Y direction

k1:k2
: Starting block index *k1* to ending block index *k2* in Z direction

node1(:node2)
: Local nodal point (or a range of nodal points) of a finite element to which the boundary condition is assigned. *KDIR *UP and *KDIR *DOWN are taken into account when a local nodal point is used for prescribed boundary conditions.

direction
: Direction of the displacement at a local nodal point is given as:

01 displacement along I direction

02 displacement along J direction

03 displacement along K direction

01:03 displacement in three directions.

displacement

Value of displacement (m | ft) at the indicated boundary local nodal point(s) in the indicated direction(s). A positive value means the displacement has the same directions as that of the coordinate, A negative value means the displacement has the opposite direction to the direction of the coordinate.

CONDITIONS:

Keyword *PRESCBC3D may be used only if *GEOM3D is present.

Keyword *PRESCBC3D may not appear more than once, so all prescribed displacement-type boundary conditions must be specified with one *PRESCBC3D keyword.

DEFAULTS:

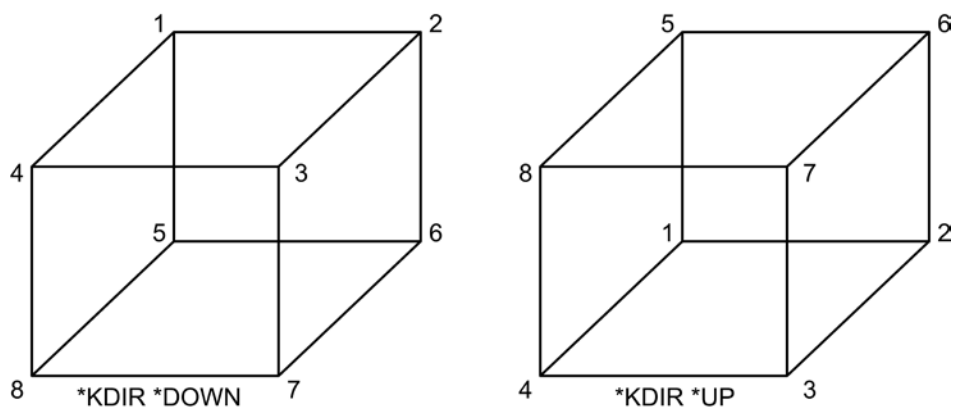
If keyword *PRESCBC3D is absent, the following grid boundaries are constrained:

Radial grid:	Bottom only
All other grids:	Bottom, left, right, front, back (all except top)

Individual nodes that are not constrained will move freely in space.

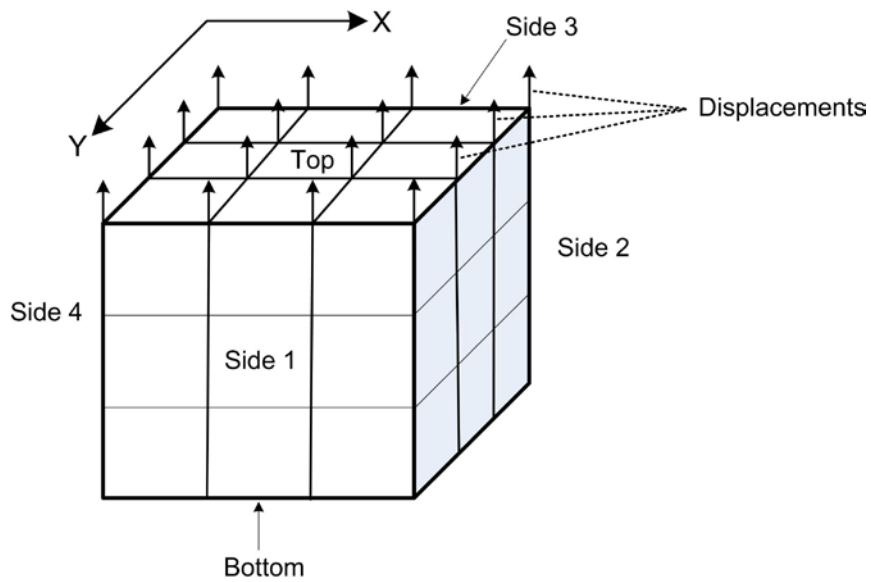
EXPLANATION:

Beside the essential boundary conditions which are default, the optional keyword *PRESCBC3D is designed so that users can select any grid block of a reservoir and assign a displacement value on it. When the keyword is entered in a data set, sub-keyword *IJK must also be accompanied to indicate the constrained block. Following that local node number, direction of the displacement and value of the displacement must be given. If the keyword *PRESCBC3D appears and the sub-keyword *IJK as well as prescribed displacement are not entered, the reservoir is moving freely. The solution, therefore, cannot be predicted. The local node numbers of a finite element also depend on the keyword *KDIR *UP or *KDIR *DOWN. The local node numbers used in the module are shown as follows:



Example:

Consider a reservoir field which has NI = 3, NJ = 3, NK = 3 as shown in the figure below:



Assuming the reservoir is constrained all its sides except the top side; the boundary conditions are expressed as follows:

For a case of *KDIR *DOWN

```
*PRESCBC3D     ** Keyword for prescribed boundary condition
    **Bottom
*IJK  1:3  1:3    3  ** Keyword for grid block and I,J,K block
                     ** local node   direction    displacement
      5:8   03  0.0  ** nodes 5, 6, 7, 8  are constrained in Z
direction
    ** Side 1
*IJK  1:3  3   1:3
                      ** local node   direction    displacement
                             3:4         02            0.0
                             7:8         02            0.0
    ** Side 2
*IJK  3  1:3   1:3
                      ** local node   direction    displacement
                             2:3         01            0.0
                             6:7         01            0.0
     ** Side 3
*IJK  1:3  1   1:3
                       ** local node   direction    displacement
                             1:2         02            0.0
                             5:6         02            0.0
    ** Side 4
*IJK  1  1:3   1:3
                          ** local node   direction    displacement
                                 1          01             0.0
                                 4          01             0.0
                                 5          01             0.0
                                 8          01             0.0
```

```
For a case of *KDIR *UP
*PRESCBC3D            ** Keyword for prescribed boundary condition
  **Bottom
*IJK  1:3   1:3    1  ** Keyword for grid block and I,J,K block
                      ** local node   direction    displacement
      1:4   03   0.0  ** nodes 1, 2, 3, 4  are constrained in Z
direction
    ** Side 1
*IJK 1:3   3   1:3
                          ** local node   direction    displacement
                                3:4          02            0.0
                                7:8          02            0.0
    ** Side 2
*IJK  3   1:3   1:3
                          ** local node   direction    displacement
                                2:3          01            0.0
                                6:7          01            0.0
     ** Side 3
*IJK   1:3   1   1:3
                          ** local node   direction    displacement
                                1:2          02            0.0
                                5:6          02            0.0
     ** Side 4
*IJK   1   1:3   1:3
                          ** local node   direction    displacement
                                  1          01            0.0
                                  4          01            0.0
                                  5          01            0.0
                                  8          01            0.0
Assuming that the top of reservoir is pulled by an amount of 0.5 ft (or
m) indicated by the arrows on the top, the prescribed boundary
conditions at those nodal points are given as:
For *KDIR *DOWN
*IJK   1:3   1:3   1
                           ** local node   direction    displacement
                               1:4            03           -0.5
       ** minus sign since its direction is opposite to Z direction
For *KDIR *UP
*IJK   1:3   1:3   3
                           ** local node   direction    displacement
                               5:8            03            0.5
       ** plus sign not minus sign
```

Casing Treatments with Radial Grids

See "Casing Treatments with Radial Grids" in section "Summary of Geomechanic Model" at the beginning of this chapter.

Point Loads (2D) *PLOADBC

PURPOSE:

Specify external loads on any nodal points of a finite element.

FORMAT:

```
*PLOADBC
   { node1(:node2) I/J_load K_load }
or
*PLOADBC
   { *PLANE  p1(:p2)
      { node1(:node2) I/J_load K_load } }
or
*PLOADBC
   { *IJK     i1:i2     j1:j2     k1:k2
      {local  node1(:local node2)   I/J_load   K_load } }
```

DEFINITIONS:

*PLOADBC
: This keyword specifies external loads applied to nodal points.

node1(:*node2*)
: Single nodal point (or range of nodal points) of a finite element to which the boundary condition is assigned. See the description for *PRESCBC.

*IJK
: Keyword for reservoir grid blocks

i1:i2
: Starting block index *i1* to ending block index *i2* in X direction

j1:j2
: Starting block index *j1* to ending block index *j2* in Y direction

k1:k2
: Starting block index *k1* to ending block index *k2* in Z direction

local node1(:*local node2*)
: Local nodal point (or a range of local nodal points) of a 2D finite element on which the loads are applied. These local nodal points are strongly dependent on keywords *KDIR UP, *KDIR *DOWN and *PLSTRAINY. The convention of nodal points is given in the explanation.

I/J_load

Load (kN | tonf) applied to the indicated node(s) in (1) the I direction when strains are computed in planes normal to the J direction (*PLSTRAINY is present), and (2) the J direction when strains are computed in planes normal to the I direction (*PLSTRAINY is absent). The load value is positive or negative depending on the actual load direction relative to the grid direction. Enter a zero value if no load is applied to the node(s) on this direction (e.g., only *K_load* is non-zero).

K_load

Load (kN | tonf) applied to the indicated node(s) in the K direction. The load value is positive or negative depending on the actual load direction relative to the grid direction. Enter a zero value if there is no load applied to the node(s) (e.g., only *I/J_load* is zero).

*PLANE *p1*(:*p2*)

See the description for *PRESCBC.

DEFAULTS:

If keyword *PLOADBC is absent then all external loads are zero.

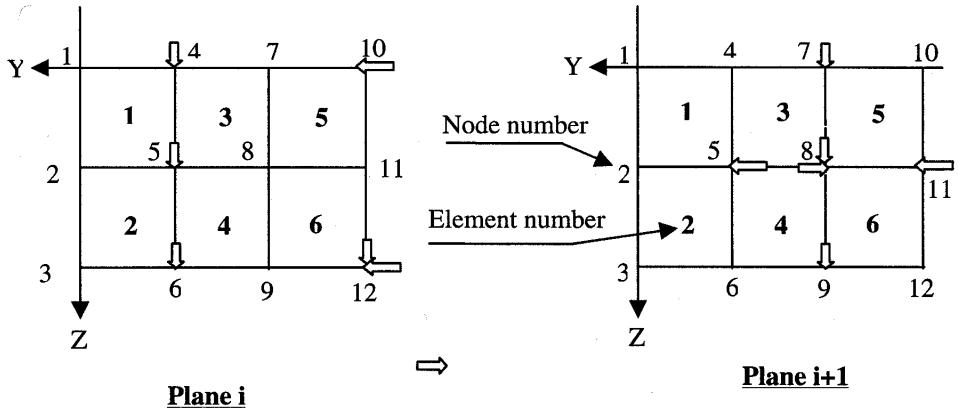
If sub-keyword *PLANE is absent, all the planes have the same prescribed boundary conditions.

EXPLANATION:

Keyword *PLOADBC allows you to assign external loads at any node on any finite element and on any plane which is cut from the reservoir either along the I or along the J direction. Similar to the optional keyword *PRESCBC, there are two different approaches which are used to describe the loading boundary as follows:

First approach: Using global nodal points

Example:



The point loads at nodes as showed in the above figures is specified as follows.

```
*PLOADBC
  *PLANE  i
  ** Node     Y Load    Z Load
      4:6       0        100
      10       15         0
      12       10        10
  *PLANE  i+1
  ** Node     Y Load    Z Load
      5        100        0
      7         0        10
      8        -10       10
      9         0        15
      11       15         0
```

Note that a zero value is entered when no load is applied.

<u>Second approach</u>: Using I, J, K block and local nodal points of a finite element. For convention of local nodal points of a 2D finite element, please see keyword *PRESCBC.

The point loads as seen in the above figure are expressed as:

```
*PLOADBC
** on plane i

*IJK    I    1    1    ** j = 1, k = 1
           **local node         Y load       Z load
                3:4                0          100
*IJK    I    2    1    ** j = 2, k = 1
           **local node         Y load       Z load
                1:2                0          100
*IJK    I    3    1    ** j= 3, k = 1
           **local node         Y load       Z load
                 4                15            0
*IJK    I    1    2    ** j= 1, k = 2
           **local node         Y load       Z load
                3:4                0          100
*IJK    I    3    1    ** j= 2, k = 2
           **local node         Y load       Z load
                1:2                0          100
*IJK    I    3    2    ** j= 3, k = 2
           **local node         Y load       Z load
                 3                10           10
** on plane i+1
*IJK    i+1    1    1    ** j = 1, k = 1
           **local node         Y load       Z load
                3                100           0
*IJK    i+1    2    1    ** j = 2, k = 1
           **local node         Y load       Z load
                     2              100          0
                     3              -10         10
                     4                0         10
*IJK    i+1    3    1    ** j= 3, k = 1
           **local node         Y load       Z load
                     1                0           10
                     2              -10           10
                     3               15            0
```

```
*IJK    i+1    1    2    ** j= 1, k = 2
           **local node         Y load       Z load
                  4              100             0
*IJK    i+1    3    1    ** j= 2, k = 2
           **local node         Y load       Z load
                  1              100             0
                  3                0            15
                  4              -10            10
*IJK    i+1    3    2    ** j= 3, k = 2
           **local node         Y load       Z load
                  1              -10            10
                  2                0            15
                  4               15             0
```

Point Loads (3D) *PLOADBC3D

PURPOSE:

Specify external load on any nodal points of a 3D finite element.

FORMAT:

```
*PLOADBC3D
  {*IJK          i1:i2   j1:j2   k1:k2
        node1(:node2)  I_load   J_load   K_load }
```

DEFINITIONS:

*PLOADBC3D
: The keyword indicates external loads applied to nodal points.

*IJK
: Keyword for reservoir grid blocks

i1:i2
: Starting block index *i1* to ending block index *i2* in X direction

j1:j2
: Starting block index *j1* to ending block index *j2* in Y direction

k1:k2
: Starting block index *k1* to ending block index *k2* in X direction

node1(:node2)
: Single local nodal point (or range of nodal points) of a finite element to which the load is assigned. See the keyword *PRESCBC3D for details of local nodal point.

I_load
: Load (kN | tonf) applied in the I direction.

J_load
: Load (kN | tonf) applied in the J direction.

K_load
: Load (kN | tonf) applied in the K direction.

DEFAULTS:

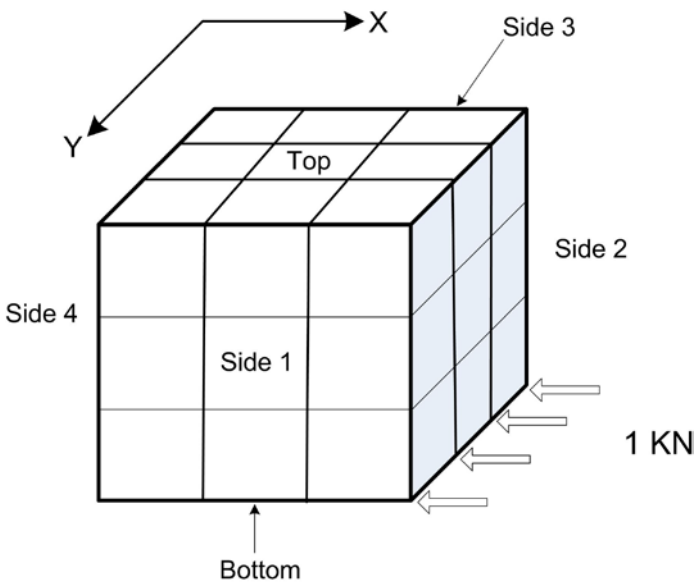
If the keyword *PLOADBC3D is absent, there is no external load.

EXPLANATION:

Similar to the keyword *PLOADBC for 2D plane strain, the optional keyword *PLOADBC3D is applied for 3D finite element. With this keyword, the user can assign external loads on any nodal points of a 3D finite element. The load value is positive or negative in one direction depending on the actual load direction relative to the grid direction. Values of I_load, J_load and K_load must be entered in the table even though they are zero.

External loads expressed on a finite element also strongly depends on *KDIR *UP and *KDIR *DOWN.

Example:



Assuming external loads 1 KN are applied at the bottom edge on side 2 of the above reservoir as illustrated by arrows. Expressing the above loads in a data set is shown as:

```
For *KDIR *DOWN
*PLOADBC3D
*IJK   3      1:3      3       ** I = 3, J=1 to J=3, K = 3
    ** node     I_load   J_load   K_load
      6:7       -1.0       0.0      0.0
           ** negative load due to opposite direction to X
For *KDIR *UP
*PLOADBC3D
*IJK   3      1:3      1       ** I = 3, J=1 to J=3, K = 1
    ** node     I_load   J_load   K_load
      2:3       -1.0      0.0       0.0
           ** negative load due to opposite direction to X
```

Distributed Edge Loads (2D) *DLOADBC, *DLOADBC2D

PURPOSE:

Specify distributed loads along an edge of a finite element.

FORMAT:

*DLOADBC *or* *DLOADBC2D
{ *ELEMENT *ele*
{ *en1*:*en2* *NL1* *TL1* *NL2* *TL2* } }

or

*DLOADBC *or* *DLOADBC2D
{ *PLANE *p1*
{ *ELEMENT *ele*
{ *en1*:*en2* *NL1* *TL1* *NL2* *TL2* } } }

or

*DLOADBC *or* *DLOADBC2D
{ *IJK *i1*:*i2* *j1*:*j2* *k1*:*k2*
{ *en1*:*en2* *NL1* *TL1* *NL2* *TL2* } } }

DEFINITIONS:

*DLOADBC

Specify distributed loads for finite element edges where 2D plane strain is considered as a thin plate. The unit of loading quantities is (kN/m | tonf/ft).

*DLOADBC2D

Specify distributed loads for finite element edges for 2D axi-symmetric problems. The specified edge loading is multiplied by the entire circumference to get distributed loading per unit area. The unit of loading quantities is (kPa/m | psi/ft).

*ELEMENT *ele*

Element number. The allowed a range is 1 to the maximum number of non-null elements in a plane. The definition of null elements is the same as that of null grid blocks in the reservoir section. Null elements are treated as non-existing elements.

en1:*en2* *NL1* *TL1* *NL2* *TL2*

en1:*en2* are the starting and ending nodes, respectively, of an edge of finite element *ele*. An edge is defined by two nodes on the element. More than one of these data lines may follow *ELEMENT *ele*.

The remaining four quantities are load per unit length. Enter zero for a quantity for which no load is specified.

NL1	normal load at *en1*
TL1	tangential load at *en1*
NL2	normal load at *en2*
TL2	tangential load at *en2*

*PLANE *p1*

See the description for *PRESCBC. Only one plane number is allowed.

DEFAULTS:

If sub-keyword *PLANE is absent, all planes have the same distributed load.

EXPLANATION:

Keyword *DLOADBC allows you to distribute loads on edges of finite elements. An edge is determined by two sequential nodes on the element. There are 4 edges on an element constituted by 4 nodes, and there are 8 edges on an element constituted by 8 nodes or 9 nodes. Similar to prescribed boundary conditions and point loads, there are also two different approaches to define the distributed load on the edges of a finite element.

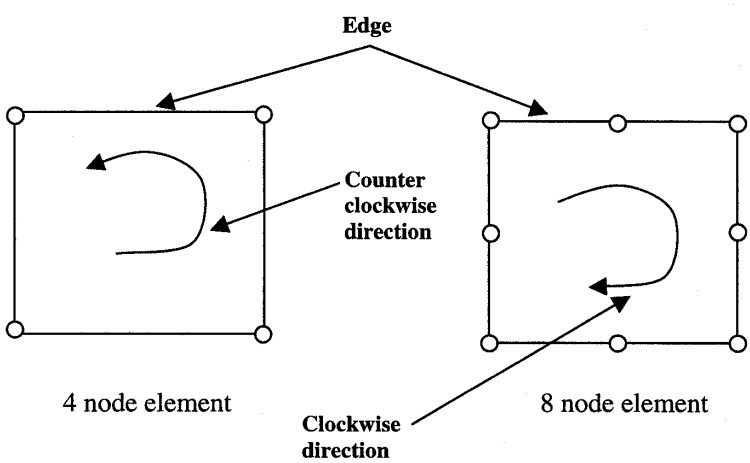
First approach: Based on global nodal points on a plane.

In this approach, rules for assigning starting nodes and ending nodes correctly are:

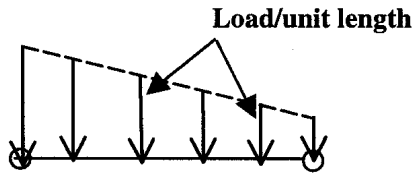
i) *For plane strains along I direction*: starting nodes and ending nodes must be entered in the counter clockwise direction.

ii) *For plane strains along J direction:* (i.e. when keyword *PLSTRAINY appears): starting nodes and ending nodes must be entered in the clockwise direction.

iii) *For axis-symmetric case*: starting nodes and ending nodes must be entered in the clockwise direction.

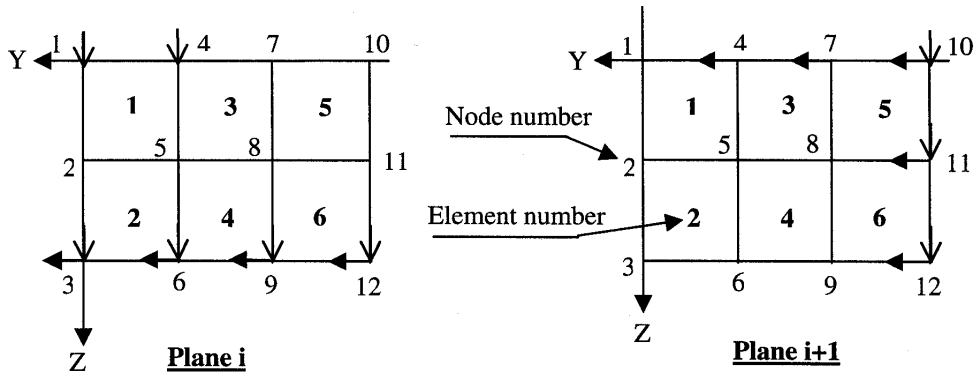


An illustration of distributed edge loads on one edge is shown in the following figure.



If the load is positive its direction should be the same as the coordinate direction, and if the load is negative its direction is opposite to the coordinate direction.

Example:



Input formats corresponding to the above figures are shown below:

```
*DLOADBC
  *PLANE i
    *ELEMENT 1
      ** SN1:EN2   NL1   TL1   NL2   TL2
           4:1     10     0     5     0
    *ELEMENT 2
      ** SN1:EN2   NL1   TL1   NL2   TL2
           3:6      1     2     5     5
    *ELEMENT 4
      ** SN1:EN2   NL1   TL1   NL2   TL2
           6:9     10     1    10     1
    *ELEMENT 6
      ** SN1:EN2   NL1   TL1   NL2   TL2
           9:12     2     1     2     1
  *PLANE i+1
    *ELEMENT 3
      ** SN1:EN2   NL1   TL1   NL2   TL2
           7:4      0     1     0     1
    *ELEMENT 5
      ** SN1:EN2   NL1   TL1   NL2   TL2
          10:7      4     4     3     3
          11:10     7     8     1     2
    *ELEMENT 6
      ** SN1:EN2   NL1   TL1   NL2   TL2
          12:11     4     5     3     1
```

Note the zero values in some column corresponding to no force. In plane i+1, there is one null element; thus, the total number of elements in that plane is only 5 instead of 6 as shown in plane i. Since the strains are computed in plane **normal to the I direction**, the starting node and ending node are entered in the counter clockwise direction as shown in the example.

Second approach: Based on the local nodal points which are defined in the section of keyword *PRESCBC.

In this approach the direction of local nodes is not as important as it is for the first approach. The value (sign) of normal load determines whether the edge is compressive (positive) or tensile (negative). In addition, users need to know only the grid block location which is based on (I,J,K), the local nodes of a finite element and the values of loads on assigned nodes.

Using this approach, the above example of distributed loads can be expressed as:

```
*DLOADBC
** on plane i
*IJK    i     1    1
   ** local node1:local node2     NL1      TL1       NL2      TL2
                4:1                10        0         5        0
*IJK    i    1    2
   ** local node1:local node2     NL1      TL1       NL2      TL2
                2:3                 1        2         5        5

*IJK    i    2    2
   ** local node1:local node2     NL1      TL1       NL2      TL2
                2:3                10        1         10       1
*IJK    i    3    2
   ** local node1:local node2     NL1      TL1       NL2      TL2
                3:2                 2        1          2       1

** on plane i+1
*IJK    i    2    1
   ** local node1:local node2     NL1      TL1       NL2      TL2
                4:1                 0        1          0       1
*IJK    i    3    1
   ** local node1:local node2     NL1      TL1       NL2      TL2
                4:1                 4        4          3       2
                3:4                 7        8          1       2
*IJK    i    3    2
   ** local node1:local node2     NL1      TL1       NL2      TL2
                3:4                 4        5          3       1
```

Distributed Surface Loads (3D) *DLOADBC3D, *DLOADIJK

PURPOSE:

Specify distributed load on a surface of a 3D finite element.

FORMAT:

```
*DLOADBC3D
   {*IJK      i1:i2   j1:j2   k1:k2
         node1   node2   node3   node4   load }
*DLOADIJK
   {*IJK      i1:i2   j1:j2   k1:k2
         *Face
         node   loadx   loady   loadz }
```

DEFINTIONS:

*DLOADBC3D
: Specify distributed normal loads on a surface of a 3D finite element.

*IJK
: Keyword for reservoir grid blocks.

i1
: Starting block index in I direction.

i2
: Ending block index in I direction.

j1
: Starting block index in J direction.

j2
: Ending block index in J direction.

k1
: Starting block index in K direction.

k2
: Ending block index in K direction.

node1
: First nodal point on a surface.

node2
: Second nodal point on a surface.

node3
: Third nodal point on a surface.

node4
: Fourth nodal point on a surface.

load

Uniform load (kN/m^2 | tonf/m^2); (tonf/m^2 = 13.88 psi).

*DLOADIJK

Specify distributed loads on three directions at each node of an element.

**Face*

Face of a local element. It may be *LEFT, *RIGHT, *FRONT, *BACK, *TOP, *BOTTOM.

node

Node number of a local 3D element.

loadx

Distributed load in the I direction (kPa | psi).

loady

Distributed load in the J direction (kPa | psi).

loadz

Distributed load in the K direction (kPa | psi).

CONDITIONS:

When using *DLOADBC3D, four nodal points on a surface of a 3D finite element must be entered.

When using *DLOADIJK, face of the element must be defined before the nodes are assigned.

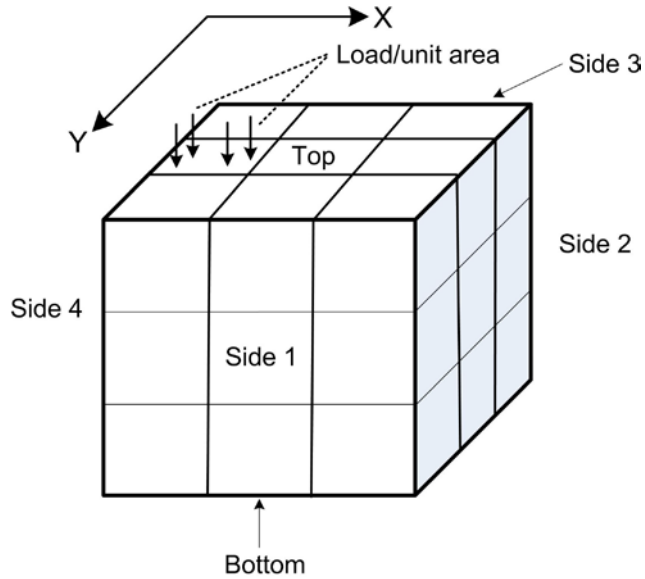
The local nodes must lie on the surface of the element.

EXPLANATION:

Distributed Normal Loads

Keyword *DLOADBC3D allows the user to distribute loads evenly on a surface of a 3D finite element. As mentioned in the keyword *GEOM3D, a 3D finite element consists of 8 nodal points and each face of it is composed by 4 nodal points. Therefore, to indicate which surface of a finite element is subject to distributed loads, a set of four nodal points must be entered but is not necessarily in order. A uniform load is given after the nodal points. If the load value is positive, the surface of a finite element is compressed. If the load value is negative, the surface is stretched. Again, *KDIR *DOWN and *KDIR *UP must be taken into account.

Example: Consider the reservoir shown below. A uniform distributed load of 1kN/m2 is applied on one block of the reservoir. Since the load is positive, the arrows are pointing on the surface as illustrated.



For *KDIR *DOWN:

Since the load is distributed on surface of the block I =1, J = 2 and K = 1, the data entry for such loads is given as (unit load in tonf/m^2):

```
*DLOADBC3D
*IJK          1    2    1
          ** node1    node2     node3   node4     load
               1        2         3       4        1.0
```

For *KDIR *UP:

In this case, the load is distributed on surface of the block I =1, J = 2 and K = 3, the data entry for such loads is given as (unit load in tonf/m^2):

```
*DLOADBC3D
*IJK          1    2    3
          ** node1    node2     node3   node4     load
               5        6         7       8       1.0
```

Distributed Arbitrary Loads

Keyword *DLOADIJK can handle arbitrarily distributed loads on a face of a finite element. When using this keyword, the following procedures must be followed:

1. Enter keyword *DLOADIJK in the geomechanical section.
2. Enter keyword *IJK to determine the region on which distributed loads are applied.
3. Enter a face of the element on which the load is acting.
4. Enter node numbers with loads in directions. The load is positive when it points to the same direction of the coordinate.

Using the above figure, facing names are denoted as:

LEFT is equivalent to side 4
RIGHT is equivalent to side 2
FRONT is equivalent to side 1
BACK is equivalent to side 3
TOP and **BOTTOM** are defined in the figure

Local node numbers can be seen in the section of keyword *PRESCBC3D.

Example 1:

Using the above example of DLOADBC3D, the equivalent data for this keyword are given as follows:

For *KDIR *DOWN:

```
*DLOADIJK
*IJK  1 2 1
*TOP  **loadx  loady  loadz
   1      0      0    13.88     ** Unit load in psi
   2      0      0    13.88     ** Unit load in psi
   3      0      0    13.88     ** Unit load in psi
   4      0      0    13.88     ** Unit load in psi
```

The positive load in this case indicates that the surface is compressed.

For *KDIR *UP:

```
*DLOADIJK
*IJK  1 2 1
*TOP  **loadx  loady  loadz
   1      0      0    -13.88     ** Unit load in psi
   2      0      0    -13.88     ** Unit load in psi
   3      0      0    -13.88     ** Unit load in psi
   4      0      0    -13.88     ** Unit load in psi
```

The negative load in this case indicates that the surface is compressed.

Example 2:

When the shearing loads along the K direction are applied to side 2 and side 4, the data entry is given as:

For KDIR DOWN:

```
*DLOADIJK
*IJK  1 1:3 1:3
*LEFT  ** equivalent to side 4 of the figure
   1      0      0    -13.88  ** Stretching this face
   4      0      0    -13.88  ** along K direction
   5      0      0    -13.88
   8      0      0    -13.88
*IJK  3 1:3 1:3
*RIGHT  ** equivalent to side 2 of the figure
   1      0      0    13.88  ** Compressing this face
   4      0      0    13.88  ** along K direction
   5      0      0    13.88
   8      0      0    13.88
```

Example 3:

When the normal distributed loads along the I direction are applied to side 4, the data entry is given as:

For KDIR DOWN:

```
*DLOADIJK
*IJK  1 1:3 1:3
*LEFT  ** equivalent to side 4 of the figure
   1      13.88      0      0  ** Compressing this face
   4      13.88      0      0  ** along I direction
   5      13.88      0      0
   8      13.88      0      0
```

Example 4:

Another feature of the keyword *DLOADIJK is that the number of nodes may vary from 1 to 4. For instance, if only three nodes at the corner of the top surface of the element (1,1,1) in the figure with KDIR DOWN are loaded, the data entered in the geomechanical section is:

```
*DLOADIJK
*IJK  1 1 1
*TOP
   1      0      0      13.88
   2      0      0      13.88
   4      0      0      13.88
```

Here there are only three nodes on a TOP face subject to distributed loads. In this case, the loads are distributed on the area defined by nodes 1, 2 and 4.

It should be noted that the above distributed loads are different from these distributed loads.

```
*DLOADIJK
*IJK  1 1 1
*TOP
   1      0      0      13.88
   2      0      0      13.88
   4      0      0      13.88
   3      0      0        0.0
```

In this case the loads are distributed on the area defined by 4 nodes 1, 2, 3 and 4.

Gravity Loads (2D) *GLOADBC, *SPECGRAV

PURPOSE:

Specify direction and magnitude of body forces on a per-plane basis.

FORMAT:

*GLOADBC
{ *p1*(:*p2*) *theta* }
*SPECGRAV *spec_grav*

DEFINITIONS:

*GLOADBC
Specify gravity load direction on a per-plane basis.

p1(:*p2*)
Single plane number, or range of plane numbers. See the description for *PRESCBC.

theta
Angle in degrees between the gravity direction (down) and K direction.

spec_grav
Specify gravity (density divided by water density) of the material in the planes. The body force acting on a plane is computed from *spec_grav*, specific water weight of 9806.65 N/m^3 (62.43 lbf/ft^3) and each grid block volume.

CONDITIONS:

Keywords *GLOADBC and *SPECGRAV must appear together if at all, that is, they must be either both absent or both present.

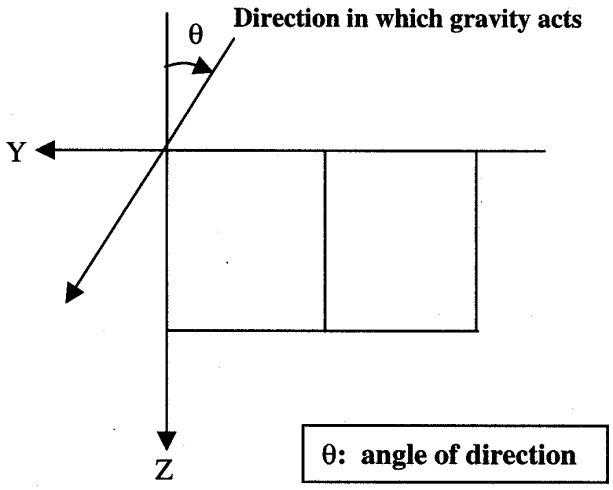
DEFAULTS:

If keyword *GLOADBC is present but does not refer to a particular plane, then the angle is zero for that plane.

There is no default for *SPECGRAV. If *SPECGRAV is absent then *GLOADBC must be absent as well.

EXPLANATION:

Keyword *GLOADBC allows you to assign a body force on any plane. Specific gravity controlled by keyword *SPECGRAV is used to compute the body force per unit volume for each element in the plane. Keyword *GLOADBC is active only when *SPECGRAV is given.



Example:

```
*GLOADBC
  ** plane #        ** theta (degrees)
       1                 10
      2:5                20
*SPECGRAV 1.2  ** Specific weight is 1.2x9806.65 = 11767 N/m3
```

On plane # 1 the K axis is 10 degrees from vertical, whereas on planes 2, 3, 4 and 5 the K axis is 20 degrees from vertical.

Gravity Loads (3D) *GLOADBC3D, *SPECGRAV

PURPOSE:

Specify direction and magnitude of body forces on a per-plane basis in 3D.

FORMAT:

*GLOADBC3D
{ *p1*(:*p2*) *theta1* *theta2* }
*SPECGRAV *spec_grav*

DEFINITIONS:

*GLOADBC3D
Specify gravity load direction on a per-plane basis when 3D finite element is used in computation.

p1(:*p2*)
Single plane number, or range of plane numbers. The plane number for 3D finite element is equivalent to the index I on X axis.

theta1
Angle in degrees between the gravity vector and X direction.

theta2
Angle in degrees between the gravity vector and Y direction.

spec_grav
Specify gravity (density divided by water density) of the material in the planes. The body force acting on a plane is computed from *spec_grav*, specific water weight of 9806.65 N/m^3 (62.43 lbf/ft^3) and each grid block volume.

CONDITIONS:

Keywords *GLOADBC3D and *SPECGRAV must appear together if at all, that is, they must be either both absent or both present.

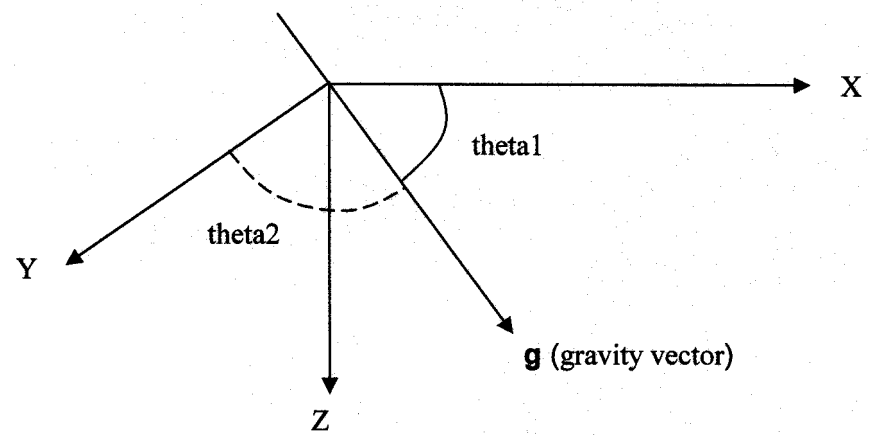
DEFAULTS:

If keyword *GLOADBC3D is present but does not refer to a particular plane, then the angles theta1 and theta2 are 90 degrees.

There is no default for *SPECGRAV. If *SPECGRAV is absent then *GLOADBC must be absent as well.

EXPLANATION:

Keyword *GLOADBC3D allows you to assign a body force in space. Specific gravity controlled by keyword *SPECGRAV is used to compute the body force per unit volume for each element in the plane. Keyword *GLOADBC3D is active only when *SPECGRAV is given.



Note: The angles with respect to X axis and Y axis must satisfy the following condition:

$\{\cos(\text{theta1})\}2 + \{\cos(\text{theta2})\}2 \leq 1$

Example:

```
*GLOADBC3D
  ** plane #       ** theta1 (degrees)   ** theta2
       1                70                    80
      2:5               90                    60
*SPECGRAV 1.2  ** Spec weight is 1.2x62.43 = 74.916 lbf/ft3
```

On plane # 1 (i.e. I = 1) , the gravity vector made an angle 70 degrees with respect to X axis and an angle 80 degrees with respect to Y axis. Whereas, on plane 2:5 (I=2 to I=5), the angles are 90 degrees and 60 degrees with respect to X axis and Y axis.

Fixed Null Block *RIGIDNULL

PURPOSE:

Assign fixed boundary to nodes around a null block.

FORMAT:

*RIGIDNULL *ALL
or
*RIGIDNULL *ELEMENT
{ *plane element* }

DEFINITIONS:

*RIGIDNULL *ALL
Specifies fixed boundary for nodes around all null blocks.

*RIGIDNULL *ELEMENT
Specifies fixed boundary for nodes around null blocks associated with the indicated elements.

plane element
Plane and element number, respectively, of the element corresponding to the null block. See the description of plane and element numbering for keyword *PRESCBC.

CONDITIONS:

If keyword *RIGIDNULL is present, only one of sub-keyword *ALL or *ELEMENT may be used.

DEFAULT:

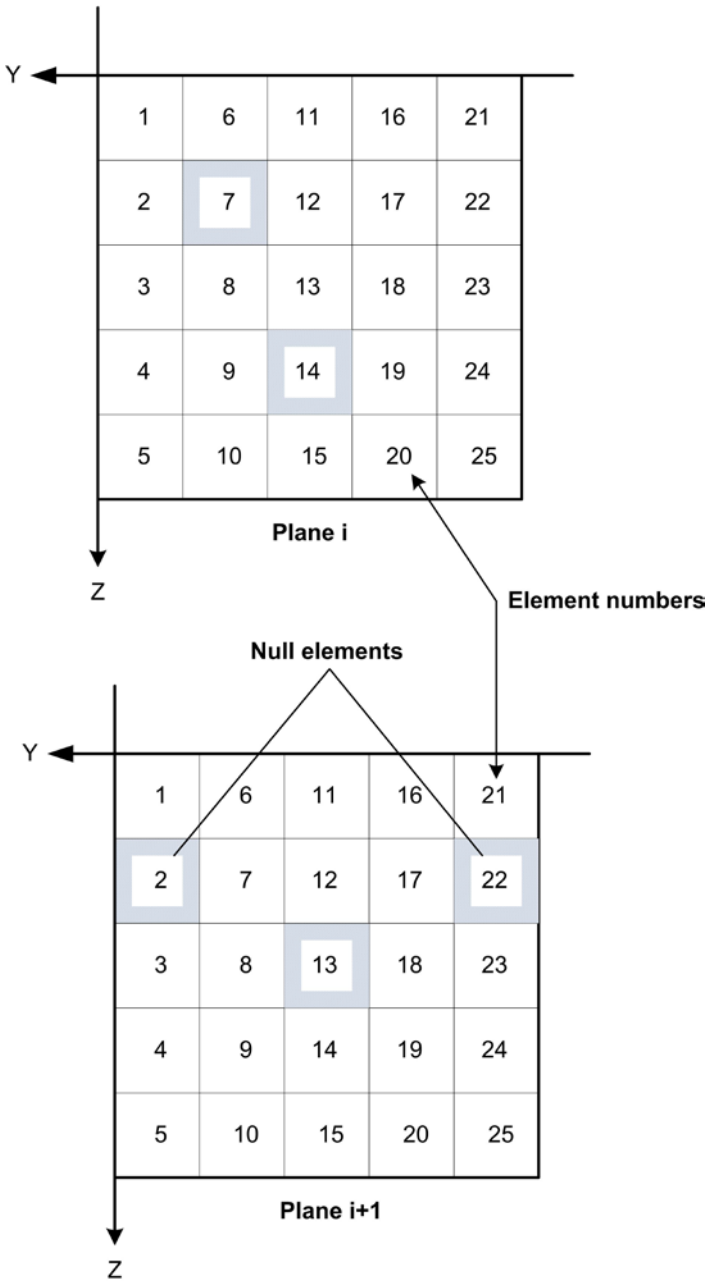
When keyword *RIGIDNULL is absent, movement is allowed for nodes on the edge common to a null block and a non-null block in the plane in which strain is calculated. Note that, by definition, node movement is not allowed in the direction perpendicular to the strain plane.

EXPLANATION:

Keyword *RIGIDNULL allows you to specify null blocks whose surrounding nodes will be completely rigid, that is, will not be allowed to move during calculation of the strain problem. Thus, all nodal points on the attached edges around the specified null block are not moving in any direction.

Example:

In the figures, plane i contains two null blocks while plane i+1 contains 3 null blocks.



If all null blocks are fixed, use the following:

```
*RIGIDNULL *ALL
```

If null blocks only in plane i are fixed, use the following:

```
*RIGIDNULL *ELEMENT
  ** plane_number  element_number
          i              7
          i             14
```

Fixed Cap Rock *RIGIDTOP

PURPOSE:

Constrain vertical movement on the cap rock of a reservoir.

FORMAT:

*RIGIDTOP

DEFINITIONS:

*RIGIDTOP
This single keyword causes the cap rock of a reservoir to be vertically constrained at the top.

DEFAULT:

If keyword *RIGIDTOP is absent, then the cap rock of the reservoir is either not constrained or is constrained by another keyword such as *PRESCBC.

EXPLANATION:

Keyword *RIGIDTOP allows you to assign constrained boundary conditions at the top of the reservoir on the vertical direction. This keyword is an alternative to using keyword *PRESCBC to assign displacement boundary conditions on a reservoir. However, keyword *RIGIDTOP is more convenient because the "essential" boundary conditions are not removed automatically as they are when keyword *PRESCBC is used. Essential boundary conditions are defined as laterally constrained movements on vertical sides of a reservoir and vertically constrained movements at the bottom of a reservoir.

Geomechanics Domain *GEODOMAIN

PURPOSE:

Specify grid sub domains in which geomechanics calculations are performed.

FORMAT:

*GEODOMAIN *ALL
or
*GEODOMAIN *IJK
{ *i1:i2 j1:j2 k1:k2* }

DEFINITIONS:

*GEODOMAIN *ALL
Geomechanics calculations are performed for all active grid blocks.

*GEODOMAIN *IJK
Geomechanics calculations are performed only in selected regions of grid blocks called geomechanics domains.

i1:i2 j1:j2 k1:k2
Range of grid indices in the I, J and K directions. Each index value must fall within 1 to the maximum number of blocks in that direction (see keyword *GRID). Each lower index must not exceed the corresponding upper index. Enter a single index as a range, for example, 5:5.

DEFAULT:

If keyword *GEODOMAIN is absent, then *GEODOMAIN *ALL is assumed.

If keyword *GEODOMAIN *IJK is present, then geomechanical calculations are not done for blocks that are not referenced explicitly.

EXPLANATION:

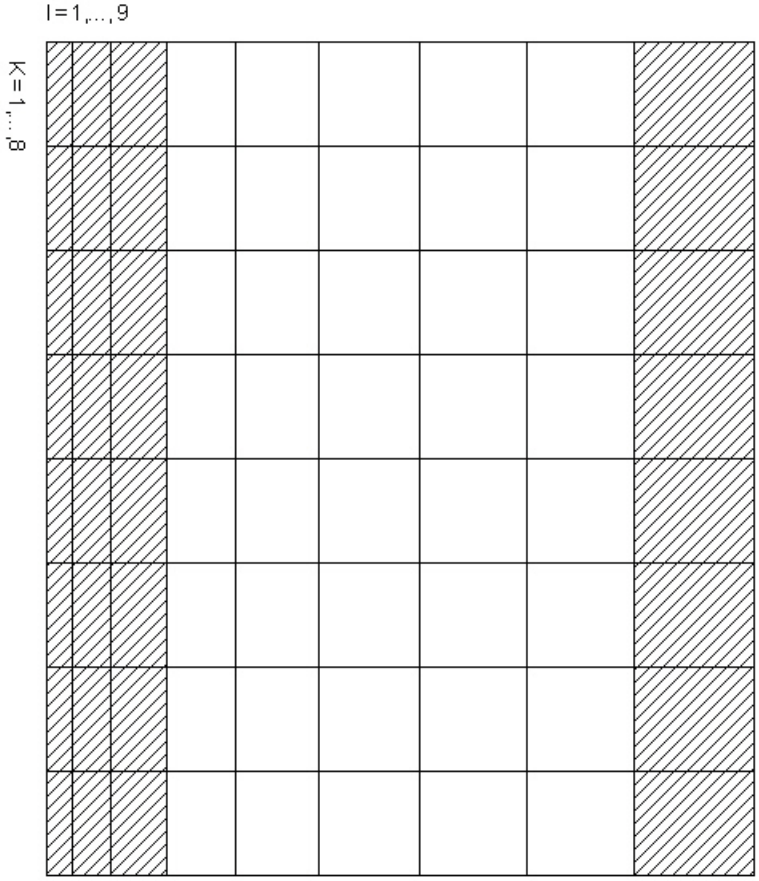
Keyword *GEODOMAIN allows you to restrict coupled geomechanics calculations to specified sub domains of a reservoir, resulting in increased efficiency and saving in CPU time.

Geomechanical response results (stresses, strains, Young modulus, Poisson ratio, etc.) are calculated only for blocks in the geomechanics domains. However, to satisfy textual and graphical output requirements these quantities are given zero values for blocks that are not in a geomechanics domain.

Example: In an axis-symmetry reservoir with NI = 9, NJ = 1, NK = 8, there are two geomechanics domains defined as follows:

```
*GEODOMAIN *IJK
  ** i1:i2   j1:j2   k1:k2
      1:3     1:1     1:8  ** domain 1
      9:9     1:1     1:8  ** domain 2
```

As shown in the figure, the marked grid blocks belong to geomechanics domains and the blank grid blocks are outside the geomechanics domains.



Pressure Boundary Domain *BCDOMAIN

PURPOSE:

Specify pressure boundary between geomechanics and non-geomechanics grid domains.

FORMAT:

*BCDOMAIN *ALL
or
*BCDOMAIN *IJK
{ *i1:i2 j1:j2 k1:k2* }

DEFINITIONS:

*BCDOMAIN *ALL
Apply pressure boundary between all parts of geomechanics and non-geomechanics grid domains.

*BCDOMAIN *IJK
Apply pressure boundary between selected parts of geomechanics and non-geomechanics grid domains.

{ *i1:i2 j1:j2 k1:k2* }
Table of ranges of grid indices in the I, J and K directions. Each index value must fall within 1 to the maximum number of blocks in that direction (see keyword *GRID). Each lower index must not exceed the corresponding upper index. Enter a single index value as a range, for example, 5:5.

DEFAULT:

If keyword *BCDOMAIN is absent then no pressure boundary is applied.

If keyword *BCDOMAIN is present, keyword *GEODOMAIN must appear.

EXPLANATION:

Keyword *GEODOMAIN specifies one or more block regions or domains in which geomechanics calculations are performed. Blocks omitted from *GEODOMAIN make up one or more non-geomechanics domains. The result is one or more surfaces that separate the blocks in geomechanics domains from blocks in non-geomechanics domains.

Consider two adjacent blocks, one in a geomechanics domain and the other in a non-geomechanics domain. By default, the block in the non-geomechanics domain has no pressure influence on the block in the geomechanics domain, as though there is a no-flow boundary between them. This is appropriate in some circumstances, for example, when a rigid unperforated casing isolates a wellbore volume from the surround formation.

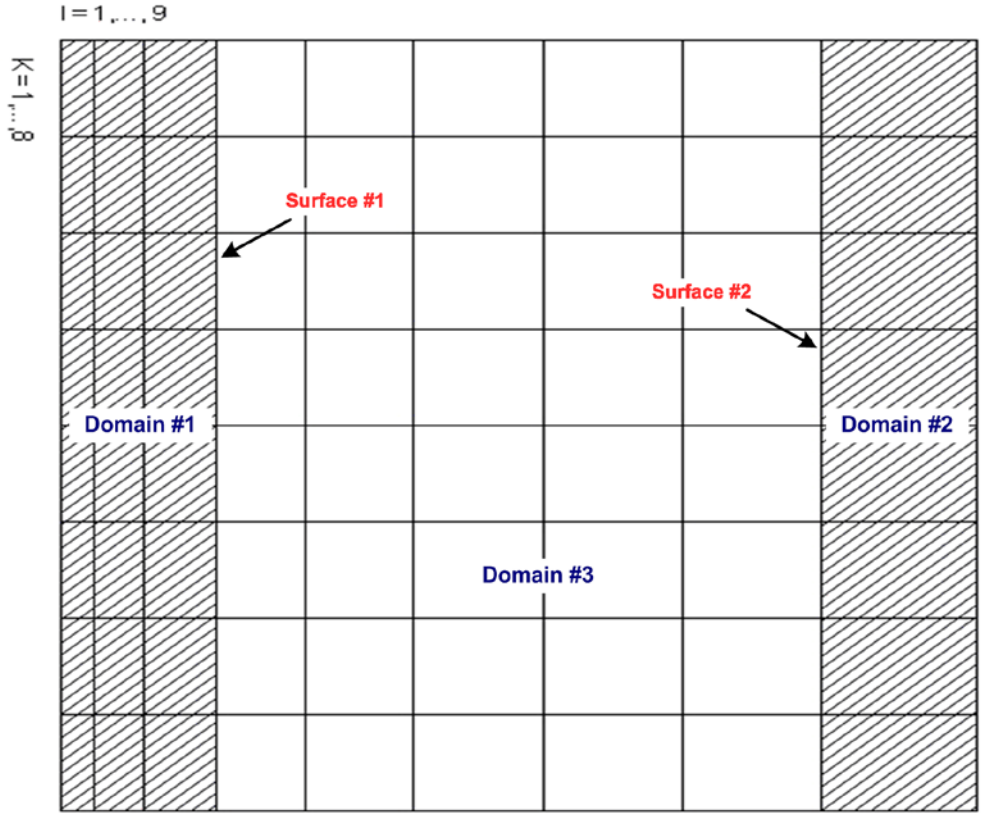
However, in cases where there is no rigid no-flow boundary between the geomechanics and non-geomechanics domains, use keyword *BCDOMAIN to apply a pressure-type boundary between them. This lets the fluid pressure of a block in the non-geomechanics domain influence its neighbouring block in the geomechanics domain as a stress constraint. This is

appropriate when modelling deforming regions together with open or void volumes, such as non-cased wellbores and open caverns.

Due to the CPU characteristics of the algorithm used to search for boundary locations, the number of cells referenced in keyword *BCDOMAIN should be as few as possible.

Example: Continuation of *GEODOMAIN

This 9x1x8 grid consists of geomechanics domains "Domain #1" in cells (1:3,1,1:8) and "Domain #2" in cells (9,1,1:8), along with non-geomechanics domain "Domain #3" in cells (4:8,1,1:8). Use keyword *GEODOMAIN to specify this information. In addition, Domain #3 is an open volume containing fluid but no rock (100% porosity).



Without *BCDOMAIN, Domains #1 and #2 are not influenced by the fluid pressure in Domain #3, since rigid "no-flow" boundary constraints are applied to the Surfaces #1 and #2. If all the (fluid) driving forces are in Domain #3, then there will be no geomechanics responses in Domains #1 and #2.

However, we do want the pressure of the fluid in Domain #3 to act as a stress boundary condition for Domains #1 and #2. Use keyword *BCDOMAIN to specify the location of this boundary condition by referencing cells on both sides of the desired boundary surfaces.

The following data specifies that only "Surface #1" in the above diagram has a pressure boundary condition.

```
*BCDOMAIN *IJK
  ** i1:i2   j1:j2   k1:k2
      3:4     1:1     1:8  ** Surface #1
```

The following data specifies that both "Surface #1" and "Surface #2" have that boundary condition. This refers to as few blocks as possible and so gives the most efficient boundary searching.

```
*BCDOMAIN *IJK
  ** i1:i2   j1:j2   k1:k2
      3:4     1:1     1:8  ** Surface #1
      8:9     1:1     1:8  ** Surface #2
```

The following data specifies both "Surface #1" and "Surface #2" as well but the boundary searching will be more expensive.

```
*BCDOMAIN *ALL
```

Coupling Options

***GCOUPLING, *NOCOUPERM**

PURPOSE:

Control porosity and permeability coupling from deformation to fluid flow.

FORMAT:

*GCOUPLING (0 | 1 | 2 | 3)

*NOCOUPERM

DEFINITIONS:

*GCOUPLING 0
: Fluid-flow porosity does not depend upon deformation.

*GCOUPLING 1
: Fluid-flow porosity is a function of pressure, temperature and volumetric strain.

*GCOUPLING 2 & 3
: Fluid-flow porosity is a function of pressure, temperature and total mean stress.

*NOCOUPERM
: Fluid-flow permeability does not depend upon deformation, despite the presence of deformation-dependent permeability keyword data.

DEFAULT:

If keyword *GCOUPLING is absent, then *GCOUPLING 2 is assumed.

If *NOCOUPERM is absent, all deformation-dependent permeability data is honoured.

EXPLANATION:

Background

Fluid flow and formation deformation (geomechanics) are coupled together in a sequential manner, that is, the two calculations alternate while passing information back and forth. The fluid flow calculation updates the pressures and temperatures over an interval specified by *GCUPDATE. The geomechanics module updates the formation deformation in response to the new pressures and temperatures. To complete the loop, the geomechanics module sends the new deformation information back to the fluid flow calculation for use in the next time interval. It is clear that information flows from fluid flow to geomechanics via pressure and temperature. However, it is not obvious how information flows back the other way.

The fluid flow module calculates porosity as a function of pressure and temperature, in a way that pore volume and hence mass is conserved between time steps. Here "conserved" means that the porosity at the beginning of a time step is equal to the porosity at the end of the previous time step, at that particular pressure and temperature. When the porosity function $\varphi(p,T)$ itself does not change with time, mass conservation across time steps is ensured.

However, the porosity function can vary between time steps and still conserved mass. Let p^n and T^n be the solution for a grid block for time step "n" that used porosity function $\varphi^n(p,T)$. The next time step "n+1" starts with $p = p^n$ and $T = T^n$ but has a different porosity function $\varphi^{n+1}(p,T)$. Porosity and hence mass will be conserved between these two time steps if $\varphi^n(p^n,T^n) = \varphi^{n+1}(p^n,T^n)$. However, φ^n and φ^{n+1} may have different derivatives with respect to dependent variables p and T at $p=p^n$ and $T=T^n$.

The geomechanical deformation response is expressed in the fluid flow calculation through changing parameters in the porosity function. These parameters are kept constant during time-step convergence but are updated between time steps such that porosity and hence mass is conserved. The deformation response is accounted for on a block-by-block basis since each grid block has its own set of porosity function parameters.

Keyword *GCOUPLING allows you to select the particular form of porosity function for the coupling of reservoir flow equations and geomechanical calculations. In the model descriptions below, superscript "n" denotes the *nth* fluid-flow update, "n-1" denotes the previous update, and "0" denotes initial conditions.

Porosity Coupling Models

***GCOUPLING 0:**

Fluid flow porosity contains no parameters that depend upon deformation from geomechanics. This is "one way" coupling since fluid pressures and temperatures are still used by the geomechanics module to update formation strains and stresses. This option is useful when the fluid-flow calculations use a porosity model that cannot be approximated well by the other "two-way" coupling models.

STARS: An example of this is the STARS empirical porosity model *DILATION. See keyword *PGDILA for a discussion on modelling dilation with the geomechanics module. If *PGDILA is used then Young's modulus is not used.

Young's modulus is approximated from rock compressibility as:

$$E = \frac{(1-2\nu)(1+\nu)}{\phi_0 C_R (1-\nu)}$$

where:

E	:	Young's modulus (kPa | psi)
ν	:	Poisson's ratio
ϕ_0	:	Initial porosity
C_R	:	Rock compressibility

***GCOUPLING 1**

Porosity is a function of pressure, temperature and volumetric strain and has the form:

$$\phi^{n+1} = \phi^0 \left[\frac{V_p^{n-1}}{V_p^0} + \frac{\left(V_p^n - V_p^{n-1}\right)}{V_p^0\left(p^n - p^{n-1}\right)}\left(p - p^n\right) \right] - \beta\left(T - T_0\right)$$

where:

p	:	Pressure (kPa | psi)
T	:	Temperature (C | F | F)
V_p	:	Pore volume (m^3 | ft^3)
β	:	Volumetric thermal expansion coefficient of the formation (1/C | 1/F | 1/C) given by *CTPOR
ϕ	:	Porosity

***GCOUPLING 2**

Porosity is a function of pressure, temperature and total mean stress formula and has the form (Tran et al, SPE/ISRM 78192):

$$\phi^{n+1} = \phi^n + (c_0 + c_2 a_1)(p - p^n) + (c_1 + c_2 a_2)(T - T^n)$$

where:

$$c_0 = \frac{1}{V_b^0}\left(\frac{dV_p}{dp} + V_b \alpha c_b \frac{d\sigma_m}{dp} - V_p \beta \frac{dT}{dp}\right)$$

$$c_1 = \frac{V_p}{V_b^0}\beta$$

$$c_2 = -\frac{V_b}{V_b^0}\alpha c_b$$

$$a_1 = \text{factor}\left\{\frac{2}{9}\frac{E}{(1-\nu)}\alpha c_b\right\}$$

$$a_2 = \text{factor}\left\{\frac{2}{9}\frac{E}{(1-\nu)}\beta\right\}$$

c_b	:	Bulk compressibility (1/kPa | 1/psi)
E	:	Young's modulus (kPa | psi) factor given by*GCFACTOR
V_b	:	Bulk volume (m3 | f3)
α	:	Biot number
ν	:	Poisson's ratio
σ_m	:	Mean total stress (kPa | psi)

This porosity model is expected to give similar results to *GCOUPLING 1 when the stress response is small but may give different results when the stress response is significant.

***GCOUPLING 3**

Porosity is a function of pressure, temperature and total mean stress formula and has the form

$$\phi^{n+1} = \phi^n + c_0 \Delta p + c_1 \Delta T + c_2 \Delta\sigma_m$$

where c_0, c_1 and c_2 are the same as for *GCOUPLING 2.

Permeability Coupling

Several options allow you to vary the fluid-flow permeability as a function of deformation.

Matrix: *GPERMLC, *GPERMES, *GPERMTS, *GPERMVL

Fracture: *GPERMBB

Keyword *NOCOUPERM allows you to disable permeability coupling globally without the need to remove all the deformation-dependent permeability keyword data. This is useful when evaluating the effect of permeability variations when such data has been specified for a number of rock types. To disable permeability coupling only for selected rock types, the corresponding data must be removed for those rock types.

Examples:

```
*GEOMECH
*GCOUPLING 2  ** Porosity depends on p, T, total stress
*NOCOUPERM    ** Disable all variable-permeability data
```

References:

Tran, D., Settari, A. and Nghiem, L.: *"New Iterative Coupling between a Reservoir Simulator and a Geomechanics Module"*, SPE/ISRM 78192, 2002.

Geomechanical Coupling Factor *GCFACTOR

PURPOSE:

This optional multiplier factor is only used along with the second coupling formula in the geomechanical module, i.e. *GCOUPLING 2.

FORMAT:

*GCFACTOR *factor*

DEFAULT:

If keyword *GCFACTOR is absent, the value of multiplier factor is one.

DEFINITION:

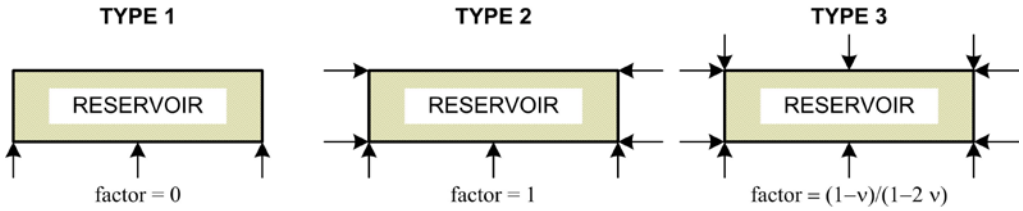
factor
A positive real value which has a range from 0 to 10.

CONDITION:

If the keyword *GCFACTOR appears, a value of factor must be given.

EXPLANATION:

When the second formula of coupling (*GCOUPLING 2) is used, the optional keyword *GCFACTOR is designed along with its value factor to handle different types of constrained boundary conditions which are assigned on a reservoir. For instance, when the value of factor is zero (by default), the reservoir is constrained only at its bottom and freely moves in other directions (Type 1). If the reservoir is constrained laterally and is free to move vertically (Type 2), the factor has a value of one. If the reservoir is constrained in all directions (Type 3), the value of factor equals to (1- ν)/(1-2 ν); where ν is the Poisson's ratio of porous rock. For other types of constrained boundary conditions, the value of factor may be approximated by a real number between 0 and (1- ν)/(1-2 ν). The above three types of boundary constraints are illustrated in 2D forms as follows:



Pressure Tolerance Multiplier *GPTOLMUL

PURPOSE:

Adjust pressure tolerance for computing porosity formulae given in keyword *GCOUPLING.

FORMAT

*GPTOLMUL *multiplier*

DEFINITIONS:

*multiplier
A positive real number

DEFAULT:

If keyword *GPTOLMUL is absent, then the value of multiplier is one.

EXPLANATION:

By default, the pressure tolerance used in computing coefficients for porosity formulae in the section of coupling options is 1 psi or 6.89475 kPa. This tolerance can be adjusted by an amount such as: tolerance = tolerance * multiplier; where multiplier is a positive real number ranging from 1E-6 to 1E+6. As seen in porosity formulae and the coefficients in the coupling options, because there is existence of the term 1/dp, the pressure tolerance must be used to avoid the case of infinitive when dp is zero. Depending on the problem, the pressure tolerance should be adjusted to suit with the pressure change between two consecutive time steps. For instance, if the pressure change is always less than 1 psi, value of multiplier should be less than 1.

Example:

```
*GPTOLMUL   0.5  ** pressure tolerance is 0.5 psi
```

Coupling Update Times *GCUPDATE

PURPOSE:

Specify at what frequency or times that coupling updates are done.

FORMAT:

*GCUPDATE (*freq* | *TIME | *TNEXT)

DEFINITIONS:

*GCUPDATE

Specify frequency or time of updating geomechanical conditions and the resulting porosities as specified by keyword *GCOUPLING. *GCUPDATE may appear more than once in a data set, perhaps with different sub-options. The action indicated by the default or specified sub-option stays in effect until the new appearance of this keyword.

freq

Non-negative integer *freq* indicates that an update is done for each time-step number evenly divisible by *freq*, in addition to each recurrent data time. Setting *freq* = 0 disables updating. This option may result in geomechanics updating that is unnecessarily frequent when time-step sizes are small.

*TIME

Updating is done for each recurrent data time. This is equivalent to setting a large value for *freq*. This option may result in geomechanics updating that is unnecessarily frequent, especially for recurrent data containing many closely-spaced times.

*TNEXT

Updating is done only for the next recurrent data time after which updating is disabled, resulting in one update per keyword occurrence. This option is useful when specifying updates at infrequent but know times in recurrent data.

DEFAULTS:

This keyword is initialized to *freq* = 1.

CONDITIONS:

This keyword is valid also in the Well and Recurrent Data section.

EXPLANATION:

For larger runs the time spent in performing geomechanics calculations can be a significant fraction of the total simulation run time. If geomechanical changes are small or slow compared to fluid flow changes, geomechanical coupling can be done less frequently thereby resulting in shorter total run times. Frequency sub-options *TIME and *TNEXT allow you to tailor a large run for optimal run times with adequate geomechanical representation.

Keyword *GCUPDATE affects the frequency at which the deformed geomechanics grid is dumped to the SR2. At the dump times specified by *WSRF *GRIDDEFORM in the Input/Output Control data section, the grid is dumped only if it has been changed via geomechanics updating after the last dump time. This prevents dumping of the "same" grid more than once to the SR2. For example, suppose that you wish to dump the geomechanics grid every time that the update is done as specified by *GCUPDATE. Do this with *WSRF *GRIDDEFORM 1, since the grid will not be dumped until it changes. This is much easier to specify in data than manually synchronizing *WSRF *GRIDDEFORM with *GCUPDATE.

Keyword *GCUPDATE affects the frequency of dumping geomechanics quantities specified by *OUTSRF *GRID to the SR2. At dump times specified by *WSRF *GRID, these geomechanics quantities are dumped only if the geomechanical response has been updated since the last dump time. This prevents dumping a geomechanics solution that is out-of-date. The dumping of non-geomechanics quantities does not depend upon *GCUPDATE.

Keyword *GCUPDATE affects the frequency at which geomechanics quantities are updated for special histories. But special histories are assigned every time step and interpolation is not done between update times. Therefore, less frequent geomechanics updates may result in a corresponding special history plot that exhibits artificial "stair-step" behavior. In this case it may be more appropriate to use the RESULTS Graph source option "Add Block Property Versus Time" which will not attempt to plot values between the actual dump times.

Example:

```
*GCUPDATE 10     ** Update every 10 time steps.
*GCUPDATE *TIME  ** Update each *TIME/*DATE.
*GCUPDATE *TNEXT ** Update only next *TIME/*DATE.
```

Example: Update at known but infrequent times

```
*TIME 100
*TIME 110
...  many well changes from history match
*TIME 245
  *GCUPDATE *TNEXT
*TIME 250  ** Geomechanics update time
*TIME 260
...
```

Porosity Calibration *CALIB_POR

PURPOSE:

Improve accuracy of porosity coupling option *GCOUPLING 2.

FORMAT:

*CALIB_POR

DEFAULTS:

If the keyword does not appear, there is no porosity calibration.

CONDITIONS:

This keyword applies to all geomechanics rock types.

This keyword is used only with *GCOUPLING 2.

EXPLANATION:

The purpose of this option is to improve the accuracy of porosity computed from a second order formula of *GCOUPLING 2 given in the EXPLANATION for *GCOUPLING. Since the current reservoir porosity is computed on the basis of porosity coefficients which were estimated in the geomechanics module in a previous time step, the reservoir porosity may not be as accurate as desired. Because of the explicit calculation of those coefficients and the approximate nature of the porosity function, there is a difference between the porosity calculated in the host simulator and the actual porosity calculated by GEOMECH. The difference could become substantial, especially for plastic deformation and shear dilation.

After the reservoir porosity is determined, it is compared to the actual porosity which is computed in the geomechanics module. The actual porosity at time step *n* is defined here as a ratio between a pore volume at time step *n* and the initial bulk volume of a grid block. The difference between the reservoir porosity and actual porosity at the end of one time step will be used to calibrate the next time-step porosity through a weighting function. The main goal of this option is to bring reservoir porosity close to the corrected porosity gradually at the later time steps.

The porosity formula used in *GCOUPLING 2 in the Coupling Options Section can be written in an extended form as:

$$\phi_{n+1} = \phi_n + (c_0 + c_2 a_1)_n (p - p_n) + (c_1 + c_2 a_2)_n (T - T_n) + \omega^{(k)} \left(\phi_n^c - \phi_n\right)$$

where:

$\phi_n^c = \dfrac{V_n^p}{V_0^b}$: Actual porosity based on the definition

V_n^p : Pore volume at time step n

V_0^b : Initial bulk volume

$\omega^{(k)}$: Weighting function at k^{th} iteration is expressed as:

$$\omega^{(k)} = \alpha_p f\left(\varepsilon_p^k\right) + \alpha_T f\left(\varepsilon_T^k\right)$$

where:

$$f\left(\varepsilon_p^k\right) = 1 - \exp\left(-\frac{1}{1 - \frac{\left|p_{n+1}^{(k)} - p_{n+1}^{(k-1)}\right|}{p_n}}\right)$$

$$f\left(\varepsilon_T^k\right) = 1 - \exp\left(-\frac{1}{1 - \frac{\left|T_{n+1}^{(k)} - T_{n+1}^{(k-1)}\right|}{T_n}}\right)$$

$$\alpha_p = \frac{\left(c_0 + c_2 a_1\right)_n \left(p_n - p_{n-1}\right)}{\left(c_0 + c_2 a_1\right)_n \left(p_n - p_{n-1}\right) + \left(c_1 + c_2 a_1\right)_n \left(T_n - T_{n-1}\right)}$$

$$\alpha_T = \frac{\left(c_1 + c_2 a_1\right)_n \left(T_n - T_{n-1}\right)}{\left(c_0 + c_2 a_1\right)_n \left(p_n - p_{n-1}\right) + \left(c_1 + c_2 a_1\right)_n \left(T_n - T_{n-1}\right)}$$

Example:

```
*GCOUPLING 2
*CALIB_POR
```

Iterative Coupling to Fluid Flow *NCOUPLING, *PRESSTOL, *STRESSTOL, *POROSTOL

PURPOSE:

Specify parameters that control iterative coupling to the fluid-flow solution.

FORMAT:

*NCOUPLING	*ngeo_iter*
*PRESSTOL	*pres_tol*
*STRESSTOL	*stress_tol*
*POROSTOL	*poros_tol*

DEFINITIONS:

*NCOUPLING *ngeo_iter*

Maximum number of geomechanics updates performed per fluid-flow time step, where *ngeo_iter* is an integer greater than or equal to 1. If the multi-update option is used then normally *ngeo_iter* is given a large number (e.g., 20) so that an adaptive convergence criterion like *PRESSTOL can operate freely.

*PRESSTOL *pres_tol*

Maximum allowed pressure difference allowed for each grid block, between consecutive Newton cycles, before the couple geomechanics-fluid-flow solution is deemed converged. The unit of *pres_tol* is (kPa | psi) and the allowed range is 0 to 10 kPa (0 to 1.45 psi).

*STRESSTOL *stress_tol*

Maximum allowed stress difference allowed for each grid block, between consecutive Newton cycles, before the couple geomechanics-fluid-flow solution is deemed converged. The unit of *stress_tol* is (kPa | psi) and the allowed range is 0 to 10 kPa (0 to 1.45 psi).

*POROSTOL *poros_tol*

Maximum allowed porosity difference allowed for each grid block, between consecutive Newton cycles, before the couple geomechanics-fluid-flow solution is deemed converged. The quantity *poros_tol* is dimensionless and the allowed range is 0 to 1.0.

DEFAULTS:

If keyword *NCOUPLING is absent then *ngeo_iter* = 1 is assumed.

If *ngeo_iter* > 1 and keywords *PRESSTOL, *STRESSTOL and *POROSTOL are absent, the criterion is pressure difference with *pres_tol* = 0.01 kPa (0.00145 psi).

CONDITIONS:

At most one of *PRESSTOL, *STRESSTOL and *POROSTOL is used. If more than one of these keywords is appears, the last one is used.

EXPLANATION:

In most cases only one geomechanics update is performed per time step, which usually is sufficient for convergence and acceptable results. However, some geomechanics problems such as arching or Mandel's effects may require more than one geomechanics update to obtain a satisfactory result. In a time step, each geomechanics update occurs before a fluid-flow Newton cycle. The number of such updates depends upon the parameters specified by these keywords as well as the number of fluid-flow Newton iterations (*NEWTONCYC).

Boundary Stress Unloading *UNLOADSTR

PURPOSE:

Assign total stress to be unloaded at the well boundary.

FORMAT:

```
*UNLOADSTR
        kl          (:k2)       stress
         :            :            :
```

DEFINITIONS:

*UNLOADSTR
This keyword turns on the single-well boundary stress unloading option.

k1(:k2)
Range of grid layers where boundary stress unloading applies.

stress
Radial boundary stress to be unloaded (kPa | psi).

DEFAULTS:

Required keyword for the single-well boundary stress unloading option. There are no defaults.

CONDITIONS:

Boundary stress unloading option works only with a radial grid. Cartesian grid is not allowed.

EXPLANATION:

This keyword allows the user to specify the amount of external boundary stress to be unloaded at the wellbore boundary. The actual radius where the unloading occurs is specified by the *WRADIUS keyword. This unloading may occur as a result of fluid and sand production close to the wellbore. A large reduction in external boundary support stress may lead to tension failure for the elements adjacent to the well.

A scheme for removing elements which is shown to fail in tension has NOT been implemented. Therefore, it is advisable to run the model with some cohesion in this case to avoid stability problems. As well, consideration of sand flow with petroleum fluid is NOT considered in this model.

STARS: Geomechanics Post Processing *GEOPOSTPRO, *GEOPOSTFREQ

PURPOSE:

Run Geomechanics in STARS in post-processing mode.

FORMAT:

*GEOPOSTPRO (*start stop*)
*GEOPOSTFREQ *freq*

DEFINITIONS:

*GEOPOSTPRO (*start stop*)

Generate geomechanics results for each pressure and temperature read from an input SR2 file set or restart chain. This allows you to obtain geomechanics results in "post-processing mode", that is, from the existing output of a run that did not use geomechanics. Fluid-flow time steps are not redone, but full geomechanics calculations are done and reported.

The post-processing run must have the same data as the original run, with geomechanics data added. This way of using geomechanics necessarily implies one-way coupling (*GCOUPLING 0). Temperature is not read for isothermal runs.

You can choose to post-process only part of the original run by specifying optional starting time-step number *start* and ending time-step number *stop*, both time-step numbers of the original run. They must be positive integers within the range of the time-step numbers of the original run, and *start* must not exceed *stop*.

*GEOPOSTFREQ *freq*

Positive integer *freq* indicates that geomechanics updates are done for every *freq* set of p and T values read. This lets you specify fewer geomechanics updates than the existing p and T output. The use of *freq* = 1 results in a geomechanics update for each p and T set in the original run.

Note that *freq* does not necessarily correspond to time-step numbers. For example, if *freq* = 2 and p and T are dumped every 5 time steps of the original run, then a geomechanics update would be done every 10 time steps of the original run.

DEFAULTS:

If *GEOPOSTPRO is absent, no Geomechanics post-processing is done.

If *GEOPOSTPRO is present but *start stop* are absent, post-processing is done for the entire parent run.

If *GEOPOSTFREQ is absent, *freq* = 1 is assumed.

CONDITIONS:

An input SR2 file set must be specified via command-line argument "-r" (like a restart).

This option is not allowed for a restart run, that is, a run using keywords *RESTART, *RESTIME, *RESDATE or their corresponding command-line arguments.

The only *GCOUPLING option allowed is "0" (one-way coupling). If another coupling option is specified via *GCOUPLING or its default (which is not "0"), a warning message is issued and the run proceeds with one-way coupling.

EXPLANATION:

This option allows you to generate geomechanics (stress-strain) estimates for existing fluid-flow simulation results. For example, you could match observed fluid-flow performance with multiple runs without expensive geomechanics calculations, and then from the final fluid-flow results you could perform multiple geomechanics estimates with different geomechanics data. It would be wise to do at least one calibration run with fluid-flow and geomechanics together in two-way coupling mode, to evaluate the validity of the one-way coupling assumption built into *GEOPOSTPRO.

Geomechanics post processing works for both *GEOGRID and the host-grid options. It works also for all the geomechanics subfeatures that depend only on p and T from the fluid-flow host simulation. Graphical and text results are fully functional but only p, T and geomechanics quantities are available.

Note that some advanced geomechanics features, e.g., dependence on solid component composition, are not fully installed and may give spurious results.

Geomechanics post processing involves two simulation runs: (1) an existing complete parent run, and (2) the post-processing run. The parent run usually has no *GEOMECH data. The post-processing run data usually is a copy of the parent run data, but with *GEOMECH data added. The post-processing run reads the parent run's SR2 files specified via command-line argument "-r". This command-line argument is used also to specify the input file set for a restart run, but a post-processing run itself is not a restart run.

If the parent simulation is a set of runs linked by restarting (a restart chain), then the post-processing run should specify the SR2 file set at the end of that chain. In this case the post-processing run scans over all the SR2 file sets in the parent restart chain, producing geomechanics updates in the correct time order.

Like any other run, the names of the output files of the post-processing run are based on the name of the input data file or the "-o" command-line argument. Therefore, it may be most convenient to put each geomechanics data variation into its own main data file so that the run outputs are separated. Since the non-geomechanics data in these runs should be the same, you may make use of the *INCLUDE facility to refer to a single copy of the non-geomechanics data.

Example:

A parent run has data file "sample.dat" has no *GEOMECH data. Two geomechanics post-processing runs are performed, from data files "sample_geo1.dat" and "sample_geo2.dat" that contain the original non-geomechanic data from "sample.dat" plus two variations of geomechanics data. The *GEOMECH data contains the additional line:

```
*GEOPOSTPRO
```

The runs are performed as follows in this command line sketch:

```
stars.exe –f sample.dat –log
copy sample.dat sample_geo1.dat
edit sample_geo1.dat  (add *GEOMECH data, variation #1)
stars.exe –f sample_geo1.dat -log –r sample
copy sample_geo1.dat sample_geo2.dat
edit sample_geo2.dat  (*GEOMECH data, variation #2)
stars.exe –f sample_geo2.dat –log –r sample
```

Restrict post processing to first 200 time steps in "sample" run, with this line in the *GEOMECH data.

```
*GEOPOSTPRO 1 200
```

Restrict post processing to every third set of p,T values in "sample" run, with this line in the *GEOMECH data.

```
*GEOPOSTFREQ 3
```

Restart chain and diary echo are illustrated with template "stgeo006". The parent simulation is in two segments: (1) "stgeo006", and (2) "stgeo006_1" restarted from "stgeo006" at time step #10. The geomechanics post-processing run is data file "stgeo006_1_scan.dat", being a copy of "stgeo006_1.dat" with *GEOMECH data added. The diary file echo is:

```
GEOMECH SCANNING MODE
=====================
Perform geomechanics calculations at times scanned from input SR2 file:

Opened INDEX-IN         on unit 19, filename is 'stgeo006_1.irf'
Opened INDEX-IN         on unit 19, filename is 'stgeo006.irf'
Opened MAIN-RESULTS-IN  on unit 20, filename is 'stgeo006.mrf'
     At time step #0         0.0 days read PRESSURE from SR2 record GRID-VALUE.
     At time step #1  5.0000E-02 days read PRESSURE from SR2 record GRID-VALUE.
     At time step #1  5.0000E-02 days perform geomechanics update.

     At time step #2  1.5348E-01 days read PRESSURE from SR2 record GRID-VALUE.
     At time step #2  1.5348E-01 days perform geomechanics update.
...
     At time step #10    22.36964 days read PRESSURE from SR2 record RESTART.
     At time step #10    22.36964 days perform geomechanics update.

Opened INDEX-IN         on unit 19, filename is 'stgeo006_1.irf'
Opened MAIN-RESULTS-IN  on unit 20, filename is 'stgeo006_1.mrf'
     At time step #11    41.88658 days read PRESSURE from SR2 record GRID-VALUE.
     At time step #11    41.88658 days perform geomechanics update.
...
     At time step #16   365.00000 days read PRESSURE from SR2 record RESTART.
     At time step #16   365.00000 days perform geomechanics update.

     Done scanning SR2 file.
```

The post-processing run opens the parent SR2 file "stgeo006_1.irf", sees that it has a parent run "stgeo006" and opens those SR2 files instead. Time steps 0 (initial conditions) to 10 are processed from "stgeo006". After that the SR2 files for "stgeo006_1" are re-opened and scanning continues until the end at time step 16.

Pressure is read from either a GRID-VALUE or a RESTART record in the SR2, so it does not matter if the parent run wrote restart records or not. Temperature is not read since this is an isothermal run. In parent run "stgeo006", p and T were dumped every time step for illustration purposes, but normally they are dumped at recurrent *TIME and *DATE times.

GEM: Geomechanics Post Processing of Prior Simulator Response (Optional) *GEOPOSTPRO

PURPOSE:

Specify that one-way geomechanics calculation will be carried out using prior simulator response.

FORMAT:

*GEOPOSTPRO (*ippbeg*, *ippend*)

DEFINITIONS:

*GEOPOSTPRO

Indicates that the geomechanics calculations are to be done between two indicated timesteps by using pressure/temperature response from prior simulation.

*GEOPOSTPRO keyword can be followed by two positive integers indicating the first and last SR2 or restart record to be processed for one-way geomechanics calculation. The two integers should fall within the timestep numbers on which the SR2 or restart was written in the prior run from which post-processing is to be done

ippbeg

The starting or beginning timestep number

ippend

The finish or ending timestep number

DEFAULTS:

If *GEOPOSTPRO keyword is absent, no geomechanics calculations will be done by post-processing of prior simulator response. The default values for *ippbeg* and *ippend* are:

ippbeg = 1

ippend = 99999

The defaults are enforced when *GEOPOSTPRO is not followed by any number, or *ippend* is less than *ippbeg*.

CONDITIONS:

This keyword, if present, should appear in the Geomechanics section only on a separate line anywhere after *GEOMECH keyword.

EXPLANATION:

The one-way geomechanics calculations can be performed by using pressure/temperature output from a prior run if these have been written to GEM's simulation results (SR2) files or restart (RST) files. *GEOPOSTPRO, if followed by start- and end-timestep numbers, signals

GEM to scan for pressure and temperature input from prior simulation SR2 or restart files between these two timesteps. The geomechanics calculations are performed when pressure/temperature values are obtained at a timestep. If no pressure/temperature information is available in prior simulation SR2 file, no geomechanics calculations are done. It implies that ippbeg and ippend may not necessarily be the first and last timesteps at which geomechanics calculations would be done. Note that pressure is a prerequisite for both isothermal and non-isothermal runs, whereas temperature output (if available) is used only for non-isothermal runs.

Example:

```
*GEOMECH
   ** Perform one-way geomech between timesteps 7 through
   ** 55 of the prior GEM run
   *GEOPOSTPRO  7  55
```

Methodology

The SR2 or restart record is first created by running GEM and writing out pressure and temperature (if non-isothermal) arrays at different times including the ones at which geomechanical calculations are desired in subsequent post-processing GEM runs. The post-processing GEM run has a data file that is like a regular GEM data file with specification of input index results file (IRF) and additional specification of beginning and end of post-processing timestep via a new keyword *GEOPOSTPRO in the geomechanics section.

The post-processing GEM run in addition to regular tasks, scans for presence of *GEOPOSTPRO keyword in the data file to set an identifying flag that distinguishes this run to be different from a regular GEM run.

The post-processing GEM run requires presence of a prior SR2 or restart file and its specification in the data file. The GEM log file (*primary*) for post-processing runs is different from the regular runs, and shows only the status of processed SR2 or restart records. The *secondary* log file is not generated.

Though the post-processing GEM run can use either a SR2 or a restart file, it is not a restart*ed* run.

After reading the SR2 or restart data, geomechanics calculations are performed for the desired timestep number for geomechanics parameters using the pressure and temperature arrays read. The process ends when the last requested timestep or last record in the SR2 (RST) file, whichever is earlier, is reached.

Note that usual fluid flow time-stepping computations are skipped in the post-processing run. The feature was initially introduced in GEM 2010.12, where only restart files were scanned for post-processing. Version 2011.10 and later versions on the other hand can use both restart and SR2 files for getting the prior simulator response; the latter may be more convenient in certain situations.

There is some difference in the manner in which geomechanics data input is handled in the recurrent section in the post-processing runs. Please refer to the Tutorial Section in GEM Users' Guide on how to make a GEOPOSTPRO run.

GEM: Data Source from Prior Simulator Response for Geo-Post-Processing (Optional) *GEOPOSTINPUT

PURPOSE:

Choose data source of prior simulator response, between GEM's SR2 and RST files for a geo-post-processing (GEOPOSTPRO) run.

FORMAT:

*GEOPOSTINPUT *SR2 | *RST

DEFINITIONS:

*GEOPOSTINPUT
: Indicates that the type of input file to use for geomechanics calculations for geo-post-processing.

*SR2
: Use a simulator results file generated via prior simulation.

*RST
: Use a restart file generated via prior simulation.

DEFAULTS:

If *GEOPOSTINPUT keyword is absent, SR2 file from prior simulation is used.

CONDITIONS:

This keyword, if present, should appear in the Geomechanics section only on a separate line anywhere after *GEOMECH keyword.

EXPLANATION:

The pressure and temperature output from a GEM run is always written to GEM's restart (RST) file along with other variables required for restarting the run, at the times specified by the user. Similarly, pressure and temperature output can be written to simulation results file system (SR2) at user specified times, along with any other arrays needed for visualization. In GEM the geomechanics post-processing run can use the pressure and temperature information from either SR2 or RST file.

Keyword *GEOPOSTINPUT helps user choose one of these two files for geo-post-processing. The default is to use SR2 file system. Note that writing restart runs frequently during a GEM run can make the restart file very large, since a large amount of information is written in addition to pressure and temperature required for geo-post-processing. On the other hand the amount of information written to SR2 file system can be controlled by the user via *OUTSRF *GRID keyword. Of course *PRES, and *TEMP (in case of non-isothermal runs) need to be specified along with other arrays, if needed for visualization, for the SR2 file system to be useful for a subsequent geo-post-processing.

Though writing of pressure (*PRES) array is always necessary to create the prior simulator response for geo-post-processing, for isothermal runs writing of temperature array (*TEMP) is not mandatory.

It is necessary to note that though *GEOPOSTINPUT enables user to specify which of the prior simulator response files are to be used for geo-post-processing, none of the irf, mrf, or rst files from prior simulator run should be removed to run GEM in geo-post-processing mode.

Also see related keyword *SEEKPARENT in the Input/Output section of the GEM Users' Guide.

Well and Recurrent Data

Notes on Well and Recurrent Data for the 2016.10 Release

***BHPDEFAULT keyword**

For wells that do not have an explicit definition of the bottom-hole pressure constraint, the default values can now be assigned by the user by a global keyword at the beginning of a run (*BHPDEFAULT *PROD | *INJE *val*). This keyword does not affect wells that already have the BHP constraint defined through keyword *OPERATE.

Individual Well Completion Layer Control

The new keyword *LAYER-CTRL allows modification of layer status and / or well indices for individual well completions without having to redefine the entire well. (*LAYER-CTRL *wn* action, where action = *OPEN | *CLOSE | *WIMULT *val* | *WIMULTO *val*).

Allow *AUTOLAYER in Injector's *BACKFLOW Monitor Constraint

Now *AUTOLAYER action is permitted in injector's *BACKFLOW monitoring constraint.

New Keyword *RECYCLE_GEM

A new keyword, *RECYCLE_GEM, is introduced as an alias to the GEM-only (primary) keyword *RECYCLE. The latter was conflicting with the general sub-keyword of *GCONI. Primary keyword *RECYCLE is still supported but is deprecated.

Notes on Well and Recurrent Data for the 2015.10 Release

GENERALIZED INPUT OF WELL PERFORATION DATA

The *PERF keyword reading has been generalized to allow layer-wise input of all the parameters involved in Well Index calculations. The new layer table structure is very flexible and backward compatible. It includes *WI, *SLIMTUBE, *DIR, (*GEO | *GEOA), (*KH | *KHA), *RE, *RW, *SKIN , *TURB, and (*QUAD | *PSEUDOP). For entering the {well index} table, the user may put a dash ('–') at where a number is expected to indicate that the corresponding value will be taken from the defaults.

WELLBORE RATE STABILITY MONITORING

A new subkeyword *RATSTAB to *MONTOR keyword is added, that can be used for wells for which the method of computing WHP is to use the hydraulic tables. The keyword ensures that a real well would operate only above a certain rate at which the bottom-hole pressure remains in the stable region of the tubing-intake or tubing-outflow curve (a region where the BHP vs. rate has a positive slope for producers or negative slope for injectors while WHP is held constant). This monitor is ignored if the well operating conditions are out of the range of the used hydraulic tables.

USER SPECIFIED LIMITS ON WELL INDICES

Auto well index thresholding is implemented to take action on out of range WIs. If WI exceeds the value specified by second value of the range, it is set to the value of the second range. For any WI less than first range it is either set to first range or set to zero depending on the qualifying sub keyword. For mean and median options, the range checking is done after multiplying the specified values to mean or median. The range regulation is available on a well-wise basis and can be turned off in recurrent when necessary, via new keyword *WIRANGE.

CUSTOM OUTPUT OF WI AND WI STATISTICS

A new keyword *WELPRN is now available for customized output of well indices and well index statistics for all or selected number of wells that can be turned on or off as desired. The statistics may be used as a guide for selecting parameters for *WIRANGE.

BHO, BHG AND BHL CONSTRAINTS FOR PRODUCERS

Bottomhole Oil (BHO), Bottomhole Gas (BHG) and Bottomhole Liquid (BHL) constraints are made available for production wells, as subkeywords of *OPERATE.

HYDRAULIC RADIUS FOR WELLBORE FRICTION CALCULATIONS

A new sub-keyword *HYRADIUS of *HEAD-METHOD is added to introduce the hydraulic radius for the gravity-friction head calculations. The frictional part in the head calculations now uses the hydraulic radius which will override the default well radius from either *GEOMETRY or *PERF *RW.

TRIGGER ENHANCEMENTS

New subkeywords *BHTEMP and *WHTEMP are added that can be used to specify bottom-hole- or well-head temperature respectively in setting up triggers on wells.

WHT CALCULATION VIA HYDARULIC TABLES

The *PTUBE1 option has been enhanced to include well head temperature entries in the hydraulic tables. This allows GEM to calculate well head temperature using these tables. New keyword *TWELLBORE allows the specification hydraulic table which will be used for well head temperature calculation.

Notes on Well and Recurrent Data for the 2014.10 Release

GENERALIZED INPUT OF WELL GEOMETRY PARAMETERS

Keyword *WELGEO allows modification of well direction, radius, geometric factor, well-fraction, or skin for a given list of wells. These parameters are used by the simulator to calculate the well index internally.

WELLBORE HYDRAULIC TABLE ENHANCEMENT

Wellhead enthalpy can now be input via new sub-keyword *ENT of *ITUBE1 for non-isothermal runs.

TRIGGER ENHANCEMENTS

*TRIGGER, ON_GROUP (well stream and separator oil and gas streams)

Specify trigger conditions on mole percent of a component for the separator oil and/or gas streams for groups (*MPLS,*MPVS,*MPWS).

SIMPLIFICATION IN LAYER LUMP INPUT DATA SPECIFICATIONS

*LAYERCLUMP, *LAYERRLUMP, *RM-LAYERCLUMP, *RM-LAYERRLUMP

Previous versions of GEM required each layer to be individually specified when defining (creating) control and/or reporting layer lumps. In order to simplify (condense) the input data requirements a number of layers may now be collectively referenced using a range in any of the principal directions and at any refinement level. As before these layers have to have been specified earlier for the well by a *PERF card. These changes are fully backward compatible and all existing cases with control/report lumps should run as before and no new keywords are introduced. For more details refer to manual pages for the above noted keywords.

USE CURRENT CONSTRAINT WITHOUT CHECKING FOR ITS VIOLATION

New keyword *WCURRCN enables wells to run on their current operating constraints without checking for constraint violation and possible switching. This keyword allows the user to operate wells on their current constraints without checking for constraint violation and possible constraint switching to the most restrictive. This effectively disables all the other constraints while *WCURRCN *ON is active. Furthermore, wells with *WCURRCN turned on are automatically removed from participating in the apportionment of higher-level group production or injection targets. However, their rates still contribute to the higher-level targets since the wells are still members of the group structure.

SETTING INJECTED FLUID COMPOSITION

New keyword *INCOMPWL allows setting of the composition of injected lift-gas for producers or injected fluid for hydrocarbon (solvent) injectors.

WILDCARDING FOR GROUP NAMES

‘*’ Wildcard characters may now be used in the group names.

KEYWORD DEFAULT CHANGES

*APPOR-METHOD – The default has been changed to *INGUIDE, from *IP.

All runs using Group Control will be affected. To reproduce earlier behavior, use *APPOR-METHOD *IP for each group.

Notes on Well and Recurrent Data for the 2013.10 Release

TRIGGER ENHANCEMENTS

**TRIGGER trigger_definition *AVRGTIME average_time_value*

The trigger functionality allows for specific actions to be taken if the specified trigger condition such as a surface rate of a well or group is satisfied. In addition to the instantaneous values of rates or production ratios, it is now possible to specify a moving average of a rate, GOR etc over a specified time period using the keyword *AVRGTIME as a trigger condition.

**TRIGGER, *ON_WELL, ON_GROUP(lift gas)*

Specify trigger conditions on lift gas for wells and/or groups. The lift gas rates, cumulative and production ratios such as GOR can be specified as a trigger condition.

**TRIGGER, ON_SECTOR (maximum & minimum)*

Specify maximum and minimum in saturations, pressure or temperature for a sector. This adds to the previously available average saturation, pressure and temperature in a sector.

AUTODRILL OPTION

**DRILLT time_prd_groups time_inj_groups*

*DRILLT allows the user to enter a time delay between successive well automatic drilling operations. *DRILLT applies to drilling priority specified via *DRILLQ keyword or to the case where the priority list is generated internally by algorithms such as *IPP.

**DRILLMXW nwells_prd_groups nwells_inj_groups*

*DRILLMXW allows the user to specify the maximum number of wells with auto drill status that can be opened at one time. In conjunction with the *DRILLT keyword, this keyword enables the user to specify a prescribed drilling schedule, for example choosing to drill a maximum of 4 wells every month. *DRILLT and *DRILLMXW can be used to define an upper limit to the number of wells that can be drilled within a given time interval. *DRILLT is required for *DRILLMXW to have any effect.

**DRILLQ *IPP|*IPPQT|*IPPDRNINP|*IPPDRNINPQT*

or

**DRILLQ 'well-1' 'well-2' … 'well-n' *IPP|*IPPQT|*IPPDRNINP|*IPPDRNINPQT*

*DRILLQ allows the user to either specify a list of wells, with the order in the list defining the drilling priority, or allow GEM to calculate the priority order internally by a ranking algorithm such as *IPP, or as a new option, specify a subset of wells with autodrill status as a list and let the internal algorithm rank the remaining wells.

As well with this release, new algorithm(s) for ranking (prioritizing) wells with auto drill status are available in addition to *IPP. *IPP (instantaneous production potential) ranks wells based on the maximum rate of the target phase that each well can produce. The maximum rate is calculated assuming a BHP of one atmosphere. The new algorithm is predicated on the concept that in addition to high productivity, a large in place resource (i.e. target phase) in the vicinity of a well is desirable. Therefore given two wells with relatively equal productivity (that is maximum rate) the well with the higher in place amounts of the target phase is opened

first. To calculate and allocate in place amount of the target phase a reservoir volume or a drainage radius for a given production well is required. This drainage radius can be entered by the user for each well using keyword *DRNRADW. If a value is not specified then the internal default is 745 ft which corresponds to a 40 acre spacing. A negative value results in a lower priority since wells with a positive drainage radius will be opened first.

**DRNRADW well_numbers|well_names well_drn_values*

Specify a drainage radius for each of the listed production wells by name or number. The drainage radius is used to calculate in place amounts for each well and surface stream.

**DRILLALTP 'group_name' *STO|*STG|*STW|*STI|*STL|*WTG|*BHF*

Specify an alternate phase for ranking wells with auto drill status. The primary list consists of wells that can produce the target phase at a rate greater than a specified minimum rate. For wells that cannot produce the minimum target phase rate, a second lower priority list is generated with the ranking based on the maximum rate of the alternate phase.

**DRILLQTREFP 'group_name' qminrefp*

Specify a minimum rate of target phase. Wells that can only produce below the minimum rate are assigned to a lower priority list.

**DRILLQTALTP 'group_name' qminaltp*

Specify a minimum rate of alternate phase. Wells that can only produce below the minimum rate are assigned to a lower priority list.

New Keywords: *OPERATE-HIST, *ALTER-HIST

These new keywords allow operating wells in the special history matching mode. It is useful for making the well produce/inject correct amount of voidage with approximately correct rate of pressure changes before the mobility ratios are fully matched.

New Sub-keyword: * WCUTBACK under *OPERATE

A new group of sub-keywords that allows operating wells on a reduced rate target of the specified control phase whenever the wells violate certain stream-ratio (*GOR, *GLR, *WOR, *WGR) or well block-pressure (*PRS) types of constraints.

New Sub-keyword: *GTARGET

Sub-keyword *TARGET under group control *GCONP / *GCONI has been renamed to *GTARGET in order to differentiate with the primary keyword *TARGET for well controls.

Notes on Well and Recurrent Data for the 2012.10 Release

Use of On-Time Averaged Rates in Layer Lump Calculations

New keyword *LAYERLUMP_ACTR selects actual or on-time averaged rates and ratios based on the rates in all calculations involving lumps. The alternative (default) is to use instantaneous rates in layer lump calculations.

Removing Targets from Group Control

The keywords *GCONP ‘group_name’ *GTARGET *NONE will remove any production target specified previously (including *VREP, *RECYCLE and *PMAINT) for group ‘group_name’. Existing maximum / minimum constraints are not affected.

The keywords *GCONI ‘group_name’ *GTARGET ‘stream_name’ *NONE will remove any previously specified injection target (including *VREP, *VREFP, *RECYCLE and *PMAINT) for the indicated ‘stream_name’ (gas or water) and group ‘group_name’. Existing maximum constraints are not affected.

Trigger on Reporting Group

The keyword *TRIGGER has a new subkeyword for trigger definition: *ON_REPGRP ‘reporting_group_name’ ‘reporting_group_condition’. This subkeyword indicates that the trigger test condition is to be applied to a reporting group ‘reporting_group_name’. The reporting group must be previously defined to enable reporting group based triggers to be used. See the *TRIGGER documentation for the choices available for ‘reporting_group_condition’.

New *MONITOR Capability

The keyword *MONITOR *SHUTLAYER1 will plug the most offending layer (which, when shut, causes the well to have the lowest/highest value of the monitored variable) and continue the simulation. *SHUTLAYER1 has the same effect as *SHUTLAYER for a rate type monitor (e.g. STO), but will be different for a ratio type monitor (e.g. GOR).

The keyword *MONITOR *AUTOLAYER1 is similar to *AUTOLAYER but corresponding to *SHUTLAYER1.

Reinstating Groups or Wells to Group Production/Injection Targets

Keywords *GCPON and *GCION have been introduced to return groups or wells to the apportionment of higher level production or injection targets, if they have been removed from the apportionment through use of the *GCPOFF or *GCIOFF keywords.

Notes on Well and Recurrent Data for the 2011.10 Release

*FRAC Added to *WELL Keyword

Well fractions can now be defined per well and is useful for defining fractional (½, ¼, etc) wells when simulating elements of symmetry. Rates and well indices for full well are input, *FRAC determines the fraction of the rate and index to use. See the *WELL keyword for more information.

*AND and *OR Conditions in Triggers

The trigger facility allows for conditional execution of certain user specified actions related to well and group control. Whereas previously only a single condition per trigger could be entered by the user, it is now possible for the user to specify multiple conditions for a single trigger. This essentially constitutes a compound logical statement with connecting logical operators AND/OR. The user also needs to assign a priority to each logical operator as this influences how the overall compound logical statement is evaluated. For more information refer to the explanation section of the *TRIGGER keyword in the well and recurrent data section of the user manual.

Voidage Replacement Injection Group

The new keyword *VRI_GROUP is used to specify the injection group from which injection rates will be taken to calculate a production recycling or voidage replacement target for a given group or groups. Please see keyword *VRI_GROUP for more information.

Voidage Replacement by Pattern Capability

A group named as a source for recycling or voidage replacement (in a keyword like *GPRODGROUP) can now be a reporting group.

Since reporting groups allow for fractional well membership, use of a reporting group as a contributing or nominating group for recycling or voidage replacement enables imposition of injection targets based on a pattern type flood (using a reporting group which represents an element of symmetry). For example, the source group could be made up of the four ¼ producing wells surrounding an injector.

The injection group defined using *VRI_GROUP may be a reporting group. This allows fractional well membership, allowing production targets based on pattern injection.

Output of Well Productivity Index

The keyword *OUTSRF *WELL *WELLPI can be used to specify that well productivity index information be sent to Results for plotting.

Notes on Well and Recurrent Data for the 2010.10 Release

NEW KEYWORDS AND OPTIONS FOR GEM 2010.10

1. Completion Lumping Option
2. *WHYSTAB *AUTOWELL Monitor
3. *MXCNRPT Maximum number of Continue Repeats

Completion Lumping Option

*LAYERCLUMP provides a way for the user to define a control lump - a set of layers (perforations, completions) which possesses a name and a set of computed rate, cumulative, and stream ratio values. To each control lump there is also an associated setting value.

*LAYERRLUMP provides a way for the user to lump well layers in a group called a report lump. For report lumps, stream rates, cumulatives and production rate ratios are calculated and the user can design triggers based on these quantities. A given layer can be assigned a partial membership to report lump and the same layer can belong to more than one group.

*RM-LAYERCLUMP removes membership of a layer from the named control lump.

*RM-LAYERRLUMP removes membership of a layer from the named report lump.

*CLUMPSETTING assigns a control setting or flow restriction parameter to the named control lumps. The flow restriction parameter is used as a multiplication factor in adjusting the value of the well index for all the layers belonging to a given control lump.

There is also a set of computed rate, cumulative, and stream ratio values for each control lump. Triggers may be defined using these quantities specific to named control lumps; actions can be any valid well data lines, including lines which alter the control lump setting values. The syntax of the control lump based trigger statement is:

*TRIGGER ‘trigger_name’ *ON_CTRLLUMP ‘control_lump_name’ quantity > value

The syntax of the report lump based trigger statement is:

*TRIGGER ‘trigger_name’ *ON_RPTLUMP ‘report_lump_name’ quantity > value

See *LAYERCLUMP keyword for a list of quantities which can be used in control and report lump triggers.

***WHYSTAB *AUTOWELL Monitor**

“*MONITOR WHYSTAB *AUTOWELL frequency” can be used to monitor production well wellbore hydraulics stability using the *AUTOWELL option, which shuts the well for a specified time.

***MXCNRPT Maximum Number of Continue Repeats**

Every well having *CONT *REPEAT (continue-repeat) specified through *OPERATE is entitled to one repeat of the current timestep if one of its operating constraints is violated. During convergence of a timestep, it is possible that different wells violate and then switch their operating constraints consecutively after each repeat. Although the maximum number of possible repeat cannot exceed the number of wells, it may lead to a significant increase of run

time if there are many wells. Quantity nmxrpt limits the number of such repeats per timestep, and keyword *MXCNRPT lets the user override its default value (of 1).

Notes on Well and Recurrent Data for the 2009.10 Release

NEW KEYWORDS AND OPTIONS FOR GEM 2009.10

1. *ITUBE1/*PTUBE1
2. *GCONI ‘Stream’ *PMAINT *PMMAXR d1 d2
3. *GCONPMULT *RECYCLE
4. *PERFRG keyword removed

***ITUBE1/*PTUBE1**

PTUBE has been replaced by the more flexible PTUBE1, in addition ITUBE1 allows for the definition of injection tables using CMG standard well head pressure input table format. *PTUBE tables are still supported.

***GCONI ‘Stream’ *PMAINT *PMMAXR d1 d2**

Introduces the maximum surface rate of the specified injection stream that is available for the pressure maintenance. It is the summation of two sources: a fraction (d1) of the total produced (from the same group) and a make-up rate (d2). The values must be non-negative real numbers. The fraction d1 is nondimensional and defaulted to 0. The make-up rate d2 is (m3/day | bbl/day | cm3/min) for water and (m3/day | scf/day | cm3/min) for gas or solvent, and is defaulted to 1.0e+20.

***GCONPMULT *RECYCLE**

This subkeyword specifies a recycling production target. This indicates that the production wells connected to this group produce such that the phase injected by the injection wells connected to this group as specified by *GAS, *WATER or *SOLVENT is reproduced (recycled) out of the reservoir.

***PERFRG keyword removed**

This keyword is no longer required as the *PERF keyword can be used instead.

New Keywords and Options for the 2008.10 Release

NEW KEYWORDS AND OPTIONS FOR GEM 2008.10

1. Cycling Groups
2. *GAPPOR *AUTODRILL “stream identifier”
3. New group constraint monitor MINREC and action SHUTCYCINJ

Cycling Groups

Allow the user to specify data for EOR/IOR processes which rely on alternating production and/or injection cycles. Generally multiple cycles are required and each cycle consists of a number of well-defined parts. A typical example is water alternating gas (WAG) injection.

*GCONCYCLE_START allows the user to begin to specify data for EOR/IOR processes which rely on alternating production and/or injection cycles.

*GCONCYCR_START allows the user to revise data specified earlier with *GCONCYCLE_START. The revisions will be applied immediately.

*GCONCYCLE allows the user to turn on or off group cycling control specified earlier with *GCONCYCLE_START. The action indicated with GCONCYCLE takes effect immediately.

*CYCPRT_START allows the user to select the cycle starting part. CYCPRT_END allows the user to select the cycle ending part. Group cycling control must be specified earlier with *GCONCYCLE_START. The action indicated with CYCPRT_START and/or CYCPRT_END take effect immediately.

GAPPOR Stream Specification

The group control may direct a group to switch target (value and stream) if there is a maximum constraint (*GCONP *MAX) being violated. *AUTODRILL followed by the stream identifier ensures that auto-drillable wells can be opened only when the group is apportioned for the desired stream.

Group Constraint Monitor MINREC and Action SHUTCYCINJ

The keyword MINREC specifies that if the total group gas rate available for recycling falls below a specified value, then a remedial action will be taken. This remedial action may be one of *STOP, *SHUTALL or the new action *SHUTCYCINJ. *SHUTCYCINJ indicates that all currently open gas cycling injectors in the group should be shut.

Notes on Well and Recurrent Data for the 2007.10 Release

NEW KEYWORDS AND OPTIONS FOR GEM 2007.10

1. Shut in Wells above Formation *MODELSHUT
2. Two New Drawdown Constraint Types *DWA, *DWB
3. Monitoring Well Head Pressure Stability *WHYSTAB
4. Voidage Replacement Production Target
5. Recycling Production Target *RECYCLE
6. Sector Pressure Maintenance Production Target *PMAINT
7. Shutting in Multiple Wells to meet Production Targets/Monitors *SHUTMOWS
8. Voidage Replacement Injection Targets including Makeup of Gas or Water *GMKUP, *WMKUP
9. Sector Pressure Maintenance Injection Target *PMAINT
10. New Apportionment Methods for Meeting Group Targets *APPOR-METHOD, *IP, *GUIDE, *INGUIDE, *PRIOR, *PRIOR-FORM
11. On-Time Factors Applying to Groups

Shut in Wells above Formation

*MODELSHUT indicates that the listed wells identified by well_list will be closed above the formation with crossflow modelling of the constituent (open) layers, whenever the wells are shut in due to a well operation.

Two New Drawdown Constraint Types

Well control keyword *OPERATE has two new subkeywords:

*DWA defines the maximum drawdown within all open layers,

$$\Delta P_d = \max_{l,\,open} \pm \left(Pblock_l - Pwell_l\right)$$

which corresponds to the notion of imposing the maximum drawdown constraint to avoid formation damage.

*DWB specifies the average drawdown for all open layers:

$$\Delta P_d = \pm \sum_{l,\,open} PI_l\left(Pblock_l - Pwell_l\right) \Big/ \sum_{l,open} PI_l$$

weighted by the total product/injectivity (PI) at the reservoir condition.

Monitoring Well Head Pressure Stability

A real well can operate only above a certain rate at which the bottom-hole pressure remains in the stable region of the well-head pressure curve (a region in which the curve of WHP vs. BHP has a positive slope). New *MONITOR subkeyword *WHYSTAB allows you to monitor hydraulic stability for a well. It can only be used for wells for which a method of computing WHP has been introduced with the *PWELLBORE or *IWELLBORE keyword.

Voidage Replacement Production Target (*GCONP *VREP)

This indicates that the production wells connected to this group produce an amount of the bottom-hole fluid in proportion to the total bottom-hole fluid injected into the reservoir by the injection wells connected to this group.

Recycling Production Target (*GCONP *RECYCLE)

This indicates that the production wells connected to this group produce such that the phase injected by the injection wells connected to this group as specified by *GAS, *WATER or *SOLVENT is reproduced (recycled) out of the reservoir.

Sector Pressure Maintenance Production Target (*GCONP *PMAINT)

This subkeyword specifies that the group production rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector (*PMSECT) at a desired level (*PMTARG).

Shutting in Multiple Wells to Meet Production Targets/Monitors (Subkeyword *SHUTMOWS)

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then a list of prioritized most offending wells (MOWS – the wells with the higher rates of the named surface stream) should be shut.

Voidage Replacement Injection Targets including Makeup of Gas or Water (GCONI *VREP *GMKUP/*WMKUP)

A make-up stream can be added with *GMKUP or *WMKUP to meet a total voidage replacement fraction for each group under voidage replacement.

Sector Pressure Maintenance Injection Target (*GCONI *PMAINT)

This subkeyword specifies that the group injection rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector (*PMSECT) at a desired level (*PMTARG).

New Apportionment Methods for Meeting Group Targets

*APPOR-METHOD defines how to distribute a group target within its pool – the collection of all the contributing wells or groups. The pool for a targeted group consists of all its child groups and the wells directly attached. . CMG's multi-level well management module currently supports the following methods:

Instantaneous potentials (*IP, default)

Guide rates (*GUIDE)

Internally generated guide rates (*INGUIDE)

Priority ranking (*PRIOR)

*PRIOR-FORM defines the priority formulae and numerical control parameters for the priority ranking apportionment method set by *APPOR-METHOD *PRIOR to meet group targets.

On-Time Factors Applying to Groups

*ON-TIME specifies the fraction of time during which a well or group operates. Please see the *ON-TIME keyword for a detailed description of how *ON-TIME operates with Groups.

Notes on Well and Recurrent Data for the 2006.10 Release

Relaxation of some Restrictions upon Ordering of Well Data

In previous versions of GEM it was necessary to define a well with a *WELL line, assign it a type with *PRODUCER, *INJECTOR, or *CYCLPROD, and finally to specify perforations for the well with *PERF or a related keyword, all at the same *TIME or *DATE in the well data. Starting with the 2006 versions, a well can be defined with *WELL at one *TIME or *DATE, then return to the simulator with the well type and perforations not yet defined but with the well's status defaulted to *SHUTIN; then at a later *TIME or *DATE have the perforations defined, and again return to the simulator still with defaulted *SHUTIN status; and finally at a still later *TIME or *DATE define the well type with *PRODUCER, *INJECTOR, or *CYCLPROD (at which time the well status defaults to *OPEN). Operating constraints can then be defined and the well begins operation. For more information see the EXPLANATION section of manual pages for the Well Type Definition keywords (*PRODUCER, *INJECTOR, and *CYCLPROD). This change was made so that, for example, all wells could be defined at the beginning of the data set but *PERF lines could be placed in the data set at the times the perforation were actually done.

WLISTSHUT and WLISTOPEN

These two new keywords allow the shutting of a list of wells and their later opening without disturbing the pattern of open and shut wells which existed when the shutting list was defined. This allows, for example, a short field shutdown to be modelled without reopening previously shut wells when the shutdown is over. Please see the manual page for these two keywords for more information.

TRIGGER ACTIONS

In GEM it is now possible to define conditions which, when satisfied (the trigger) cause an action to be taken. Each trigger has an action separately defined for it; each action is a sequence of valid recurrent data lines (with some additions, such as place holders which allow action to be directed toward the violating well or group without knowing the well or group name in advance). This allows considerable flexibility in the definition of the action. Please see the manual pages for the TRIGGER, WTMULT, GCONPMULT, and GCONIMULT keywords for more information.

Notes on Well and Recurrent Data for the 2005.10 Release

Manifolds

A new facility has been added to GEM to represent (undersea) manifolds, and the pressure drop between the manifold and the surface platform. The pressure drop between the wells and the manifold are represented in the same way that tubing head pressures have been calculated in GEM. The pressure drop to the surface is computed by interpolation in a hydraulics table. Group production and injection can be treated as going through a manifold, and the manifolds support rate, manifold pressure, and surface pressure constraints. Only groups with wells directly attached can be represented as manifold groups. For more information see the manual pages describing the MANIFOLD, GCONP, GCONI, PCON-MANIFOLD, GPTABLE, GPHYDDEP, GROUPALQ, WELLALQ, and IWELLBORE keywords.

KRPERF

The KRPERF keyword is now accepted by GEM. This keyword allows the specification of a wider variety of special relative permeability endpoints for well completions.

Notes on Well and Recurrent Data for the 2004.10 Release

Two new group production monitors, MAXGAS and MAXSTW, have been added under the control of the GCONM keyword. These monitors trigger upon high gas and high water rates respectively, but when an action is specified which acts upon the “most offending” well or layer, “most offending” is deemed to mean the well or layer with the highest GOR or water cut, rather than that with the highest gas or water rate. See the manual page for the GCONM keyword for more information.

The operation of the *GAPPOR *AUTODRILL apportionment option has been altered so that automatic opening of wells to meet a group’s rate target can commence even when no wells are initially open in the group. When instantaneous production potential is used to rank wells for automatic drilling (using the *DRILLQ keyword), the instantaneous potential is defined as the well’s rate at a bottomhole pressure of 1 atm. When instantaneous injection potential is used to rank injectors for automatic drilling, the potential is defined as the well’s injection rate at a bottomhole pressure of 3500 atm.

The maximum well name length supported has been increased from 16 to 40 characters. The maximum group and reporting group name lengths remain at 16 characters. Please see the documentation for the *WELL keyword.

Notes on Well and Recurrent Data for the 2003.10 Release

Two major keywords have been added to the well data for this release: *HEADITER and *BHPHEADINIT. Both address the problem of getting the pressure differences between completions in a well, as calculated from frictional and gravitational terms, consistent with the flow rates resulting from the consequent completion wellbore pressures.

In many cases, for example vertical liquid-producing wells experiencing no backflow, the pressure differences depend only very weakly upon flow rates. In these cases the establishment of consistent wellbore pressures and rates is not difficult and no special intervention is needed to establish it. In some other cases, however, such as horizontal gas producers and wells in which many completions backflow or nearly backflow, the wellbore pressure differences depend very strongly upon the completion pressures. In these cases, when the pressure differences ("heads") are calculated once per timestep with no intervention to establish consistency, the wellbore pressures may oscillate from timestep to timestep among several values. This is manifested as rates of a non-controlled stream which oscillate between two (or more) values at succeeding timesteps.

*HEADITER allows the user to direct that the head calculation done at the beginning of a timestep should be done iteratively to achieve consistency between the completion pressures and the computed pressure differences between completions. *BHPHEADINIT allows the user to direct that the initialization of a well's bottom-hole pressure to give the target rate or tubing-head pressure should include resetting of the heads through an iterative calculation which enforces both the meeting of the target and the consistency of the completion pressures with the completion pressure differences.

Our experience has been that *HEADITER suffices to eliminate oscillations in horizontal wells in which only gas flows; *BHPHEADINIT is often necessary for backflowing wells with significant depth differences between completions and some liquid flow. The iterations may not converge; in this case the heads resulting from the first step of the iteration (those that would have been obtained without invoking *HEADITER or *BHPHEADINIT) are used.

Please see the relevant manual pages for more information.

Notes on Well and Recurrent Data for the 2002.10 Release

There are three major additions to well data for this release: the correlation-table for computing tubing head pressure using the pressure-drop correlation in GEM (the *WHPCRTB keyword), the introduction of reporting groups (the *REPORTING-GROUP keyword), and the addition of a bottom-hole fluid (BHF) group production rate target and monitor (see the entries for the *GCONM, *GCONP, and *GUIDEP keywords).

The purpose of *WHPCRTB is to allow faster simulations using the WHP correlation with nearly identical results.

The purpose of the reporting groups is to allow reporting of results for well aggregates which are very flexibly defined, without the restrictions which apply to the hierarchical groups through which group controls are imposed.

The BHF group target allows group production strategies to be set during prediction runs which may better allow reservoir pressure to be maintained while respecting maximum rate limits upon injectors when used in combination with the injection voidage replacement target.

There is also a minor keyword addition in this section. The keyword *NWHYFS has been added to give users extra control over some aspects of the numerical method used to compute the bottomhole pressure which yields a specified tubing head pressure. In nearly all cases the default will work well and the user need not be concerned with this keyword.

Please see the relevant manual pages for more detail.

Notes on Well and Recurrent Data for the 2001.10 Release

Enhancements to well management introduced in this release are in two areas: group recycling injection targets (keywords *GCONI, *GPRODGROUP, *WPRODGROUP, *GSALESFRAC, *GSALESMAXR,*GFUELFRAC, *GFUELMAXR, *WMKMAXR, *WMKUPTO, *GMKMAXR, *GMKUPTO, *GMKCOMP, *WRECYMAXR, *GRECYMAXR, *GSALESMASK, *GFUELMASK, and *GRECYMASK), and wellbore and bottom-hole pressures (keywords *BHPDEPTH, *BHPGRAD, and *LAYERGRAD).

There have also been some enhancements to the options available for the calculation of well indices from geometric data (the *GEOA and *KHA subkeywords of the *PERF (*PERFV, etc. keywords) and the *LAYERIJK keyword.

The enhancements in recycling allow fuel and sales rates to diminish the amount of gas available for re-injection (keywords *GSALESFRAC, *GSALESMAXR,*GFUELFRAC, *GFUELMAXR, *GSALESMASK, and *GFUELMASK) and allow the production fixing the injection target to come from a named group (keywords *GPRODGROUP and *WPRODGROUP). Also gas and water make-up injection can run in either a constant-rate mode or a mode in which the make-up rate is adjusted to fix total injection at a given rate if possible with the given well constraints (keywords *WMKMAXR, *WMKUPTO, *GMKMAXR, and *GMKUPTO). It is also possible to cap recycling injection rates at a given maximum value (keywords *WRECYMAXR and *GRECYMAXR).

The enhancements in the handling of wellbore and bottom-hole pressures allow a well's bottom-hole pressure to be referred to a user-specified depth rather than being the wellbore pressure in the reference completion (keyword *BHPDEPTH). It is also possible to specify the pressure gradient value which will be used to compute the difference between the wellbore pressure in the reference completion and the bottom-hole reference depth (keyword *BHPGRAD). In addition the user can now specify that certain layers (or all layers) of a well should use a specified pressure gradient for calculating the wellbore pressure differences between adjacent completions (keyword *LAYERGRAD).

*GEOA and *KHA indicate that the anisotropic Peaceman well model is to be used to compute well indices. This model accounts automatically for permeability anisotropy and grid block aspect ratio. It is available for the model-based well index calculations (*GEO => *GEOA, and *KH => *KHA). The geometry keyword remains as is, but the geometric factor C (normally 0.37) is ignored when *GEOA or *KHA is in force. In most cases the well indices obtained using *GEOA or *KHA do not differ greatly from those calculated using *GEO or *KH.

*LAYERIJK allows layer-by-layer specification of the direction (parallel to the I, J, or K axis) of the direction of perforation of a well.

Please see the relevant manual pages for more detail.

Notes on Well and Recurrent Data for the 2000.10 Release

GEM now accepts the *TURB subkeyword for the *PERF, *PERFV, *PERFRG, and *PERFVHY keywords to activate the turbulent skin modification of the layer reservoir gas phase productivity/injectivity calculation. An additional parameter value (the D-factor) must be entered for each layer on the perforation line when this option in is force. GEM now also supports two different ways of accounting for the layer gas phase productivity/injectivity changes due to the property changes which the reservoir gas undergoes as the pressure changes from the grid block value to the wellbore value. The first way, activated by the *QUAD subkeyword of *PERF, *PERFRG, *PERFV, and *PERFVHY, is the quadratic inflow technique, in which the gas viscosity is assumed to remain at the grid block value and the gas density is assumed to vary linearly with pressure. The second, activated with the *PSEUDOP subkeyword, includes a full recalculation of the gas density and viscosity at the wellbore pressure. No additional data are necessary for the *QUAD and *PSEUDOP modifications; the two cannot be applied simultaneously, as they represent two different ways of treating the same effect. Please consult the manual pages in this chapter describing the *PERF, *PERFRG, *PERFV, and *PERFVHY keywords for more information.

In addition, the *PERF keyword now accepts the general i1 j1 k1 / i2 j2 k2 / … format for specifying the grid block in which a layer is perforated. This means that all perforations into refined grids, including hybrid grids, can now be done through *PERF. The *PERFRG and *PERFVHY keywords are still supported but are no longer necessary. Please see the manual page for the *PERF keyword for more information.

Placement of Keywords in the Well and Recurrent Data

A large fraction of the keywords that belong to the well and recurrent data section can be located anywhere within this group; the *WELSEP and *DTMAX keywords are typical examples. Some keywords, however, are conditional on other pieces of information having been read in before they can be processed. This is necessary to ensure that all the required data is entered. The following guidelines provide an overview of the some of the limitations with regards to placement of some keywords. For greater detail please refer to the "Conditions" section listed under each individual keyword.

1. Groups must be defined before wells. If no groups are declared then all wells are considered to be connected directly to the field and no group control is possible. If groups are defined, then wells cannot be connected directly to the top-level group; wells not explicitly attached to a group are attached to an internally-created group which has the name 'Default-Group'. The top-level group, which is conceptually the field, does not have an assigned name; it acquires the name which the user assigns as the only name following *ATTACHTO in a *GROUP line which does not follow *GROUP on another line.

   For a more detailed discussion of the group hierarchy please refer to the section entitled "Well Management and Group Control" which immediately follows this section. A statement must appear to indicate which groups are connected to the 'FIELD'. The following is an example of valid data input for a case with group control. If any of the *WELL keywords had been encountered before the *GROUP keyword, then a warning message would be generated but the well would be attached to the group as directed and simulation would continue. This allows group structures to be defined late in a run, for example at the beginning of the prediction stage after a history has been simulated. Note that in this example, 'Field' is the only group name to appear after *ATTACHTO but not directly after *GROUP.

```
*GROUP 'GRP-1'      'GRP-2'        *ATTACHTO 'Field'
   *WELL 1 'PR-15,10'   *ATTACHTO 'GRP-1'
   *WELL 2 'INJ-5,11'   *ATTACHTO 'GRP-1'
   *WELL 3 'PR-03,03'   *ATTACHTO 'GRP-1'
   *WELL 4 'PR-10,13'   *ATTACHTO 'GRP-1'
   *WELL 5 'INJH2O '    *ATTACHTO 'GRP-2'
   *WELL 6 'INJ-6'      *ATTACHTO 'GRP-1'
```

2. Groups must be defined before any group operating or monitoring constraints are specified. The following example shows a correct sequence.

```
*GROUP 'GRP-1'      'GRP-2'  *ATTACHTO 'Field'
   *GCONP   'GRP-1'
      *GTARGET   *STO  274.0
      *MAX      *GOR  400.0  *SHUTMOW
   *GCONI   'GRP-1'
       *RECYCLE *GAS  1.00
   *GCONI   'GRP-2'
       *GTARGET  *STW  350.
```

3. The wellbore geometry data must be entered prior to the well perforation data, i.e. *GEOMETRY keyword should precede *PERF keyword.

4. The perforation specification keyword must be entered after the well has been defined with a *WELL line but may be entered before the well type is defined with *PRODUCER or *INJECTOR:

5. For injectors the well type declaration must be followed by the following sequence: the *IWELLBORE keyword if wellbore hydraulics calculations are desired or a *WHP constraint is specified, followed by *INCOMP to specify the injection fluid composition, and finally operating and monitored constraints. For producers the *PWELLBORE keyword, which is optional unless a *WHP operating constraint is imposed, should precede the well constraints. Examples are provided below:

```
*INJECTOR  1:2
  *IWELLBORE  ** THIS IS OPTIONAL IN THIS CASE, AS
              ** THERE IS NO *WHP CONSTRAINT
       500.0  500.0  0.0008  16.0  36.0  0.25
  *INCOMP    *SOLVENT    1. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  *OPERATE   *MAX  *STG  7000.0

*PRODUCER  3:4
  *PWELLBORE  *MODEL  ** THIS IS MANDATORY IN THIS CASE, AS
                      ** WHP CONSTRAINT IS SPECIFIED
       500.0  500.0  0.0008  16.0  36.0  0.25
  *OPERATE  *MAX  *STO    15.0
  *OPERATE  *MIN  *WHP  6000.0
```

6. If a list of well or group names following a keyword, a limited wildcard facility is available to allow the user to specify the desired list of wells or groups without having to name each well or group separately. For keywords that allow both the well list or group list (e.g. *ON-TIME), the wildcard characters can only be used for the well names. The description of this facility is as follows:

   a. There are two wildcard characters, '?' and '*'.

   b. Any number of ?'s can appear in a well/group name character string; each ? is counted as matching any character in the same position in a well/group name, including embedded blanks but NOT final blanks. For example, 'WELL??' matches 'WELL 1' but 'WELL?' does not match 'WELL'.

   c. A single * can appear as the last non-blank character in a well/group name character string; when * is present only the characters preceding * are checked for lack of match. For example, the character string '*' matches all wells/groups, and 'WELL*' matches 'WELL'.

   d. ? and * can appear in the same string; for example 'WELL????PROD*' would match 'WELL_NW_PROD_15' and 'WELL_SE_PROD_2'

   e. A list of the wells/groups matched when wildcard characters are used is printed in the output file so that the user may check the list generated.

Note on Recycling and Well Stream Compositional Monitoring

To use recycling with group control use *GCONI 'group_name' *RECYCLE *RECFRC (or *RATE). To maintain backward compatibility the stock tank gas recycle constraint (*STR) as well as the *RECYCLE_GEM keyword without group specification has been retained to be used when group hierarchy is not specified. They should not be used if group control is turned on. The program will not work if *STR is used for a cycling injector when a *GCONP, GCONI or *GCONM keyword is encountered in the data set. For groups the *STR constraint is not required since the *GUIDEI *STG keywords can be used to apportion the production from the group. Well stream compositional and minimum recycling rate monitoring can also be used with groups.

Miscellaneous Notes

The *DATE or *TIME keywords indicate a new well change time. Any well keywords that appear between two *DATE or *TIME cards will be applied between the times indicated. The first well change must be a *DATE keyword indicating the starting date of the simulation.

Well names are entered via the keyword *WELL. Wellbore geometries can be entered using the *GEOMETRY keyword, and *PERF specifies the grid blocks where the well is perforated. Refer to Appendix A for the calculation of well indices from well geometries and other information on the well model used.

The type of well is described by the keywords *PRODUCER, *INJECTOR or *CYCLPROD. If wellbore hydraulics calculation is desired or if a well head pressure *WHP operating constraint is specified then the *PWELLBORE for producers or *IWELLBORE keyword for injectors must follow. They must be followed by the operating conditions and constraints of the well, indicated by the *OPERATE and *MONITOR keywords. For injectors, the injection fluid composition must be specified by the *INCOMP keyword prior to the *OPERATE or *MONITOR keywords. All producers must be connected to a separator whose conditions are specified using the keyword *WELSEP. Wells can be connected to a group using the *ATTACHTO subkeyword under the *WELL keyword. If a well is not attached to any group it is attached to a group called 'DEFAULT-GROUP' by GEM. Any operating condition and its value can be altered at a later time using a single keyword *TARGET.

Minimum rate operating constraints have been replaced by minimum rate monitored constraints. If a minimum rate well constraint is specified, it is converted automatically to a monitored constraint and a warning message is printed.

A well can be shut at any time with the keyword *SHUTIN. Once the well type has been defined, a well may be reopened with the *OPEN keyword. When the well type is defined the well is automatically opened. Once type has been defined the well may be placed in an automatic drill queue by specifying its status as *AUTODRILL. The automatic drilling option is useful during the prediction phase of a simulation study, allowing specified group targets to be met by opening (drilling) new wells when a target cannot be met during apportionment.

The gas recycling model involves the *CYCLPROD keyword which indicates from which producers the separator gas will be recycled and the *RECYCLE keyword which indicates the additional processing of the cycling fluid stream. The injectors that re-inject the recycling gas stream will have the injection fluid specifications of *INCOMP *CYCLING. Refer to Figure 8 in Appendix B for more information on the recycling model implemented in GEM.

The wellbore models relate the bottom-hole pressures to the well head pressures of a well. For producers the tubing pressure loss can be calculated using two methods: a correlation, or multidimensional interpolation in a user-entered table. For injectors only the correlation option is available. The keyword *PWELLBORE or *IWELLBORE is required if a well head pressure constraint is specified or if information about the well-head pressure is desired. This keyword must be entered after the well type specification keywords (*PRODUCER, *INJECTOR or *CYCLPROD). Refer to Appendix A for more information on the correlation used by GEM and under the keyword *PTUBE1 and *ITUBE1 for documentation on the tabular entry option.

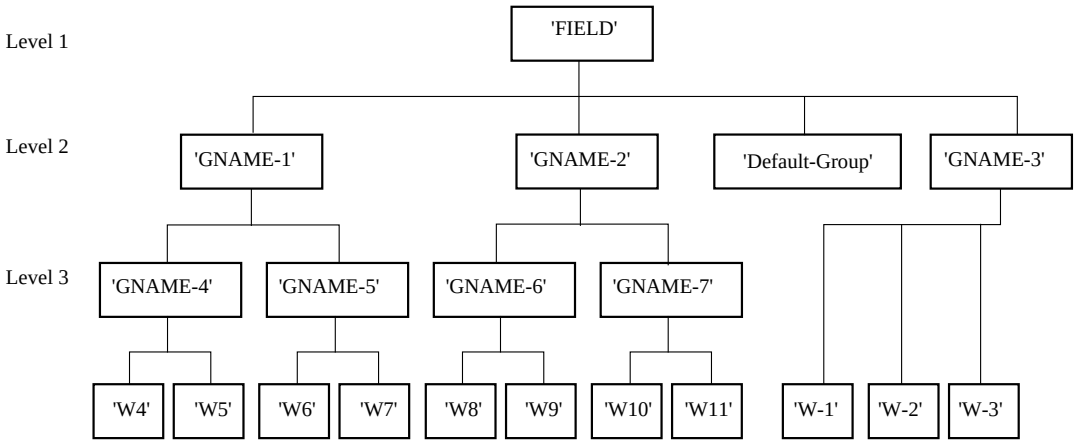
The simulation is started by *RUN and stopped by *STOP. The first timestep size used after a well change is controlled by *DTWELL. All timestep sizes are limited by *DTMIN and *DTMAX.

Initial implicit-explicit pattern of the grid blocks are controlled by the keywords *AIMSET and *AIMWELL.

Well Management and Group Control

The group hierarchy may have unlimited group levels; i.e., the hierarchy can consist of one top-level group, second-level groups connected to the top-level group, and third-level groups connected to second-level groups, and so on.

An example is shown below:



The highest level group is the 'FIELD'. The highest level is not optional. If *GROUP lines appear and either no top-level group is specified

Example:

```
*GROUP 'G1' *ATTACHTO 'G2'
*GROUP 'G2' *ATTACHTO 'G1'
```

or more than one top-level group is specified

Example:

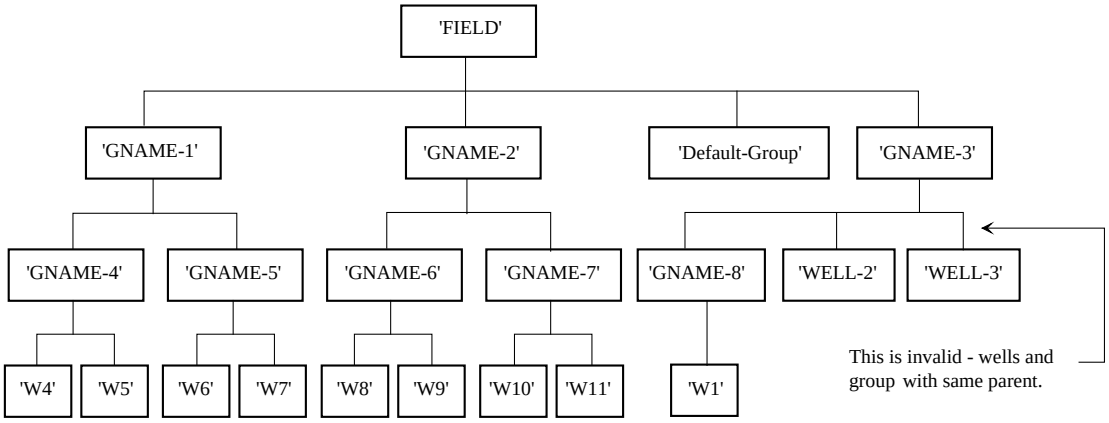
```
*GROUP 'G1' *ATTACHTO 'G3'
*GROUP 'G2' *ATTACHTO 'G4'
```

then an error is generated and simulation terminates.

Wells can be attached to any group except the field. A group to which a well is attached can have only wells attached to it, and not other groups. Wells and groups cannot be attached to the same group.

The group control hierarchy is optional. Thus all GEM data sets from previous versions that do not have gathering centres can be run essentially unchanged (aside from some rearrangement in the order that certain keywords are placed in the well and recurrent data section) with this version of GEM. If a group control hierarchy is used, then not all wells need to be attached explicitly to groups. Those wells that are not attached to a group by the user will be automatically attached to the internally-generated group 'Default-Group'.

An example of an invalid well-management hierarchy is given below:


'FIELD'
'GNAME-1'
'GNAME-2'
'Default-Group'
'GNAME-3'
'GNAME-4'
'GNAME-5'
'GNAME-6'
'GNAME-7'
'GNAME-8'
'WELL-2'
'WELL-3'
'W4'
'W5'
'W6'
'W7'
'W8'
'W9'
'W10'
'W11'
'W1'
This is invalid - wells and group with same parent.

Specifying the Well and Group Control Hierarchy

The control hierarchy for wells and groups is constructed using the *GROUP and *WELL keywords. Group controls (injection and production targets) and monitored constraints are specified by the *GCONP, *GCONI and *GCONM keywords. The injection and production distribution to wells and groups is done using one of the available apportionment methods specified by *APPOR-METHOD. The most offending well can be shut by using the *SHUTIN action under the *GCONM keyword.

Wells can be drilled automatically to maintain production or injection targets by specifying their initial status as *AUTODRILL and by specifying *GAPPOR 'group' *AUTODRILL *ON.

Well operating and monitored constraints are specified by using the *OPERATE and *MONITOR keywords. Well layers can be initially either specified as open or candidates for recompletion by using the *OPEN or *AUTO subkeywords under the *PERF or *PERFV keywords.

Well injectivities or productivities can be automatically improved by specifying the *WORKOVER action under the *MONITOR keyword. Other possible actions under the *MONITOR keyword are to plug the most offending layer (*SHUTLAYER), monitor the layer for possible recompletion (*AUTOLAYER), and monitoring the well for possible reopening when the GOR or water-cut decreases (*AUTOWELL).

Notes on Group Well Control

PRODUCTION CONTROL

Production controls are entered using keyword *GCONP 'group_name'. Target oil, gas, or intermediate liquid production rates can be specified for the centre using *GTARGET *STO, *GTARGET *STG, or *GTARGET *STI.

The target rate is apportioned among contributing producers using one of the available apportionment methods (e.g. guide rate, instantaneous potential, priority ranking, and internal guide rate) specified by *APPOR-METHOD. All producers must have a minimum BHP. If none is specified, a minimum BHP of 101.325 kPa is assigned.

Consider the following example of two producers connected to a group with an STO target constraint, assuming SI units:

```
*GCONP  'GRP-1'
     *GTARGET *STO    274.0
     *MAX    *GOR    400.0  *SHUTMOW
*WELL 1  'PR-15,10'         *ATTACHTO 'GRP-1'
*WELL 2  'PR-03,03'         *ATTACHTO 'GRP-1'
*PRODUCER  1
     *OPERATE   *MAX  *STO  6.0E+03
     *OPERATE   *MIN  *WHP  5.0E+02
*PRODUCER  2
     *OPERATE   *MAX  *STO  6.0E+02
     *OPERATE   *MIN  *BHP  1.0d+03
```

If the target is to be apportioned using the instantaneous potential method (*APPOR-METHOD *IP), the following steps are taken to estimate the instantaneous production potential (IPP) of wells 1 and 2 which will serve as the basis for apportioning the group target of 274.0 between the two wells. Well 1 does not have minimum BHP constraint specified, therefore a value of 101.325 kPa (14.696 psia) is assumed. The minimum WHP is converted to a minimum BHP. The larger of the two values is used to compute a maximum oil rate based on the productivity index of well 1 at the specific time in the simulation. The oil rate thus computed is capped to the maximum rate specified for the well, or:

IPP (well 1) = min [qoil{max(bhp=101.325, bhp@whp=500)}, 6000.0]

A similar procedure is used to compute IPP for well 2

IPP (well 2) = min [qoil(bhp=1000.0), 600.0]

INJECTION CONTROL

Injection controls are entered using keyword *GCONI 'group_name'. A target injection can be specified for the centre using *GTARGET *STG and *GTARGET *STW for solvent (gas) and water injection rates respectively, or *VREP *GAS and *VREP *WATER for gas and water voidage replacement fractions respectively. The target rate is apportioned among all injectors using one of the available apportionment methods (see *APPOR-METHOD) for the targeted injecting stream in a manner similar to that described above for producers. All injectors must have a maximum BHP. If none is specified, a maximum BHP of 1,000,000 kPa (147,000 psia) is assumed.

CYCLING CONTROL

All separator gas from the cycling producers (wells specified as *CYCLPROD) connected to the group is collected. The separator gas stream can be stripped selectively with recycling fractions specified using *GCONI 'group_names' *RECYCLE *RECFRC sep_comp_recyc_frac. In addition, a make-up rate for each component can be specified for each group using *GCONI 'group_names' *RECYCLE *RATE comp_make_up_vol. A user-specified volume fraction of the gas obtained by summing the stripped separator gas and the make-up streams can be recycled via the cycling injectors (wells with *INCOMP *CYCLING specified). If this fraction is less than one it can be interpreted as accounting for losses to aquifer, bypassing, sales, etc. The fraction is specified by *GCONI 'group_name' *RECYCLE *GAS recyc_frac . For greater detail on the cycling model refer to appendix B. The percentage recycled to each cycling injector can be stipulated using guide rates (*GCONI 'group_name' *GUIDEI *STG 'well_names' guide_rates). Do not use the *STR well operating constraint with group recycling.

NOTE: All solvent injectors belonging to a group should be of one type, either normal injectors or cycling injectors. Cycling injectors and normal injectors cannot both be connected to the same group. However, both water injectors and solvent (gas) injectors may be connected to the same group.

INDIVIDUAL WELL CONSTRAINTS

Each well can be subjected to its own rate and pressure constraints. If the rate allocated by the centre violates the well's own constraint, the well's constraint will be used. For example, if the gas injection rate of a well allocated by the group exceeds the well's maximum gas rate, the well's maximum gas rate will be used. If a well would violate its min. BHP limit while producing the allocated rate, the well will then produce at its minimum BHP. In these situations, the rates for the other wells will be readjusted to compensate for the differences. If all wells are under their own constraints, then the group target rate will not be maintained.

Thus, the group production target rate should be less than the sum of the max. production rates from all producers in the group; and the target injection rate should be less than the sum of the max. allowable injection rates from all group injectors, if the group target is to be met.

INTRODUCING GROUP CONTROL AFTER START OF SIMULATION

Group control can be instituted after the simulation has started, at a well change time or upon restart. The simplest example consists in introducing a field target after the simulation has run for some time. Assume that a run has been started with no group structure referred to, i.e. with wells defined using *WELL lines with no *ATTACHTO subkeywords defining parent groups. To establish a group structure, it suffices (at a well change time) to introduce the single line

```
*GROUP 'Default-Group' *ATTACHTO 'Field'
```

The name 'Default-Group' must appear exactly as shown in the above line, since it is an internally set name. The top-level group is given the name 'Field' in this example but the user is completely free to choose this name (up to a maximum length of 40 characters); 'Campo' or 'FIELD' would have exactly the same effect. Then to introduce a field target, it suffices to follow the above line with the line.

```
*GCONP 'Field' *GTARGET *STO 400.0
```

Here it is important that 'Field' match exactly the character string after *ATTACHTO in the *GROUP line; if 'Campo' had been used above it would have to be used here.

DATA INPUT

The following are all the keywords related to the group well control:

Group Specification

```
*WELL well_number ('well_name') (*ATTACHTO 'group_name')
```

Production Control

```
*GCONP group_list
  (*MAX)     (*STO)     value (*STOP)
             (*STG)           (*CONT)
             (*STW)           (*SHUTMOW)
             (*STL)           (*SHUTMOL)
             (*STI)           (*SHUTMOLDOWN)
             (*WTG)           (*SHUTMOLUP)
             (*BHF)           (*SHUTMOWS)
                              (*SHUTALL)
  (*MIN)     (*MNP)     value (*STOP)
             (*CCP)           (*CONT)
  (*GTARGET) (*STO)     value
             (*STG)
             (*STW)
             (*STL)
             (*STI)
             (*WTG)
             (*BHF)
             (*MNP)
             (*CCP)
  (*VREP)               vrep_frac
  (*RECYCLE) (*GAS)     recyc_frac
             (*WATER)
  (*PMAINT)  (*PMSECT) ‘sector_name’
             (*PMTARG)  p_targ
             (*PMCOEF)  c1 c2 c3

*APPOR-METHOD *PROD group_list
            (*IP | *GUIDE | *INGUIDE | *PRIOR)

*GUIDEP     (*STO) (group_list | well_list) guide_rates
            (*STG)
            (*STW)
            (*STL)

*PRIOR-FORM  *PROD group_list
  (*PRIOR-NUMER   A0 A1 ... Anph)
  (*PRIOR-DENOM   B0 B1 ... Bnph)
  (*PRIOR-RATE    (*MRC | *BHP(bhp_val) | *WBHP))
  (*PRIOR-CTRL    freq   tcr_min  tcr_max)

*GCPOFF (group_list | well_list)
*GAPPOR group_list *AUTODRILL (*ON|*OFF)
```

```
*GCONM group_list
  (*GOR)         value  (*STOP)
  (*WCUT)               (*SHUTMOW)
  (*WGR)                (*SHUTMOL)
  (*MAX)  (*MXX)        (*SHUTMOLDOWN)
  (*GWGR)               (*SHUTMOLUP)
  (*WWGR)               (*SHUTMOWS)
  (*MAXGAS)             (*SHUTALL)
  (*MAXSTW)
  (*MINOIL)      value  (*STOP|*SHUTALL)
  (*MININL)
  (*MINGAS)
  (*MINWTG)
  (*MINBHF)
  (*MINREC)
```

Group Injection Control

```
*GCONI group_list
  (*MAX)               value (*STOP)
  (*GTARGET) (*STG)          (*CONT)
             (*STW)
             (*BHG)
             (*BHW)
             (*GMP)
             (*WMP)
             (*GCP)
             (*WCP)
  (*VREP)    (*GAS)    vrep_frac
             (*WATER)
             (*GMKUP)
             (*WMKUP)
  (*RECYCLE) (*GAS)    recyc_frac
             (*WATER)  recyc_frac make_up_vol
  (*PMAINT)  (*GAS)    (*PMSECT)  'sector_name'
             (*WATER)  (*PMTARG)  p_targ
                       (*PMCOEF)  c1 c2 c3

*APPOR-METHOD  (*GASI | *WATI) group_list
  (*IP | *GUIDE | *INGUIDE | *PRIOR)

*GUIDEI (*STG) (group_list | well_list) guide_rates
        (*STW)

*PRIOR-FORM  (*GASI | *WATI) group_list
  (*PRIOR-NUMER   A0 A1 ... Anph)
  (*PRIOR-DENOM   B0 B1 ... Bnph)
  (*PRIOR-RATE    (*MRC | *BHP(bhp_val) | *WBHP))
  (*PRIOR-CTRL    freq   tcr_min  tcr_max)

*GCIOFF (*GAS)   (group_list | well_list)
        (*WATER)
*GAPPOR group_list *AUTODRILL (*ON|*OFF)
```

Limitations of the Well Management and Group Control Module

The following limitations currently apply to the well management and group control module.

1. The topmost group level (the field) cannot have wells attached to it, but only other groups.
2. Groups to which wells are attached cannot have other groups attached to them.
3. New wells may be attached to a group at any time; however, a well cannot be attached to more than one group at a time. Redefining a well's parent group automatically detaches it from the earlier parent.
4. Group controlled injection fluids are limited to gas (solvent) and water only. Oil and intermediate liquid injection is not supported.
5. The well management module can automatically shut in and reopen well layers when GOR or WCUT exceed a certain limit. When this option is used the layers are sorted according to their depths, in order to open or close layers. If the well is perforated horizontally, the behavior of this option may be unpredictable.
6. Do not switch a normal producer to a cycling producer during a run. Define two separate wells perforated in the same location instead.
7. If cycling injectors exist in a group, at least one producer in the group must be defined as a cycling producer.
8. Group well control cannot be used together with the *STR well operating constraint. If group control is turned on then the *RECYCLE keyword can only be used in conjunction with *GCONI.
9. Cycling injectors and normal solvent injectors cannot both be connected to the same group.
10. For cycling injectors (wells specified with *INCOMP *CYCLING), if an *STG target constraint is used, rather than a recycling fraction, gas will be injected at the specified rate; in this case the cycling producers control the composition but not the rate of the fluid injected.

    In these cases GEM does not check whether the cycling producers actually produce sufficient gas to maintain the specified injection rates. It is therefore possible to inject more gas than is produced by the cycling producers. The following example illustrates the above point:

Example 1:

In this case the injection COMPOSITION for wells 2 and 6 will be determined by the well stream composition and rates of the three cyclic producers, wells 1, 3 and 4. However GEM will not adjust the group target of 1.0E+09 if the three cyclic producers fail to produce sufficient gas to meet the injection target.

```
*GROUP 'GRP-1'    'GRP-2'    *ATTACHTO 'Field'
*GCONP   'GRP-1'
   *GTARGET  *STO  274.0
   *MAX  *GOR  400.0  *SHUTMOW
*GCONI   'GRP-1'
   *GTARGET  *STG  1.0E+09
*WELL 1 'PR-15,10' *ATTACHTO 'GRP-1'
*WELL 2 'INJ-5,11' *ATTACHTO 'GRP-1'
*WELL 3 'PR-03,03' *ATTACHTO 'GRP-1'
*WELL 4 'PR-10,13' *ATTACHTO 'GRP-1'
*WELL 5 'INJH2O  ' *ATTACHTO 'GRP-2'
*WELL 6 'INJ-6'    *ATTACHTO 'GRP-1'
*CYCLPROD  1
   *OPERATE  *MIN  *BHP  20000.0  *CONT
*CYCLPROD  3
   *OPERATE  *MIN BHP    23000.0  *CONT
*CYCLPROD  4
   *OPERATE  *MIN  *BHP  20000.0  *CONT
*INJECTOR   2
*INCOMP     *CYCLING
   *OPERATE *MAX  *STG  6.0E+08
   *OPERATE *MAX  *BHP  1.0d+10
*INJECTOR  6
   *INCOMP  *CYCLING
   *OPERATE *MAX  *STG  6.0E+08
   *OPERATE *MAX  *BHP  1.0d+10
```

If *RECYCLE *GAS 0.90 had followed the *GCONI line instead of *GTARGET *STG, then the gas injection rate could not have exceeded 90% of the group's surface gas production rate.

Recurrent Data from Other Sections

There are a few keywords from the other sections whose parameters may be changed during the course of a simulation run. These keywords must be positioned after a *DATE or *TIME keyword in the Well Data section. At a given time or date the older or existent parameters will be overridden by the new ones.

From the Input/Output Control section, these keywords may be reassigned:

a) *WRST
b) *WPRN
c) *OUTPRN
d) *WSRF and
e) *OUTSRF

From the Reservoir Description section, the refined grid definition keywords and the transmissibility modifiers may appear in the Well Data section. It is critical that grid definition data be grouped together within recurrent data (i.e. not interspersed with well data), failure to do so may cause unpredictable results:

a) *RANGE
b) *REFINE
c) *TRANSI (*TRANLI)
d) *TRANSJ (*TRANLJ)
e) *TRANSK (*TRANLK)
f) *PERMI (cannot be used with *DYNAGRID, refined grid inheritance not done)
g) *PERMJ (cannot be used with *DYNAGRID, refined grid inheritance not done, no *EQUALSI functionality)
h) *PERMK (cannot be used with *DYNAGRID, refined grid inheritance not done, no *EQUALSI functionality)
i) *CTYPE (with limitations, see *CTYPE Keyword)

From the Rock-Fluid Property section, only the rock type array and the end-point arrays may appear in the Well Data section:

a) *RTYPE
b) *SWCON
c) *SWCRIT
d) *SORW
e) *SORMAX
f) *SORG
g) *SOIRW
h) *SLCON

i) *SGCON

j) *SGCRIT

k) KRWIRO

l) *KROCW

m) *KRGCL

n) *KROGCG

o) *PCWMAX

p) *PCGMAX

q) *JFWMAX

r) *JFGMAX

s) *NDARCYCOR

Most of the keywords from the Numerical Methods Control section may be used in this section. The exceptions are *SOLVER *DPLANES and *JACPAR (cannot change between serial and parallel algorithms during a run), and *TWOPTFLUX (must be used throughout the run).

Most of the keywords from Geomechanics section can also be used in this section. Please see chapter on Geomechanics and individual keywords for exceptions.

Well and Recurrent Data Identifier (Required) *RUN

PURPOSE:

*RUN identifies the beginning of all well and recurrent data keywords.

FORMAT:

*RUN

DEFAULTS:

Required keyword. No default.

CONDITIONS:

The WELL AND RECURRENT DATA keyword group follows the NUMERICAL keyword group in the data file. It is the last keyword group in the input data file. This keyword must be the first keyword in the WELL AND RECURRENT DATA keyword group.

Well Change Date (Conditional) *DATE

PURPOSE:

*DATE indicates that the well change will occur at a specified date.

FORMAT:

*DATE yyyy mm dd

DEFINITIONS:

yyyy
An integer specifying the well change year.

mm
An integer specifying the well change month.

dd
A real number specifying the well change day.

DEFAULTS:

Conditional keyword. Defaults for starting date only are:

yyyy:	0000
mm:	1
dd:	1

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Either *DATE or *TIME is required.

EXPLANATION:

To indicate when well changes occur you must use this keyword or the *TIME keyword. If this keyword does not appear immediately following the *RUN keyword, a starting date of 0000 01 01 for the simulation is assumed.

The keyword *DATE is recommended immediately after *RUN.

All well changes are assumed to occur at 0 hr, 0 min. 0 sec. of the specified date. No fractions of a day are allowed.

For example, a well change date of August 19, 1988, is entered as: *DATE 1988 08 19.

All well changes and other recurrent data changes which are specified by keywords located between one *TIME or *DATE keyword and the next *TIME or *DATE keyword are applied at the time of the first *TIME or *DATE keyword.

Well Change Time (Conditional) *TIME

PURPOSE:

*TIME indicates that the well change will occur at a specified time.

FORMAT:

*TIME num

DEFINITIONS:

num
A real number specifying the elapsed time after the simulation starting date at which a well or wells will be changed (days | days).

DEFAULTS:

Conditional keyword. Default for starting time only is: 0.0 . For other times there is no default.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Either *DATE or *TIME is required.

EXPLANATION:

To indicate well changes you must use this keyword or the *DATE keyword. If neither this keyword nor the *DATE keyword appears immediately following the *RUN keyword, the initial date 0000 01 01 and the initial time of 0.0 are assumed.

All well changes and other recurrent data changes which are specified by keywords located between one *TIME or *DATE keyword and the next *TIME or *DATE keyword are applied at the time of the first *TIME or *DATE keyword.

Well Change First Timestep Size (Optional) *DTWELL

PURPOSE:

*DTWELL identifies the first timestep size to be used immediately after the current well change.

FORMAT:

*DTWELL time_step_size

DEFINITIONS:

time_step_size
A real number to specify the timestep size of the first timestep after a non-trivial well change (days | days).

DEFAULTS:

If no *DTWELL keyword is encountered, time_step_size = 1.0E-02 days is assumed. If *DTWELL is included in the WELL AND RECURRENT DATA group, recall that dtmin is set to 1.0E-05 days by default, and that the value time_step_size entered should be at least several times larger than dtmin.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Entering time_step_size less than or equal to zero is equivalent to stopping the simulation with a *STOP line.

EXPLANATION:

This keyword is required to specify the size of the first timestep in the simulation and the timestep immediately after the current well change. This keyword is part of recurrent data, and may be changed during the simulation run.

Note that DTWELL is not applied after well changes used only to generate output at a specific time with no change in well data.

Maximum and Minimum Timestep Size (Optional)

***DTMAX, *DTMIN**

PURPOSE:

*DTMAX identifies the maximum timestep size.

*DTMIN identifies the minimum timestep size.

FORMAT:

```
*DTMAX  max_time_size
*DTMIN  min_time_size
```

DEFINITIONS:

max_time_size
: A real number to specify the maximum timestep size allowed (days | days).

min_time_size
: A real number to specify the minimum timestep size allowed (days | days).

DEFAULTS:

Optional keyword.

Default for DTMAX: 365 days.

Default for DTMIN: 0.01 days.

CONDITIONS:

These keywords may be located in the WELL AND RECURRENT DATA keyword group, and may also be located in the NUMERICAL METHODS CONTROL keyword group. Values in the WELL AND RECURRENT DATA group override those in NUMERICAL METHODS CONTROL from the time they take effect.

EXPLANATION:

If the timestep size calculated by the automatic timestep selector is larger than max_time_size, it is set to max_time_size.

If the automatic timestep selector cuts the timestep size to smaller than min_time_size specified by *DTMIN, the simulation is stopped.

The timestep size is always automatically adjusted so that a timestep will end exactly at the time specified by a *TIME or *DATE keyword in recurrent data.

Once defined by the *DTMAX keyword, the maximum timestep size remains the same until changed by a subsequent *DTMAX keyword.

Once defined by the *DTMIN keyword, the minimum timestep size remains the same until changed by a subsequent *DTMIN keyword.

Setting Grid Blocks to Implicit or Explicit (Optional) *AIMSET

PURPOSE:

*AIMSET assigns implicit or explicit formulations to grid blocks.

ARRAY:

*AIMSET values

DEFINITIONS:

values

A number indicating a given block is explicit or implicit:

0: explicit

1: implicit

2: explicit and stays explicit permanently

3: implicit and stays implicit permanently

DEFAULTS:

Optional keyword. See EXPLANATION for more information.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword sets the implicitness of grid blocks. One need not specify values for all grid blocks. Those blocks not specified will not be changed from their current settings. Initially all grid blocks are set to be explicit with value = 0.

During the course of the simulation, only those grid blocks with value = 0 or 1 may switch values if the adaptive-implicit option is used (set by *AIM). The implicit/explicit pattern for those grid blocks with value = 2 or 3 will not be altered, unless the values of these grid blocks are reset by another *AIMSET keyword at a later date, or in very unusual cases of non-convergence of the Newtonian iterations, in which case the simulator may designate a block as implicit in order to obtain convergence. This latter action cannot be overridden.

All active well blocks are set to be implicit internally, irrespective of their *AIMSET values. Optionally the neighbours or the neighbours plus their neighbours (see diagram in *AIMWELL) can be set to be implicit by using *AIMWELL *WELLN (or *WELLNN). The *AIMWELL overwrites *AIMSET.

For example,

```
*AIMSET *CON 0
```

sets all blocks to explicit (except for the active well blocks). During subsequent simulation runs, blocks may be switched to implicit if an adaptive- implicit formulation is used.

Similarly,

```
*AIMSET *CON 1
```

sets all blocks to implicit. During subsequent simulation runs, blocks may be switched to explicit if an adaptive-implicit formulation and a stability switching criterion is used.

Setting Well Blocks and Neighbours to Implicit (Optional)

***AIMWELL**

PURPOSE:

*AIMWELL is used to set blocks near wells to an implicit formulation.

FORMAT:

```
*AIMWELL      (*WELLN)
              (*WELLNN)
```

DEFINITIONS:

*WELLN

This subkeyword sets neighbours of active well blocks to implicit. A neighbour is a grid block sharing a face with the well block.

*WELLNN

This subkeyword sets the neighbours of active well blocks and the neighbours of the neighbours to implicit.

DEFAULTS:

Optional keyword. If omitted, only active well blocks are implicit. *AIMWELL used without a subkeyword is equivalent to *AIMWELL *WELLN.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

Referring to the diagram below, *AIMWELL *WELLN sets the blocks marked N to implicit. *AIMWELL *WELLNN sets the blocks marked N and NN to implicit. W denotes the well block.

J = 5			NN		
J = 4		NN	N	NN	
J = 3	NN	N	W	N	NN
J = 2		NN	N	NN	
J = 1			NN		
	I = 1	I = 2	I = 3	I = 4	I = 5

In three dimensional systems, the neighbouring grid blocks in the third direction are also set to implicit.

The *AIMWELL keyword overwrites the *AIMSET keyword.

Set Frequency of Initialization of Bottom-Hole Pressure (Optional) *WELLINIT

PURPOSE:

*WELLINIT allows the user to specify that bottomhole pressure values for all, or for specified, wells running on rate, drawdown, or implicitly imposed well head pressure constraints should be reinitialized before every timestep or before every Newtonian iteration. Both particular well initialization frequencies, which are entered for a particular well or set of wells named in a list, and a global initialization frequency, which applies to wells for which no particular frequency has been set, can be set using *WELLINIT.

FORMAT:

1. To set the global well initialization frequency, i.e. to apply to all wells, enter

```
*WELLINIT      (*CHANGE)
               (*TIMESTEP)
               (*ITER)
```

2. To set wells in a user-chosen list to a particular well initialization frequency, enter

```
*WELLINIT  well_list      (*CHANGE)
                          (*TIMESTEP)
                          (*ITER)
```

DEFINITIONS:

CHANGE

Indicates that bottomhole pressures of rate- or implicit well head pressure-constrained wells are to be reinitialized only after significant changes in well operating conditions.

TIMESTEP

Indicates that bottomhole pressures of rate- or implicit well head pressure-constrained wells are to be reinitialized at the beginning of each timestep, after new wellbore head values have been calculated for the coming timestep.

ITER

Indicates that bottomhole pressures of rate- or implicit well head pressure-constrained wells are to be reinitialized before each Newtonian iteration.

well_list

A set of 'well_names' or well_numbers; see below. The presence or absence of the well_list identifies to the simulator whether the global initialization frequency is being set or particular frequencies are being set.

well_names

Any number of well names (in quotes) to specify the wells to which this alteration of initialization frequency applies. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

well_numbers

Any number of integers, or ranges of integers to specify the well numbers to which this alteration of initialization frequency applies.

DEFAULTS:

Optional keyword. If *WELLINIT does not appear in the data set, *WELLINIT *ITER is the default global well initialization frequency in GEM.

The global initialization frequency may be reset by entering *WELLINIT in the first format above without the well list; the global frequency may be overridden for particular wells by using the second format above with the well list.

If *WELLINIT appears alone on a line with no following subkeyword or well list, then *WELLINIT *ITER is set as the global well initialization frequency.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. If a well list is included in the *WELLINIT line, then the *WELLINIT line must follow all of the *WELL lines which define the wells in the list.

EXPLANATION:

*WELLINIT *ITER may facilitate convergence of the Newtonian iterations in difficult cases where the well equations are converging slowly. The *WELLINIT *ITER option requires more CPU time per Newtonian iteration than the other options, but may result in an overall CPU time saving if the number of Newtonian iterations is reduced significantly. *WELLINIT *CHANGE or *TIMESTEP may give a sufficiently accurate initial bottomhole pressure that the Newtonian iterations for the well constraint equation converge rapidly, and in some cases these options will allow convergence where *WELLINIT *ITER will not.

Sometimes only a few wells in a large field require the *WELLINIT *ITER or *WELLINIT *TIMESTEP treatment; in such cases the default treatment may be set to *CHANGE using the first format above and the problem wells may be flagged for special treatment using the second format above, including a well list.

In GEM the global initialization frequency applies to all wells not flagged for special treatment under format (2) EXCEPT that :

a) Wells running on implicitly-imposed WHP constraints use *TIMESTEP unless *ITER is specified as the global frequency; and

b) Wells which use the *FULLY-MIXED crossflow model and have at least one layer backflowing use *ITER regardless of the global frequency setting (although this may be overridden on a well-by-well basis by using format (2) above).

Initialization frequencies specified for a well under the second format above (i.e. by inclusion of the well's name or number in a well list following *WELLINIT) are always honored, regardless of the well's current operating constraint and the current setting of the global initialization frequency.

Example:

If the *TIMESTEP frequency suffices for most wells in the field, but *CHANGE suffices for 'WELL1' and *ITER is necessary for 'WELL2', the following sequence is appropriate:

```
*WELLINIT  *TIMESTEP
*WELLINIT  'WELL1'  *CHANGE
*WELLINIT  'WELL2'  *ITER
```

in the WELL AND RECURRENT DATA section.

Group Identification (Optional) *GROUP

PURPOSE:

*GROUP is used to identify gathering centres, groups and platforms. The information entered with this keyword is used to build a tree structure of groups.

FORMAT:

*GROUP 'child_1' ... 'child_n' *ATTACHTO 'parent'

DEFINITIONS:

'child_1', ‘child_2’ .. , ‘child_n’
: Names of child groups that are attached to the 'parent' group. Each group is identified by a unique name up to 40 characters long and enclosed in single quotes. The *ATTACHTO keyword is not optional and must be present.

*ATTACHTO
: Defines the parent group of all groups named in the list following *GROUP.

'parent'
: Name of the parent group.

DEFAULTS:

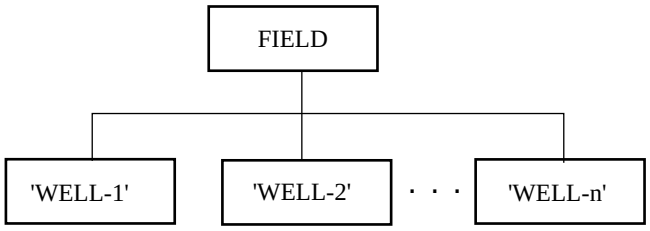
Optional keyword. If no *GROUP line appears in the data, no group structure exists in the simulation and well rates and cumulatives are summed directly into a field cumulative. This is reported as the FIELD cumulative in output, but no group called 'FIELD' actually exists in this case and no group control of production or injection is possible.

When a *GROUP line is encountered in the data a group structure is established, which always consists of at least two groups : a top-level group and the 'Default-Group'.

The top-level group has no default name; its identity is determined by finding the unique group which appears in the list of parent groups but not in the list of child groups. If there is no such group or more than one, an error is generated and simulation terminates.

Wells can only be attached to groups other than the field (top-level) group. Any wells which are not explicitly associated with a parent group are automatically attached to an internally-created group which has the name 'Default-Group' and which has the top-level group as its parent group.

For example if no reference is made to group hierarchy in the data, the following conceptual structure (not truly a group structure) will exist by default:

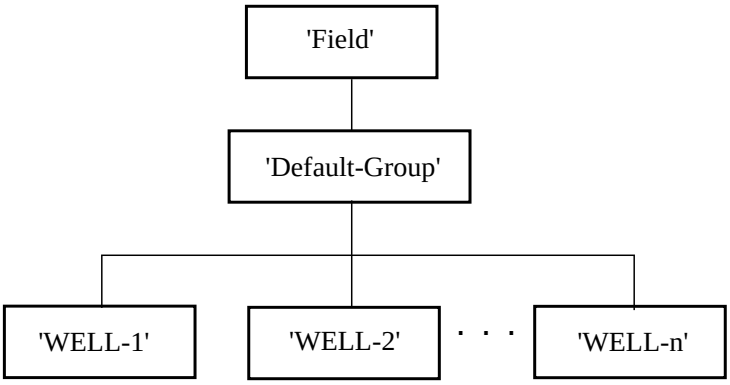


Quotes were purposely omitted around FIELD above to emphasize that no group with the name 'FIELD' actually exists in this case. No group control is possible in this case.

If the single line

```
*GROUP 'Default-Group' *ATTACHTO 'Field'
```

is added to the data, then the following group structure is established:



and group controls could then be imposed upon 'Field'.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. When the simulator encounters a *GROUP keyword, it continues to look for *GROUP keywords until a different keyword is found. It then constructs a set of hierarchical group pointers. When these pointers are constructed, all group controls (e.g. those entered using the *GCONM, *GCONP, or *GCONI keywords) for the pre-existing group hierarchy cease to be in force. Thus if the user intends to add a group to the hierarchy after the beginning of simulation, all group controls must be re-established (if the intention is for them to remain in force) by re-entering them on the appropriate data lines. Otherwise, no group controls will be in effect at the resumption of simulation.

EXPLANATION:

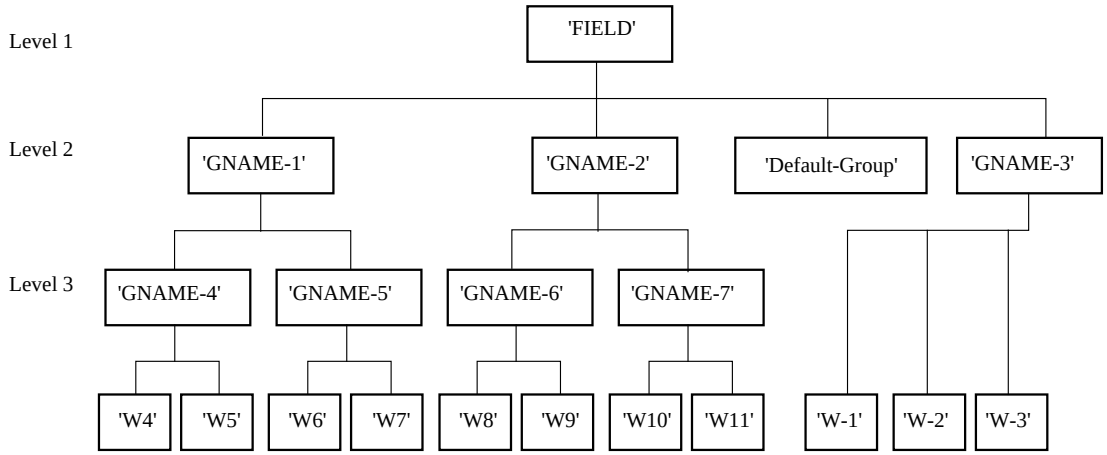
This keyword identifies the group, by name, and assigns it to a parent group. The levels of groups are unlimited in the group hierarchy:

1. Top-level. Only one group is allowed at this level; it has no default name and can be assigned any name not exceeding 40 characters in length by the user. This group cannot have wells attached directly to it. This group represents the whole field. The name of this group is entered after *ATTACHTO in a *GROUP line. The top-level group is identified as the only group whose name appears after *ATTACHTO in at least one *GROUP line but whose name never appears in a list immediately following *GROUP in a *GROUP line.

2. Level 2. These groups have the top-level group as their parent. When a group structure exists, there is always at least one group in this category, with the name 'Default-Group'. 'Default-Group' has connected to it any wells not explicitly attached to a parent group. Level 2 groups can have either wells or groups attached to them, but not a mixture of the two. That is, if a level 2 group is named after the *ATTACHTO subkeyword in a *WELL line, then that group must not appear after the *ATTACHTO subkeyword in any *GROUP line, and vice versa.
3. Level n. These groups have level n–1groups as their parents. Level n groups can have either wells or groups attached to them, but nit a mixture of the two.

Examples of valid and invalid well control trees are given below.

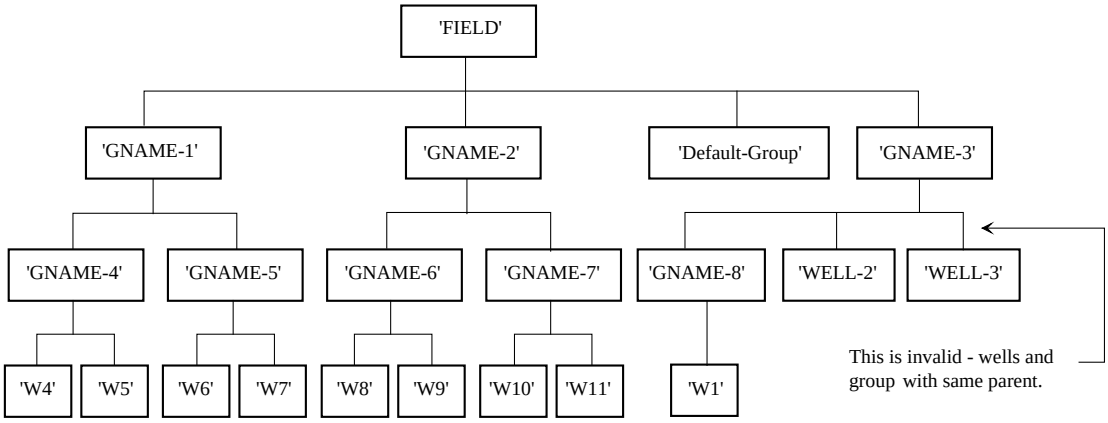
Valid example of a well control hierarchy:



This above example was obtained by using the following keywords:

```
*GROUP     'GNAME-1'   'GNAME-2'   'GNAME-3'   *ATTACHTO    'FIELD'
*GROUP     'GNAME-4'   'GNAME-5'               *ATTACHTO    'GNAME-1'
*GROUP     'GNAME-6'   'GNAME-7'               *ATTACHTO    'GNAME-2'
*WELL      'W-1'                               *ATTACHTO    'GNAME-3'
*WELL      'W-2'                               *ATTACHTO    'GNAME-3'
*WELL      'W-3'                               *ATTACHTO    'GNAME-3'
*WELL      'W4'                                *ATTACHTO    'GNAME-4'
*WELL      'W5'                                *ATTACHTO    'GNAME-4
*WELL      'W6'                                *ATTACHTO    'GNAME-5'
*WELL      'W7'                                *ATTACHTO    'GNAME-5'
*WELL      'W8'                                *ATTACHTO    'GNAME-6'
*WELL      'W9'                                *ATTACHTO    'GNAME-6'
*WELL      'W10'                               *ATTACHTO    'GNAME-7'
*WELL      'W11'                               *ATTACHTO    'GNAME-7'
```

Invalid example of a well control hierarchy:


'FIELD'
'GNAME-1'
'GNAME-2'
'Default-Group'
'GNAME-3'
'GNAME-4'
'GNAME-5'
'GNAME-6'
'GNAME-7'
'GNAME-8'
'WELL-2'
'WELL-3'
'W4'
'W5'
'W6'
'W7'
'W8'
'W9'
'W10'
'W11'
'W1'
This is invalid - wells and group with same parent.

Define Reporting Group (Optional) *REPORTING-GROUP

PURPOSE:

*REPORTING-GROUP allows the user to define a set of wells with differing membership weights which has data reported just as for the hierarchical groups which are the basis for group control (see the manual entry for *GROUP, *GCONP, and related keywords). No group controls can be specified for reporting groups, but there are no restrictions upon well membership in reporting groups. A well may be a member of an arbitrary number of reporting groups, and a well can have any non-negative membership weight in a reporting group.

FORMAT:

*REPORTING-GROUP ‘reporting_group_name’ well_list
weight_list

DEFINITIONS:

reporting_group_name
A character string containing not more than 40 characters. The character strings ‘Default-Field’ and ‘Default-Group’ are not allowed as the name of a reporting group as they are reserved for internal use. Reporting group names must be distinct from group names. If ‘reporting_group_name’ has already been used as the name of a reporting group, then the current instance of *REPORTING-GROUP has the effect of redefining the named reporting group.

well_list
A set of 'well_names' or well_numbers; see below. The reporting group will include all wells in the list.

well_names
Any number of well names (in quotes) to specify the wells in the reporting group. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

well_numbers
Any number of integers, or ranges of integers specifying the numbers of the wells in the reporting group.

weight_list
Non-negative real numbers specifying the membership weights in the reporting group of the wells in well_list. If weight_list contains only a single number, that weight is applied to all of the wells in the list; otherwise the number of entries in weight_list must equal the number of wells in the well_list. The numbers in the weight_list must be real numbers with the decimal point included, in order that the beginning of the weight list can be distinguished from a continuation of a list of well numbers. Repeat counts are allowed, e.g. 6*0.5.

DEFAULTS:

Optional keyword. If *REPORTING-GROUP does not appear in the data set, no reporting groups exist during the simulation. All of the indicated elements in the format (reporting_group_name, well_list, and weight_list) must be present.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. All wells appearing in the well list must already have been defined with *WELL lines. The reporting_group_name must not already have been used (through the *GROUP keyword) as a group name. The weights must be entered as non-negative real numbers with the decimal point explicitly included. The number of weight values must either be one, in which case the single value will be applied to all listed wells, or else must equal exactly the number of wells listed. The well list and the weight list can be spread over several lines; the weight list can begin immediately after (and on the same line as) the end of the well list.

EXPLANATION:

The quantities displayed for the reporting group (e.g. cumulative amounts or rates) are calculated as

$Q(\text{reporting group}) = \sum w(\text{well})Q(\text{well}),$

where the summation is over the wells in the well_list, w(well) is the weight value for the particular well, and Q(well) is the quantity's value for the well. Note that w(well) is the weight value as entered by the user; no automatic normalization is performed upon the weights.

Example:

```
*REPORTING-GROUP 'Producers' 'PROD1' 'PROD2'
1.
```

This establishes a reporting group with the name 'Producers' consisting of the two wells PROD1 and PROD2. The single weight value 1. applies to both wells in the list.

Well Identification (Required) *WELL

PURPOSE:

*WELL is used to identify wells.

FORMAT:

*WELL wnum ‘well_name’ (*VERT ibl jbl) (*FRAC frac)
(*ATTACHTO ‘group_name’)

DEFINITIONS:

wnum

An integer representing the well sequence number. Well sequence numbers must start at 1 and increase in increments of 1 as additional wells are defined. Either the well sequence number or the well name must be specified. If only the well sequence number is specified without the well name, then an internal well name is generated automatically by the program. The program generated name is: 'WELL-wnum'. Here wnum is the well sequence number.

'well_name'

Any alphanumeric character string. (40 characters maximum). Enclose the string in single quotes. Either the well sequence number or the well name must be specified. If only the well name is specified without the well sequence number, then an internal well sequence number is assigned to the well automatically by the program. This sequence number is simply the number of *WELL keywords in the data set up to and including the current *WELL keyword. Both sequence number and name may be entered by the user.

*VERT

This keyword indicates that the well is vertical and all completion layers have the same I and J grid block indices. The completion layers are defined by the *PERFV keyword. (Optional)

ibl

I direction grid block index for the vertical well. (Required with *VERT.) There is no default value.

jbl

J direction grid block index for the vertical well. (Required with *VERT.) There is no default value.

*FRAC *frac*

This optional subkeyword indicates that the rates and indices used internally will be the fraction *frac* of those specified directly in the well data. The allowed range for *frac* is 0 to 1. All rates and indices can be entered for a full well, and they will be multiplied by *frac* for internal use.

*ATTACHTO

Introduces the name of the parent group to which the well is attached. Specified by the name 'group_name'. (Optional)

'group_name'

Name of the group to which the well is connected. If a group name is not specified then the well is connected to an internally-generated group named 'Default-Group' by default. (Optional)

DEFAULTS:

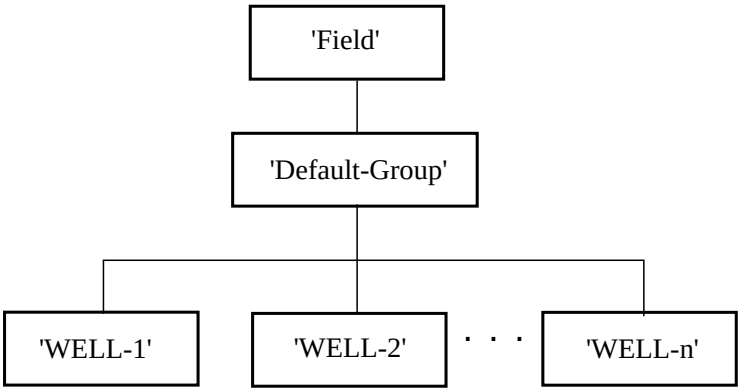
Required keyword. No defaults. Minimum required is:

*WELL 'well_name'
-or –
*WELL wnum

Wells cannot be attached to the top-level group (the group which represents the whole field). For information on how to specify the name of the top-level group, please see the manual page for the *GROUP keyword. If there are no second-level groups defined in a data set then all wells are attached to a group called 'Default-Group' automatically. For example if the line

```
*GROUP 'Default-Group' *ATTACHTO 'Field'
```

appears in the data and a target rate for 'Field' is specified using the *GCONP keyword, the following tree structure will exist by default.



CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword identifies the well number and name, and optionally assigns it to a well group.

'wnum' must be an integer number. If well numbers are used instead of well names, they should start with 1 and increase in sequence for subsequent *WELL keywords.

Example:

1. Both well sequence number and name are specified :

   ```
   *WELL 1 'Producer1'
   ```

2. Only well sequence number is specified. In this example the program will assign the name 'WELL-2' to the well :

   ```
   *WELL 2
   ```

3. Only the well name has been specified. If the above 2 lines are included in a data set with the following line, the program will assign the well number 3 to this well :

   ```
   *WELL 'Prod. 54-233a'
   ```

*FRAC is useful for defining fractions of wells when simulating parts of field patterns, e.g., symmetry elements. See *keywords *VAMOD and *VATYPE in the Reservoir Description section for geometry modification of blocks on the symmetric boundaries. Rates and well indices for a familiar full-well may be entered as data, and *FRAC will specify what fraction of these values to use (e.g., 1/2 on the side, 1/4 in a 90 deg corner, 1/8 in a 45 deg corner). For a well which uses *frac* different from 1, all the reported well performance statistics correspond to the fractional well. *FRAC affects the following quantities:

1. Rates specified by *OPERATE.
2. Well indices entered directly via *PERF or *PERFV (default). The index used will be $wi \cdot frac$.
3. Well index calculated from parameters entered via *GEOMETRY and *PERF (or *PERFV) *GEO. Specify *GEOMETRY parameters for the full well (wfrac = 1); the well fraction from *FRAC will be applied to the resulting full-well index.

For example, consider a 1/8 symmetry element of an inverted nine-spot pattern where the near and far producers have the same physical description and operating conditions. These two wells would have the same well data, except for *frac* = 0.25 for the near producer and *frac* = 0.125 for the far producer.

```
*WELL 1 'Far Prod'   *VERT 9 1 *FRAC 0.125
*WELL 2 'Near Prod'  *VERT 5 5 *FRAC 0.25
*PRODUCER 1 2
*OPERATE MIN BHP 50         ** Start on 50 psi BHP
*OPERATE MAX STW 12         ** Max water rate 12 b/d

*GEOMETRY .4 .249 1 0       ** Full well
*PERFV *GEO 1 2 ** k
                  1:4
```

Well Type Definition (Required) *PRODUCER, *INJECTOR, *CYCLPROD

PURPOSE:

Specify a well's type (producer, injector, or recycling producer that all or part of its separator gas will be reinjected).

FORMAT:

*PRODUCER / *INJECTOR / *CYCLPROD *well_list*

DEFINITIONS:

well_list

A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

DEFAULTS:

Required keywords.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. The well type specification must follow the well definition and must immediately precede the *PWELLBORE or *IWELLBORE lines if present. Perforation data may precede the well type specification.

EXPLANATION:

These required keywords are used for well definition. A well may be defined at one time using *WELL, have its completions specified at a later time with *PERF, and at a still later time have its type defined with *PRODUCER or *INJECTOR and go into operation at this time. If wellbore hydraulics calculation is desired or if the well is going to be on a WHP pressure constraint, then the *PWELLBORE keyword for a producer or *IWELLBORE keyword for injector must immediately follow the well type definition. For injectors the *INCOMP keyword must appear before the operating and monitored constraints are stipulated using *OPERATE or *MONITOR.

The wellnum may be a single integer, a series of integers delimited by spaces or commas, or a range of integers defined in the form: imin:imax.

Example:

```
*WELL 1 'Producer'
*WELL 2
*WELL 3 '54-233a'

*PRODUCER 1:2    ** define wells 1 and 2 as
                 ** normal producers
```

Shut in Wells above Formation (Optional) *MODELSHUT, *EQUILIBRATE

PURPOSE:

*MODELSHUT indicates that the listed wells identified by well_list will be closed above formation with crossflow modelling of the constituent (open) layers, whenever the wells are shut in due to a well operation.

FORMAT:

*MODELSHUT	(*well_list*)	(*ON | *OFF)
*EQUILIBRATE	*epsmds*	

DEFINITIONS:

well_list

A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

epsmds

Fluid equilibration criterion *epsmds* (m3/day | bbl/day) must be a non-negative, real number.

DEFAULTS:

Optional keyword. Default is *MODELSHUT *OFF for all wells. Use of *MODELSHUT with no occurrence of *ON or *OFF has the effect of setting *MODELSHUT *ON for the wells listed.

If keyword *EQUILIBRATE is absent then *epsmds* is assumed to be 0.1.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group after wells have been defined with the keyword *WELL.

EXPLANATION:

Various actions can cause a well to become shut in: *SHUTIN, effective *WLISTSHUT, *ALTER or *OPERATE with a zero rate value, or violation of operation constraints. The default method for shutting in a well (“fully shut”) is to close immediately all of its active layers. However, keyword *MODELSHUT allows fluid to equilibrate for some time in the wellbore before all the layers are closed (“model-shut”).

Keyword *MODELSHUT itself does not cause a well to shut in. Instead, it gives a well the permission to go through an equilibration stage when the well is shut in for the reasons listed above. During this equilibration stage the well is operated on a total rate constraint of zero, which allows cross-flow in the wellbore to redistribute fluid between layers in the reservoir (*FULLY-MIXED cross flow model). This occurs even if a different *XFLOW-MODEL

option was specified for that well. In the text output file, a well operating in an equilibration stage is identified by the well status MSHT (as opposed to SHUT).

As the equilibration stage continues, pressure drops and layer flow rates may gradually decrease. Depending on the reservoir, well completion and operation conditions, such a fluid equilibration process may last months or even much longer in cases with persistent back flow. When each layer's rate falls below the criterion defined by *EQUILIBRATE the well leaves the equilibration stage to become fully shut, that is, all the well's active layers are closed immediately. A message is issued when the equilibration criterion has been satisfied.

Once a well has switched from the equilibration stage to fully shut, fluid equilibration is no longer checked unless it is activated by another *MODELSHUT action. As a shut in well, a *MODELSHUT well will not be involved in any group target apportionment. The keyword *AUTODRILL will not put an autodrillable well in an equilibration stage unless such a well has been drilled previously.

*MODELSHUT is not effective for a single-perforation well, which will equilibrate instantly.

Various actions can cause a shut-in well to open: *OPEN, effective *WLISTOPEN, *ALTER or *OPERATE with a non-zero rate value. If an open action is encountered while the well is in an equilibration stage, the well is immediately put back on its usual (most restrictive) operating constraint with its assigned or defaulted backflow model. However, the well retains its *MODELSHUT status, that is, its ability to use an equilibration stage upon the next shut-in action. If you want to remove this ability, use *MODELSHUT *OFF explicitly.

Example:

```
** Wells #1 and #2 will experience fluid equilibration
** when they encounter a shut-in action.  Well #3 will
** be fully shut in immediately.
*TIME ...
   *MODELSHUT 1:2 *ON
   *EQUILIBRATE 1d-2   ** Larger equilibration criterion
...
*TIME ...
   *SHUTIN 1:3
```

Well Status Definition (Optional) *SHUTIN, *OPEN, *AUTODRILL

PURPOSE:

*SHUTIN indicates that the well identified by wellnum is shut in.

*OPEN indicates that a previously shut-in well is reopened. The well type must be defined before *OPEN can be specified for the well.

*AUTODRILL indicates that the well is currently not drilled (is shut in) but will be drilled (opened) automatically to meet the target rates of its group, if the group has *AUTODRILL *ON specified under the *GAPPOR keyword. Please see the manual entries for the *DRILLQ and *GAPPOR keywords for more information. The well type must be defined before *AUTODRILL can be specified for the well.

FORMAT:

*SHUTIN / *OPEN / *AUTODRILL well_list

DEFINITIONS:

well_list

A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names; please see the explanation below.

DEFAULTS:

Optional keywords. A well which is never listed after *SHUTIN, *OPEN, or *AUTODRILL has status *SHUTIN before the well type has been defined and is automatically given *OPEN status when the type is defined.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. *OPEN and *AUTODRILL must be located after the well type (*PRODUCER, CYCLPROD or *INJECTOR) as well as the operating constraints have been defined.

EXPLANATION:

These required keywords are used for well status. If a status of *OPEN or *AUTODRILL is specified, the well must be defined completely, i.e. all its perforations and operating constraints must be specified.

The well number may be a single integer, a series of integers delimited by spaces or commas, or a range of integers defined in the form imin:imax.

If well names are used in the list, a limited wild-card facility is available to allow the user to specify the desired list of wells without having to name each well separately. The description of this facility is as follows:

1. There are two wild-card characters, '?' and '*'.

2. Any number of ?'s can appear in a well name character string; each ? is counted as matching any character in the same position in a well name, including embedded blanks but NOT final blanks. For example, 'WELL??' matches 'WELL 1' but 'WELL?' does not match 'WELL'.
3. A single * can appear as the last non-blank character in a well name character string; when * is present only the characters preceding * are checked for lack of match. For example, the character string '*' matches all wells, and 'WELL*' matches 'WELL'.
4. ? and * can appear in the same string; for example 'WELL????PROD*' would match 'WELL_NW_PROD_15' and 'WELL_SE_PROD_2'
5. A list of the wells matched when wild-card characters are used is printed in the output file so that the user may check the list generated.

Example:

```
*WELL 1 'Injector' *ATTACHTO 'Group-1'

*INJECTOR 1  ** Define the type of well 1.
*INCOMP *SOLVENT composition ** Injecting gas.
** Operating on a rate specification.
*OPERATE *MAX *STG 1.0E+8
** If backflow occurs, stop the simulation
*MONITOR *BACKFLOW *STOP
*PERF 1
** if  jf  kf  WI
   1   1   3   1.0E+5
*WELL 2 'Producer' *ATTACHTO 'Group-1'

*PRODUCER 2 ** Define the type of well 2.
** Operate a rate constraint first.
*OPERATE *MAX *STO 20000.0
** If the BHP falls below 1000 psi, switch to a
** minimum pressure constraint.
*OPERATE *MIN *BHP 1000.0
** When the oil rate falls below 1000 bbl/D
** stop the simulation.
*MONITOR *MIN *STO 1000.0  *STOP
*MONITOR *BACKFLOW         *STOP
*MONITOR *GOR      20000.0 *STOP
*PERF 2
** if  jf  kf  WI
   10  10  1   1.0E+5
*SHUTIN 1
*AUTODRILL 'Prod*'

** At this point you may open any of these wells.

*TIME 900.0
** Open the injector
*OPEN 1
```

Shut and Reopen a List of Wells (Optional)

***WLISTSHUT, *WLISTOPEN**

PURPOSE:

*WLISTSHUT and *WLISTOPEN provide a means to shut temporarily a large list of wells and later re-open them without disturbing the pattern of shut and autodrillable wells that existed when the *WLISTSHUT keyword was read.

*WLISTSHUT indicates that the listed wells identified by *well_list* are to be shut in. The difference between this keyword and the keyword *SHUTIN is that *WLISTSHUT has a dynamic well list that a paired keyword *WLISTOPEN can operate upon to reopen the listed wells without disturbing the pattern of shut and open wells that existed when the *WLISTSHUT keyword was applied (see explanation).

*WLISTOPEN reopens wells or restores auto-drillable status to wells in the common set of the following well list (optional) and the effective well list from previous *WLISTSHUT. All of the *WLISTSHUT list will be opened if no well list follows *WLISTOPEN.

FORMAT:

*WLISTSHUT *well_list*
*WLISTOPEN (*well_list*)

DEFINITIONS:

well_list

A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. The *well_list* is required for the *WLISTSHUT keyword and optional for *WLISTOPEN.

DEFAULTS:

Optional keywords. No defaults.

CONDITIONS:

These keywords must be located in, but may appear anywhere in, the WELL AND RECURRENT DATA keyword group. A well must have been defined with the *WELL keyword before it can appear in a *WLISTSHUT list. A well must have had its type defined with the *PRODUCER or *INJECTOR keyword before it can appear in a WLISTOPEN list.

EXPLANATION:

*WLISTSHUT is used to pause the operations of a list of wells within the period ended by the issuance of a paired keyword *WLISTOPEN. All wells on the list will be shut-in. Auto-drillable status, if applicable, will be removed temporarily. Wells that have already been shut-in or whose operational types have not yet been defined prior to the entry of the keyword are automatically excluded from the list. Entering a second *WLISTSHUT list before *WLISTOPEN appends the second list to the original one. If there is an explicit action that could potentially modify the status for a particular well during this period (i.e. *OPEN,

*SHUTIN, *AUTODRILL, *PRODUCER, *INJECTOR, *CYCLPROD, *ALTER, *TARGET, *ALTERCP), the change is made and that well is removed from the well list.

Wells being reopened or having their auto-drillable status re-acquired by *WLISTOPEN are taken out of the remaining well list. The entire well list entered with *WLISTSHUT will be processed if there is no well_list following the keyword *WLISTOPEN.

Example:

```
*TIME 0.
*PRODUCER 1:3
*OPERATE *MAX *STO 500.
*INJECTOR 4:5
*INCOMP *WATER
     *OPERATE *MAX *STW 150.

*SHUTIN 1
*AUTODRILL 3

*TIME 100.
*WLISTSHUT 1:5    **Shutting a list of wells. Since Well 1
                    has already been shut before entry of the
                    keyword, it will be automatically excluded
                    from the list. Well 3 is temporarily not
                    allowed to be auto-drilled.
*TIME 200.
*ALTER 2
      400.        **Well 2 is explicitly altered and is thus
                    removed from the *WLISTSHUT list.

*TIME 300.
*WLISTOPEN        **Reopening wells on the remaining list issued
                    by *WLISTSHUT, i.e. Well 4 and 5. Meanwhile,
                    Well 3 re-acquires the auto-drillable candidacy.
```

Tubing Data for Injectors (Conditional) *IWELLBORE

PURPOSE:

*IWELLBORE specifies that wellbore pressure drops will be computed for this injector, in order that a well head pressure constraint may be enforced, or if information about the well head pressure is wanted. The wellbore pressure drops are computed using the correlation due to Aziz, Govier, and Fogarasi (JCPT, July-Sept. 1972, pp. 38-48). Tubing data are required for the pressure drop computation. The pressure drop can also be interpolated from a designated hydraulics table.

FORMAT:

*IWELLBORE (*MODEL)
wdepth wlength rel_rough whtemp bhtemp wradius
- or -
*IWELLBORE *TABLE
wdepth table_number

DEFINITIONS:

wdepth
A real number specifying the depth of an injection well (depth difference between the surface and the reference completion of the well) (m | ft). When tabular interpolation is used, wdepth is used to scale the pressure drop linearly relative to the depth entered with the designated table.

wlength
A real number specifying the length of an injection well (distance from the well head to the reference completion) (m | ft).

rel_rough
A real number specifying the relative well roughness. Dimensionless.

whtemp
A real number specifying the well head temperature (deg. C | deg. F).

bhtemp
A real number specifying the reservoir temperature (deg. C | deg. F).

wradius
A real number specifying the well tubing radius (m | ft).

*MODEL
Indicates that the tubing pressure drop should be computed using a correlation.

*TABLE
Indicates that the tubing pressure drop should be interpolated from a hydraulics table (input using the *ITUBE1 keyword).

table_number
*ITUBE1 table number.

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group immediately after *INJECTOR if the wellbore option is used.

When the first format above is used (with neither *MODEL nor *TABLE entered) the model is used with the parameters supplied on the next line. *MODEL has exactly the same effect as omitting the (*MODEL|*TABLE) subkeyword.

Required with *INJECTOR when *WHP is an operating constraint or when the wellbore model is desired.

When *IWELLBORE is not encountered, no wellbore pressure drop calculations are performed for the wells entered under the preceding *INJECTOR keyword. Pressure differences between layers are computed using the Aziz et al correlation with the default well radius.

EXPLANATION:

This keyword is required if one of the operating constraints is *WHP or when the wellbore model is desired for injectors.

Well depth is the vertical distance from the well head to the centre of the grid block in which the well's reference layer is perforated.

The length is the length of the well from the well head to the centre of the grid block in which the well's reference layer is perforated.

Relative well roughness is defined as:

$$\frac{\text{maximum roughness size}}{\text{tubing diameter}}$$

The "roughness size" refers to the length scale of irregularities in the tubing wall.

The acceptable range of values for well depth is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

The acceptable range of values for well length is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

The acceptable range of values for the relative roughness is:

	SI m	Field ft
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

The acceptable range of values for wellbore temperature is:

	SI ^{o}C	Field ^{o}F
min	-1.0E+4	-18032.0
max	1.0E+4	18032.0

The acceptable range of values for reservoir temperature is:

	SI ^{o}C	Field ^{o}F
min	-1.0E+4	-18032.0
max	1.0E+4	18032.0

The acceptable range of values for well radius is:

	SI m	Field ft
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

Tubing Data for Producers (Conditional) *PWELLBORE

PURPOSE:

*PWELLBORE specifies that wellbore pressure drop calculations will be done for this producer, and specifies the method by which they will be done.

FORMAT:

*PWELLBORE (*TABLE) *wdepth table_number*
-or-
*PWELLBORE *MODEL
wdepth wlength rel_rough whtemp bhtemp wradius

DEFINITIONS:

*TABLE
Specifies that the tubing pressure drop from the center of the grid block in which the well's reference layer is perforated to the surface is to be computed using a table entered by the user (see the entry for the *PTUBE1 keyword in this manual).

wdepth
A real number specifying the depth (m | ft) of the well (difference in depth between the surface and the reference well completion).

table_number
Integer index that specifies which of the tables entered under *PTUBE1 is to be used for the current well.

*MODEL
When this subkeyword is specified, the wellbore pressure drops are computed using the correlation due to Aziz, Govier, and Fogarasi (JCPT, July-Sept. 1972, pp. 38-48).

wlength
A real number specifying the length of an producing well (distance between well head and the reference completion of the well) (m | ft).

rel_rough
A real number specifying the relative well roughness. Dimensionless.

whtemp
A real number specifying the well head temperature (deg. C | deg. F).

bhtemp
A real number specifying the reservoir temperature (deg. C | deg. F).

wradius
A real number specifying the tubing radius (m | ft).

DEFAULTS:

Conditional keyword. No defaults. If both *TABLE and *MODEL are missing then the default is *TABLE.

When *PWELLBORE is not encountered, no wellbore pressure drop calculations are performed for the wells entered under the preceding *PRODUCER or *CYCLPROD keywords. Pressure differences between layers are computed using the Aziz et al correlation with the default well radius.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group immediately following *PRODUCER if the wellbore model is used.

*PWELLBORE is required when the operating constraint *WHP is used or when the wellbore model is desired.

EXPLANATION:

This keyword is required if one of the operating constraints is *WHP or the wellbore model is desired, for producers only. Injectors use the *IWELLBORE keyword to enter the well depth.

When *MODEL is entered, the Aziz et al correlation is used both to compute the surface-to-reference completion pressure drop and to compute the pressure differences between well layers within the reservoir.

When *TABLE is entered, the indicated user-supplied table is used to compute the pressure drop between the surface and the reference perforation of the well. The Aziz at al correlation with the default well radius (3 inches or whatever was entered under *GEOMETRY) is used to compute the wellbore pressure differences between layers in the reservoir.

Well depth is the vertical distance from the well head to the centre of the grid block in which the well's reference layer is completed.

The length is the length of the well from the well head to the centre of the grid block in which the well's reference layer is perforated.

Relative well roughness is defined as:

$$\frac{\text{maximum roughness size}}{\text{tubing diameter}}$$

Example 1:

```
** In the following example *MODEL is used to calculate
** Wellbore hydraulic pressure loss for a gas producer.

*DATE 1988 01 01
*DTWELL 0.003
*DTMIN 1.0E-06

*WELL 1 'Water Injector'
*INJECTOR 1
*INCOMP *WATER
*OPERATE *MAX *BHP 15000.
```

```
** Well geometry for the producer.
**             rad    geofac  wfrac  skin
*GEOMETRY *K  0.0572   0.34    1.0   0.0
*PERF *GEO 1
** if   jf   kf   ff
   4    4    4:4  1.0

*WELL 2 'Producer'
*PRODUCER 2
*PWELLBORE *MODEL
    ** wdepth  wlen   rough   whtemp  bhtemp  wrad
        8400.  8400.  0.0001   60.0   150.0   0.25
*OPERATE *MIN *WHP 2000.

** Well geometry for the producer.
**             rad     geofac  wfrac  skin
*GEOMETRY *K  0.0572    0.34    1.0   0.0

*PERF *GEO 2
** if   jf   kf   ff
   2    2    1:4  1.0

*TIME 100.0
*STOP
```

Example 2:

```
** In the following example *TABLE is used to calculate
** Wellbore hydraulic pressure loss for a producer.
** The units being used are *FIELD.

*DATE 1988 01 01
*DTWELL 0.003
*DTMIN 1.0E-06
*WELL 1 'Water Injector'
*INJECTOR 1
*INCOMP *WATER
*OPERATE *MAX *BHP 3138.
** Well geometry for the producer.
**             rad    geofac  wfrac  skin
*GEOMETRY *K  0.0572   0.34    1.0   0.0
*PERF *GEO 1
** if   jf   kf    ff
   4    4    4:4   1.0
*WELL 2 'Producer'
*PRODUCER 2
** For producer use hydraulic table # 1.
*PWELLBORE *TABLE 8400.0 1
*OPERATE *MIN *BHP 2098.0
** Well geometry for the producer.
**             rad    geofac  wfrac  skin
*GEOMETRY *K  0.0572   0.34    1.0   0.0
*PERF *GEO 2
** if   jf   kf    ff
   2    2    1:4   1.0
** Define the pressure loss table
```

```
*PTUBE1 1
*DEPTH 5000.0
*GOR  **gfr(1) gfr(2)
        500.0  1000.0
*OIL  **flo(1) flo(2) flo(3) flo(4)
        0.0    4000.0 8000.0 16000.0
*WCUT **wfr(1) wfr(2)
        0.00   0.50
*ADD  **add(1)
        0.0
*WHP  **whp(1) whp(2) whp(3)
        200.0  900.0  1600.0
*BHP
**iflo igfr iwfr iadd  bhp(1)  bhp(2)  bhp(3)
   1    1    1    1    2176.1  2873.7  3573.7
   1    2    1    1    1997.7  2670.9  3370.9
   1    1    2    1    2646.7  3344.7  4044.7
   1    2    2    1    2447.7  3124.7  3824.7
   2    1    1    1    2135.5  2876.6  3576.6
   2    2    1    1    1682.7  2674.6  3374.6
   2    1    2    1    2618.0  3351.2  4051.2
   2    2    2    1    2189.0  3132.3  3832.3
   3    1    1    1    2133.6  2884.2  3584.2
   3    2    1    1    1463.1  2684.5  3384.5
   3    1    2    1    2630.9  3368.4  4068.4
   3    2    2    1    2022.0  3152.8  3852.8
   4    1    1    1    2160.1  2912.5  3612.5
   4    2    1    1    1425.7  2721.3  3421.3
   4    1    2    1    2696.4  3433.4  4133.4
   4    2    2    1    2080.0  3231.0  3931.0
*TIME 100.0
*STOP
```

The acceptable range of values for well depth is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

The acceptable range of values for well length is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

The acceptable range of values for the relative roughness is:

	SI	**Field**
	m	**ft**
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

The acceptable range of values for wellbore temperature is:

	SI °C	Field °F
min	-1.0E+4	-18032.0
max	1.0E+4	18032.0

The acceptable range of values for reservoir temperature is:

	SI °C	Field °F
min	-1.0E+4	-18032.0
max	1.0E+4	18032.0

The acceptable range of values for well radius is:

	SI m	Field ft
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

Wellbore Temperature (Optional) *TWELLBORE

PURPOSE:

*TWELLBORE specifies that wellbore temperature calculations will be done for a producer with the assigned method.

FORMAT:

*TWELLBORE *TABLE *wdepth table_number*

DEFINITIONS:

*TABLE
: Specifies that the well-head temperature for a producer is to be computed using a table entered by the user (see the entry for keyword *PTUBE1 with *WHT as the table body in this manual). This is currently the only method for *TWELLBORE.

wdepth
: A real number specifying the depth (m | ft) of the well (difference in depth between the surface and the reference well completion).

table_number
: Integer index that specifies which of the tables entered under *PTUBE1 is to be used for the current well.

DEFAULTS:

Optional keyword. Well-head temperature will be calculated according to the user-provided wellbore temperature table if *TWELLBORE *TABLE is encountered.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group following *PRODUCER.

EXPLANATION:

When *TABLE is entered, the indicated user-supplied temperature table (*PTUBE1 *WHT) is used to compute the well-head temperature. Since the temperature calculation involves the interpolation among the tubing-head pressures, a means specified for computing tubing-head pressure must be available (see keyword *PWELLBORE).

Example:

```
**Table 1 from *PTUBE1 must have *BHP as the table body
**Table 2 from *PTUBE1 must have *WHT as the table body
*DATE 1988 01 01
*WELL 1 'PROD-11'
*PRODUCER 1
*PWELLBORE *TABLE 8400.0 1  **use table 1 for WHP
*TWELLBORE *TABLE 8400.0 2  **use table 2 for WHT
*OPERATE *MAX STO 13.0
*OPERATE *MIN *WHP 2600.0
```

Set Application Mode of Correlation for Well Head Pressure (Optional) *WHPCRTB

PURPOSE:

*WHPCRTB allows the user to invoke a time-saving option in the application of a correlation to compute the WHP. In the integration of pressure up the wellbore, the derivatives dP/dz are usually computed by performing an equation of state flash and applying the correlation each time the derivative value is required. An alternative is to compute a table of dP/dz as a function of pressure and temperature before beginning the integration, and to compute values of dP/dz needed in the integration by interpolation in this table. This latter method can potentially shorten the computation significantly, especially if the simulation has many wells on implicit WHP control. The use of the correlation for computing WHP is fixed with the *PWELLBORE *MODEL or *IWELLBORE *MODEL keywords. The tabular method invoked with *WHPCRTB differs from the use of standard tubing head pressure tables (entered using the *PTUBE1 or *ITUBE1 keyword), in that these dP/dz tables are recomputed before each pressure integration up the wellbore. At each table calculation the true wellbore fluid composition and the true temperature and pressure range in the wellbore are used. The tables entered with *PTUBE1 or *ITUBE1, on the other hand, are used for many different WHP calculations.

FORMAT:

*WHPCRTB *well_list* (*CORR | *CORR-TAB (*nptab* (*nttab*)))

DEFINITIONS:

CORR

Indicates that the correlation should be applied in the usual way, with an equation-or-state flash and an application of the correlation each time a pressure derivative is needed. This is the default.

CORR-TAB

Indicates that a table giving dP/dz as a function of pressure and temperature should be constructed before each integration of pressure up the wellbore.

well_list

A list of 'well_names' or well_numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

nptab

Number of pressure values in the table. Default is 6.

nttab

Number of temperature values in the table. Default is 3.

DEFAULTS:

Optional keyword. If *WHPCRTB does not appear in the data set, *CORR is used for all wells. *CORR-TAB can be specified for listed wells only by entering a *WHPCRTB line. The user can revert to the *CORR method for a given well by listing it and specifying *CORR.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. All wells appearing in the well list must already have been defined with *WELL lines. The specification of *CORR or *CORR-TAB will have an effect only if *PWELLBORE *MODEL has been specified for the well. Injectors always use the correlation, but *IWELLBORE must have been entered for the well in order for the WHP calculations to be done. It is not an error to include in the well list following *WHPCRTB a well for which *PWELLBORE *MODEL is not currently specified; if the well is later respecified to use *PWELLBORE *MODEL, the option specified under *WHPCRTB would take effect.

EXPLANATION:

When many wells are on implicit WHP control, the action of computing the WHP is performed many times for each well. When *PTUBE1 tables are used, each WHP calculation is completed quickly. However, in compositional simulations the changes in composition which may take place in the wellbore over time and the potential sensitivity of the pressure drop to wellbore pressure and temperature may render the fully tabular calculation inaccurate. The *CORR-TAB method allows the realization of essentially the full accuracy of the correlation calculation at a small fraction of the computational expense. For example if the correlation table has 6 pressure and 3 temperature points, the calculation of dP/dz is performed 18 times. The integration of pressure up the wellbore often requires 50 or more integration steps, each requiring several evaluations of dP/dz.

Example:

```
*WHPCRTB  'WELL1' 'WELL2' *CORR-TAB 6 4
```

Directs that a 6 X 4 pressure-temperature table be used for WHP calculations by correlation for wells WELL1 and WELL2.

Set Number of Points for WHP Root Search (Optional)

***NWHYFS**

PURPOSE:

*NWHYFS allows the user to set the number of points used to determine whether a value of the BHP exists which yields the target well head pressure (WHP). This number defaults to 9 and it is not anticipated that the user would normally alter this value. However in certain cases in which a well operating on an implicit WHP constraint suddenly shuts for no apparent reason, invoking NWHYFS for the well with a value exceeding 9 (11 for example) may allow the well to continue. Larger values of this integer may cause run times to increase.

FORMAT:

*NWHYFS *well_list*
integer_list

DEFINITIONS:

well_list
A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

integer_list
A list of integers containing either a single integer or exactly the same number of entries as *well_list* contains. The values must be odd, must be at least 5, and must not exceed 17. The default is 9. If a single integer is entered in the list, it is applied to all wells in *well_list*.

DEFAULTS:

Optional keyword. If *NWHYFS does not appear in the data set, the value 9 is used for all wells.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. All wells appearing in the well list must already have been defined with *WELL lines. The wells in the list may be producers or injectors, but it is anticipated that user specification of this parameter will be warranted only for producers and only in special circumstances. The integers listed must be odd and between 5 and 17 inclusive. The parameter will have an effect only if at least one of the specified wells is operating on a WHP constraint.

EXPLANATION:

If too few points are used in the WHP root search, the simulator may erroneously conclude that no root exists and that the well is unable to operate on the WHP constraint. Specification of a larger number of points may remedy this. Only extremely rarely is the default value of 9 points insufficient. On the other hand if there is no problem with roots not being located,

execution time may be shortened by decreasing the value, say, to 7. Again, only extremely rarely would such a decrease shorten the run time significantly.

Example:

```
*NWHYFS  'WELL1' 'WELL2'
 11 7
```

Directs that 11 points be used for WHP root searches for WELL1 and that 7 points be used for WELL2.

Composition of Injected Fluid (Conditional) *INCOMP

PURPOSE:

*INCOMP specifies the compositions of injected fluid.

ARRAY:

```
*INCOMP    (*WATER)
           (*SOLVENT  values)
           (*CYCLING)
           (*AQUEOUS  aq_values)
```

DEFINITIONS:

*WATER
This subkeyword identifies a water injector.

*SOLVENT
This subkeyword identifies a solvent injector.

values
Mole fractions of components of the injected fluid in the order defined in the component properties section.

Note that if the pseudo-component option is used, values refer to the composition in terms of the pseudo-components.

*CYCLING
This subkeyword specifies that the injection fluid be obtained from the cycling producers.

*AQUEOUS
This subkeyword identifies water injection with soluble components.

aq_values
Molalities (in g mols/kg of water) of components dissolved in the aqueous injection stream. Values for the nc EOS components must be given first, followed by the nc_aq aqueous components. The order of the components is as specified with the COMPNAME and COMPNAME-AQUEOUS keywords.

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. *INCOMP must be located after the *IWELLBORE (optional) keyword but before the *OPERATE or *MONITOR keywords. This keyword is required for all injection wells.

EXPLANATION:

The keyword *INCOMP identifies the composition of injected fluid. The injectors referred to are defined by the preceding *INJECTOR keywords.

FORMAT:

*OPERATE (*MAX or *MIN) type *value* (action)
-or-
*OPERATE (*PENALTY) type (mode) (*v*1) (action)
-or-
*OPERATE (*WCUTBACK) type mode (*v*1 (*v*2 (*v*3))) (action)

where type for *MAX or *MIN is one of:

*STO *value
*STG
*STW
*STL
*STI
*WTG
*BHP
*WHP (*INITIALIZE | *IMPLICIT)
*BHO
*BHG
*BHW
*BHL
*BHF
*DWN, DWA, DWB
*STR

type for *PENALTY or *WCUTBACK is one of:

*GOR
*GLR
*WOR
*WGR
*PRS

mode for *PENALTY is one of:

*NODAMP
*DAMP
*PHASE2

mode (control phase) for *WCUTBACK is one of:

*STO
*STG
*STW
*STI
*WTG
*STL
*BHF

action is one of:

*STOP
*CONT (*REPEAT)
*SHUTIN (*REPEAT)

DEFINITIONS:

*MAX

This subkeyword specifies that the specified value is a maximum constraint. This is the default when neither *MAX nor *MIN is present.

*MIN

This subkeyword specifies that the specified value is a minimum constraint. This subkeyword can be used only for the *WHP and *BHP constraints for producers.

*PENALTY

This subkeyword specifies that the well is to be penalized for overproduction resulting in a high GOR, GLR, WOR or WGR based on government regulations. *PENALTY cannot be the only operating constraint. A maximum oil rate constraint MUST be specified in order to use GOR or WOR penalization. A maximum gas rate constraint MUST be specified in order to use WGR penalization. A maximum liquid rate constraint MUST be specified in order to use GLR penalization.

*WCUTBACK

This subkeyword specifies that the well is to be running on a reduced rate target of the specified control phase whenever it violates the base value ($v1$) of GOR, GLR, WOR, WGR or PRS constraint. The control phase (mode) has to be chosen from one of *STO, *STG, *STW, *STI, *STL, *WTG or *BHF for producers, but should not be entered for injectors. The target rate of the control phase is calculated by multiplying its current rate with the cutback factor ($0<v2<1$). The cutback reversing process (rate increasing) is activated whenever the well violates the reversal limit ($v3$) and the well is currently under the corresponding well cutback constraint. It is not necessary to specify the rate constraint explicitly for the control phase.

*STO

This subkeyword identifies a surface oil rate (m^3/day | Stb/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.

*STG

This subkeyword identifies a surface gas rate (m^3/day | scf/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.

*STW

This subkeyword identifies a surface water rate (m^3/day | Stb/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.

*STL

This subkeyword identifies a surface liquid rate (oil + water + intermediate liquid) (m^3/day | Stb/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. *STL is applicable to producers only.

*STI

This subkeyword identifies a surface intermediate liquid rate (m^3/day | Stb/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. *STI is applicable to producers only. For more information on the intermediate liquid stream please see the description of the *WELSEP keyword in this manual.

*WTG

This subkeyword identifies a surface wet gas rate (m^3/day | SCF/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. *WTG is applicable to producers only. For more information on the wet gas stream please see the description of the *WELSEP keyword in this manual.

*BHO

This subkeyword identifies a reservoir oil rate (m^3/day | bbl/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. This keyword is applicable to producers only.

*BHL

This subkeyword identifies a reservoir liquid rate (oil + water) (m^3/day | bbl/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. This keyword is applicable to producers only.

*BHF

This subkeyword identifies a total reservoir fluid rate (oil + water + gas) (m^3/day | bbl/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.

*BHW

This subkeyword identifies a reservoir water rate (m^3/day | bbl/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. For water injectors, *BHF and *BHW are equivalent.

*BHG

This subkeyword identifies a reservoir gas rate (m^3/day | ft^3/day) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. This keyword is not applicable to water injectors. When applied to gas injectors, it is equivalent to *BHF.

*GOR

This subkeyword indicates that penalization or well cutback for the well is determined from the gas-oil ratio (m^3/m^3 | SCF/STB). Applicable to producers only.

*GLR

This subkeyword indicates that penalization or well cutback for the well is determined from the gas-liquid ratio (m^3/m^3 | SCF/STB). Applicable to producers only.

*WOR

This subkeyword indicates that penalization or well cutback for the well is determined from the water-oil ratio (fraction). Applicable to producers only.

*WGR

This subkeyword indicates that penalization or well cutback for the well is determined from the water-gas ratio (m^3/m^3 | STB/SCF). Applicable to producers only.

*PRS

This subkeyword indicates that rate cutback for the well is determined from the maximum (or minimum for injectors) grid block pressure (kPa | psi) from all the open layers of the well.

*BHP

This subkeyword identifies a bottomhole pressure (kPa | psi) constraint. All producers must have a minimum bottomhole pressure constraint. If *MIN *BHP is missing for a producer then a default minimum bottomhole pressure will be assigned. Maximum BHP constraints for producers are not allowed under *OPERATE, but may be imposed using *MONITOR. Similarly, minimum BHP constraints for injectors are not allowed under *OPERATE, but may be imposed using *MONITOR.

*WHP

This subkeyword identifies a well-head pressure (kPa | psi) constraint. To use this constraint, well tubing data must be entered via the *IWELLBORE or *PWELLBORE keyword. The *WHP constraint can be applied in two different ways, which are identified by the subkeywords:

*WHP *INITIALIZE : well bottom-hole pressure is initialized iteratively at the beginning of a timestep, taking account of changes in bottom-hole pressure but not of pressure changes in surrounding grid blocks. Bottom-hole pressure is not adjusted during the Newtonian iterations.

*WHP *IMPLICIT : well bottom-hole pressure is initialized as in *WHP *INITIALIZE, but is adjusted during the Newtonian iterations. This option may require more Newtonian iterations than does *WHP *INITIALIZE, but will honor the constraint value more accurately. This is the default and is used if *WHP is specified without a subkeyword.

Maximum WHP constraints for producers are not allowed under *OPERATE, but may be imposed using *MONITOR. Similarly, minimum WHP constraints for injectors are not allowed under *OPERATE, but may be imposed using *MONITOR.

*DWN, *DWA, *DWB

This subkeyword identifies a drawdown (kPa | psi) constraint – the difference between the wellbore pressure and the grid block pressure. Only *MAX drawdown constraint is allowed for both producers and injectors. Specified drawdown values of zero are allowed and have the same effect as the *SHUTIN keyword. There are three definitions of drawdown that the user can choose from. *DWN refers to the drawdown in the well's reference layer; symbolically,

$$\Delta P_d = \pm\left(Pblock_{ref} - Pwell_{ref}\right)$$

where the sign ± is used to designate the producer (+) or injector (–). It should be noted that when the reference layer is perforated in a null/pinched block, the simulation will either be terminated if *DWN is set as the primary (first) operating constraint at a well change time; or continue with *DWN constraint taking no effect if it is other than the primary one. *DWA defines the maximum drawdown within all open layers,

$$\Delta P_d = \max_{l,\,open} \pm\left(Pblock_l - Pwell_l\right)$$

which corresponds to the notion of imposing the maximum drawdown constraint to avoid formation damage. *DWB specifies the average drawdown for all open layers:

$$\Delta P_d = \pm \sum_{l,\,open} PI_l\left(Pblock_l - Pwell_l\right) \Big/ \sum_{l,open} PI_l$$

weighted by the total product/injectivity (PI) at the reservoir condition. This shall have the same effect as running the well on an equivalent, time-dependent BHF rate constraint. In situations where the pressure differences between completions in the wellbore differ considerably from the corresponding differences in reservoir pressures, the actual drawdown in each layer may differ considerably from the average value.

*STR

This subkeyword identifies a volume or molar percentage recycling rate constraint, or a guide rate to be normalized internally to 100%. It applies to gas injectors only. Recycling separator gases from all cycling producers are collected. Fractions of each component are then recycled according to the specifications of *RECYCLE *RECFRC. This recycle stream is first mixed with a makeup stream to form the final reinjection fluid. The specified volume or molar percentage of this fluid is then reinjected through this injector. At least one recycling producer must be defined before *STR can be used. If the sum of all values entered after *STR is less than 200 they are interpreted as percentages (allowing the user to recycle up to twice as much as is produced); if greater than

200 they are interpreted as gas guide rates and are normalized internally to sum to 100%. NOTE: Do not use *STR with group control in effect.

value

A real number specifying the constraint value.

*STOP

This subkeyword specifies that the simulation is to be terminated if the constraint is violated.

*CONT

This subkeyword specifies that if the constraint is violated, the well should switch to operating on the constraint, and the simulation continue.

This is the default remedial action for a constraint violation.

*SHUTIN

This subkeyword specifies that if the constraint is violated, the well should be shut in.

*NODAMP

Calculation mode for penalty constraints in which the reference phase target rate (oil rate for GOR and WOR penalties, gas rate for WGR penalty, liquid rate for GLR penalty) is computed by literal application of the penalty formula. For further details please see the EXPLANATION section below. In some cases this calculation mode may cause oscillations in rate.

*DAMP

Calculation mode for penalty constraints in which the geometric mean of the current reference phase rate and the literally computed target rate (that calculated under *NODAMP) is assigned as the reference phase target rate. For further details please see the EXPLANATION section below. This mode is less subject to rate oscillations than is the *NODAMP mode, but when reference phase mobilities are small near the well, this mode (and *NODAMP) may encounter convergence difficulties.

*PHASE2

Calculation mode for penalty constraints in which a rate target is assigned for the second phase (gas for GOR and GLR penalties, water for WOR and WGR penalties). This mode is not subject to rate oscillations and does not encounter convergence difficulties when the reference phase mobility is small, but may result in smaller recovery of the reference phase than do the other two modes. For further details please see the EXPLANATION section below. This mode is the default.

*v*1

A positive real number specifying the base value for the penalty or well cutback type constraints. It works as the upper limit for *GOR, *GLR, *WOR, and *WGR (producers only). For *PRS, it works as the lower limit for producers, and upper limit for injectors. The default values are:

100 m^3/m^3 (561 SCF/STB) for *GOR or *GLR;

100 m^3/m^3 (100 STB/STB) for *WOR;

100 m^3/m^3 (17.8 STB/SCF) for *WGR;

No default for *PRS.

*v*2

Rate cutback factor (fraction) for the well cutback constraints. It is the ratio of the reduced flow rate to the current flow rate of the control phase. It should be a positive real number smaller than 1.d0. If *v*2 is missing or the value entered is larger than or equal to 1.0, internally calculated cutback factor will be applied. *v*1 must be entered before entering *v*2.

*v*3

A positive real number specifying the reversal value for the well cutback constraints. It works as the lower limit for *GOR, *GLR, *WOR, and *WGR (producers only). For *PRS, it works as the upper limit for producers, and the lower limit for injectors. The unit must be consistent with *v*1. The default is to switch off the reversing process. *v*2 must be entered before entering *v*3.

*REPEAT

This subkeyword specifies that if the constraint is violated, the timestep should be repeated after switching to operating on the violated constraint. Every well with this option is entitled to repeat once within one timestep. However, the maximum number of total repeats for this timestep is limited by user input (see keyword *MXCNRPT).

DEFAULTS:

All producers must have a minimum bottomhole pressure constraint. If *MIN *BHP is missing for a producer then a default minimum bottomhole pressure of 101.3 kPa or 14.7 psia will be assigned. This value can be overridden by keyword *BHPDEFAULT *PROD.

All injectors must have a maximum bottomhole pressure constraint. If *MAX *BHP is missing for an injector then a default maximum bottomhole pressure of 1,000,000 kPa or 147,000 psia will be assigned. This value can be overridden by keyword *BHPDEFAULT *INJE.

The default action is *CONT.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group, and must follow the *PRODUCER or *INJECTOR keywords.

EXPLANATION:

The keyword *OPERATE defines well operating constraints. The remedial action when a constraint is violated is also chosen.

The first constraint entered is called the primary operating constraint. The simulator will initially attempt to operate the well on this constraint (i.e. to enforce this constraint as an equality) and monitor all other constraints regardless of whether the other constraints are operating constraints or monitored constraints.

If one of the other specified operating constraints is violated while being monitored and *CONT has been specified, then this violated constraint becomes the operating constraint, and is enforced as an equality constraint (target constraint) instead of being monitored as an inequality constraint.

If more than one constraint is violated and the most serious action is *CONT, the constraints are checked to determine which constraint is the most restrictive, and the most restrictive constraint becomes the well's target constraint.

The hierarchy of actions from most serious to least serious is as shown:

most serious
*STOP
*SHUTIN
*CONT
least serious

The value of the target constraint is changed easily at a subsequent simulation time for a number of wells with the respecification of the *TARGET keyword.

A value for each constraint, regardless of the type of constraint, is required.

For producers, the stock tank conditions for *STO, *STG, *STW, *STL, *STI, and *WTG are defined by the *WELSEP keyword in WELL AND RECURRENT DATA keyword group. For injectors, they are assumed to be the same as those used to calculate the original fluid in place (see *SEPARATOR in the INITIAL CONDITIONS keyword group) unless the *INJECTOR-P, *INJECTOR-T, or *INJECTOR-EOS keywords have been used to override the values used in the initialization.

The following table shows the constraints allowed for each type of well:

water injector	solvent (gas) injector	producer
BHP, WHP	BHP, WHP	BHP, WHP
---	---	STO, BHO
---	STG, BHG	STG, BHG
STW, BHW	---	STW, BHW
---	---	STL, BHL
---	---	STI
---	---	WTG
BHF	BHF	BHF
---	STR	---
DWN, DWA, DWB	DWN, DWA, DWB	DWN, DWA, DWB
---	---	GOR, GLR
---	---	WOR, WGR
PRS	PRS	PRS

At least one recycling producer must be defined before the *STR subkeyword can be used.

For example, to produce a well at a maximum rate of 500 m^3/d, with a minimum bottomhole pressure at 2500 kPa and with monitoring constraints of GOR at 2000 m^3/m^3, water-cut at 98% and monitoring for backflow, the input looks like this:

```
*OPERATE     *MAX        *STO    500.00
*OPERATE     *MIN        *BHP    2500.00     *CONT     *REPEAT
*MONITOR     *GOR                2000.00     *STOP
*MONTIOR     *WCUT               0.98        *STOP
*MONITOR     *BACKFLOW                       *STOP
```

If, while the primary constraint is active, the bottomhole pressure falls below 2500 kPa then the bottomhole pressure constraint becomes the operating constraint.

If more than one operating or monitoring constraint is violated, then the most serious action associated with those constraints is taken.

PENALTY CONSTRAINTS

Some regulatory agencies require that a phase ratio, such as GOR, be measured periodically (e.g. at the end of each month) and that if this measured value exceeds a prescribed base value, the well's production in the coming period (e.g. the next month) must not exceed a specified maximum rate times the ratio of the base and measured values, for example

Allowed rate = (Max rate)*(Base GOR)/(Measured GOR)

where the allowed rate is for the coming month and the GOR was measured near the end of the past month.

When the *PENALTY *GOR, *GLR, *WOR, or *WGR constraints are specified they must be preceded by a maximum reference phase rate constraint entered using the *MAX *STO, the *MAX *STG or the *MAX *STL subkeywords of *OPERATE. This rate is used as the maximum rate in the simulator's computation of the allowed rate. The "reference" phase is oil for the GOR and WOR penalties, gas for the WGR penalty and liquid for the *GLR penalty.

The *PENALTY subkeyword applies only to production wells. If *OPERATE *PENALTY is entered for an injector, the program will terminate with an error message.

When *PENALTY and its subkeywords are used, the well rate during the coming timestep may be reduced by a penalty factor calculated using the GOR which prevails at the beginning of the timestep. If the GOR, GLR, WOR, or WGR at the beginning of the timestep exceeds the specified base value, then the production rate in the coming timestep is decreased in a manner which depends on the calculation mode chosen for the penalty constraint, as described below. The maximum oil or gas rate constraints are still honored. In applying the *PENALTY keyword to simulate regulatory rate limitations, care should be taken to have timestep lengths equal to the regulatory period. This can be achieved with *TIME or *DATE data lines.

The GOR, GLR, WOR, and WGR penalty factors are defined by the following equations :

PF(gor) = Base_GOR / Produced GOR
PF(glr) = Base_GLR / Produced GLR
PF(wor) = Base_WOR / Produced WOR
PF(wgr) = Base_WGR / Produced WGR

Here Produced GOR, GLR, WOR, and WGR are the values at the beginning of the timestep for which the penalty factor is being calculated. If PF exceeds or equals one, then the *PENALTY constraint imposes no limit upon the rate. If PF is less than one, the *PENALTY constraint limits the well rate in the coming timestep as follows.

When the *NODAMP calculation mode is in effect the penalized rate for the coming timestep becomes:

Qo_new = Qo_max * PF

for GOR and WOR penalties,

Qg_new = Qg_max * PF

for WGR penalty, and

Ql_new = Ql_max * PF

for a GLR penalty. `Qo_max`, `Qg_max` and `Ql_max` are the maximum stock-tank oil, gas and liquid rates entered for the penalized well using the *OPERATE *MAX *STO, *OPERATE *MAX *STG and *OPERATE *MAX *STL keywords. If the timestep size is equal to the regulatory period this corresponds to a literal application of the expression above for Allowed rate. The *NODAMP mode is subject to oscillations in the assigned rates.

If the *DAMP mode is chosen, then the geometric mean of `Qo_new`, `Qg_new` or `Ql_new` as computed above and the current rate (the rate assigned in the timestep just finished) is assigned as the rate in the coming timestep. In this case,

Qo_new = sqrt(Qo_base * PF * Qo_old)

for GOR and WOR penalties,

Qg_new = sqrt(Qg_base * PF * Qg_old)

for a WGR penalty, and

Ql_new = sqrt(Ql_base * PF * Ql_old)

for a GLR penalty. This mode is less subject to severe oscillations in assigned rate than is the *DAMP mode, but both *DAMP and *NODAMP may give rise to simulator convergence problems when the reference phase mobility is small near the well.

If the *PHASE2 mode is chosen, and PF is less than one, then a target rate is assigned for the second phase (gas for GOR and GLR, water for WOR and WGR penalties) equal to

Qg_target = Qo_base * Base_GOR

for the GOR penalty,

Qw_target = Qo_base * Base_WOR

for the WOR penalty,

Qw_target = Qg_base * Base_WGR

for the WOR penalty, or

Qg_target = Ql_base * Base_GLR

for the GLR penalty. This mode is not subject to oscillations in assigned rate and is not affected by small reference phase mobilities near the well. However, in situations in which production penalties are imposed by regulation, the measured phase ratio (e.g. the GOR) tends to rise over the course of a timestep. Since the *DAMP and *NODAMP modes assign rates based on GOR values at the start of the timestep, the *DAMP and *NODAMP modes usually result in higher rates for the second phase (and hence also the reference phase) than are allowed by the *PHASE2 mode.

If the well is operating under a constraint other than a reference phase (for *NODAMP and *DAMP modes) or second phase (for *PHASE2) rate constraint with a *PENALTY constraint specified, and the calculated penalty factor is less than one, then the well's target constraint is switched to a rate constraint for the proper phase.

Examples: (Usage of *OPERATE with *PENALTY subkeyword)

```
*OPERATE *MAX *STO  3000.0     *CONT
*OPERATE *MIN *BHP  1500.0     *CONT *REPEAT
*MONITOR *BACKFLOW             *STOP
*MONITOR *GOR      50000.0     *STOP
*OPERATE *PENALTY *GOR *PHASE2  393.0
*OPERATE *PENALTY *WOR *DAMP    50.0
--------------------------------
*OPERATE *MAX *STG 3000000.0   *CONT
*OPERATE *MIN *BHP    1200.0   *CONT *REPEAT
*MONITOR *WGR         1600.    *STOP
*OPERATE *PENALTY *WGR *PHASE2   150.0
```

Default Bottom-hole Pressure Constraint Values (Optional)

***BHPDEFAULT**

PURPOSE:

Assign default bottom-hole pressure constraint values for wells.

FORMAT:

*BHPDEFAULT *PROD | *INJE *bhpval*

DEFINITIONS:

*PROD
: Assign default bottom-hole pressure constraint values for producers.

*INJE
: Assign default bottom-hole pressure constraint values for injectors.

bhpval
: A positive real number defining the default bottom-hole pressure constraint value (kPa | psi).

DEFAULTS:

If *BHPDEFAULT is absent and if *OPERATE (*MIN) *BHP is missing for a producer then a default minimum BHP of 101.3 kPa or 14.7 psia will be assigned; if (*MAX) *BHP is missing for an injector then a default maximum BHP of 1.0e+6 kPa or 147,000 psia will be assigned.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

For wells that do not have any explicit definition of the bottom-hole pressure constraint, the default values will be applied. This keyword does not have any effect to wells that have already the BHP constraint defined through keyword *OPERATE.

Set Injection Composition (Optional) *INCOMPWL

PURPOSE:

*INCOMPWL sets the composition of injected lift-gas for producers or injected fluid for hydrocarbon (solvent) injectors.

FORMAT:

*INCOMPWL *well_list*
values

DEFINITIONS:

well_list
One or more well numbers or quoted well names assigned with the *WELL keyword. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

values
Mole fractions of components of the injected fluid or lift-gas in the order defined in the component properties section led by keyword *MODEL. *values* must appear on one or more new lines immediately following *INCOMPWL line, must NOT appear on the same line as *well_list*. They should sum to one, but will be normalized if not. Note that if the pseudo-component option is used, *values* refer to the composition in terms of the pseudo-components.

DEFAULTS:

Optional keyword. Default is to use the composition of injected fluid specified by subkeyword *SOLVENT under *INCOMP for injectors, and the composition from field lumped separator gas for producers on gas lift.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group after wells have been defined with the keyword *WELL, and well types have been defined with the keyword *PRODUCER or *INJECTOR. Any condition on *INCOMP *SOLVENT also applies to *INCOMPWL for hydrocarbon injectors.

EXPLANATION:

This keyword sets the composition of injected lift-gas for producers on gas lift when *GLIFT is used with *PWELLBORE *MODEL. It also allows the user to reset the composition of injected fluid for a hydrocarbon (solvent) injector at any well change time without having to redefine the whole well via *INJECTOR *INCOMP. The keyword has no effect on listed wells that are not a producer nor a hydrocarbon injector. Note that *INCOMPWL does not define/change the injectors' type as opposed to *INCOMP.

Example:

```
*DATE 2000 01 01
   *INJECTOR 'Injector'
   *INCOMP SOLVENT  1.0 0.0
   ......

   *PRODUCER 'Producer'
   ......

   *GLIFT *OPT 'Producer'
   *INCOMPWL 'Producer'
        0.8 0.2     ** set lift-gas composition

*DATE 2000 02 31
   *INCOMPWL 'Injector'
     0.8 0.2        ** reset solvent composition
```

Well Current Constraint (Optional) *WCURRCN

PURPOSE:

*WCURRCN enables wells to run on their current operating constraints without checking for constraint violation and possible switching.

FORMAT:

*WCURRCN (*well_list*) (*ON | *OFF)

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*ON | *OFF

Turns on or off the feature.

DEFAULTS:

Optional keyword. The default is *WCURRCN *OFF for all wells. Use of *WCURRCN with no occurrence of *ON or *OFF has the effect of setting *ON for the wells listed.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Wells on *well_list* must have been defined with the keyword *WELL.

EXPLANATION:

This keyword allows the user to operate wells on their current constraints without checking for constraint violation and possible constraint switching to the most restrictive. This effectively disables all the other constraints while *ON remains. Furthermore, wells with *WCURRCN turned on are automatically removed from participating in the apportionment of higher-level group production or injection targets. However, their rates still contribute to the higher-level targets since the wells are still members of the group structure.

Note that the current operating constraint is by default the primary (first) well constraint following the *OPERATE keyword, and can be redefined at any well change time by keyword *TARGET. Any specified well monitoring constraint will not be affected by *WCURRCN.

Example:

```
*DATE 2000 01 01
   *WCURRCN  1:3  *ON        ** wells 1, 2, 3
    or
   *WCURRCN *ON              ** all wells
```

Maximum Number of Continue-Repeat (Optional) *MXCNRPT

PURPOSE:

*MXCNRPT specifies maximum number of continue-repeat allowed in each timestep.

FORMAT:

*MXCNRPT *nmxrpt*

DEFINITIONS:

nmxrpt
A non-negative integer defining the maximum allowed number of timestep repeats due to well constraint switches.

DEFAULTS:

The default is 1. At most, one timestep repeat due to well constraint switch is allowed per timestep, no matter which well(s) initiates the constraint switch.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It has no effect if none of the wells has *CONT *REPEAT specified through *OPERATE.

EXPLANATION:

Every well having *CONT *REPEAT (continue-repeat) specified through *OPERATE is entitled to one repeat of the current timestep if one of its operating constraints is violated. During convergence of a timestep, it is possible that different wells violate and then switch their operating constraints consecutively after each repeat. Although the maximum number of possible repeat cannot exceed the number of wells, it may lead to a significant increase of run time if there are many wells. Quantity *nmxrpt* limits the number of such repeats per timestep, and keyword *MXCNRPT lets the user override its default value.

Monitored Well Constraints (Optional) *MONITOR

PURPOSE

*MONITOR defines the monitored well constraints, and the remedial action to be taken when a monitored constraint is violated. Monitored constraints differ from operating constraints in that a monitored constraint can never be imposed directly as a well target. The constraints are checked and the actions are applied only after a timestep has been completed; actions are not applied part way through a timestep.

FORMAT:

*MONITOR monitored_constraint *value* action

where:

monitored_constraint for producers is one of:

*GOR
*WCUT
*WGR
*WWGR
*GWGR
*BACKFLOW
*WHYSTAB
*RATSTAB

*MIN	*STO
*MIN	*STG
*MIN	*STI
*MIN	*WTG
*MAX	*STG
*MAX	*STW
*MAX	*STL
*MAX	*WTG
*MAX	*MXX
*MIN	*MXX
*MAX	*WHP
*MAX	*BHP

monitored_constraint for injectors is one of:

*BACKFLOW
*WHYSTAB
*RATSTAB

*MIN	*STG
*MIN	*STW
*MIN	*REC
*MIN	*WHP
*MIN	*BHP

action for producers for monitored_constraint other than *BACKFLOW, *WHYSTAB, *RATSTAB, *MAX *WHP, or *MAX *BHP is one of:

*STOP
*SHUTIN (*REPEAT)
*AUTOWELL *frequency*
*RECOMPLETE (*UP | *DOWN | *UP_OR_DOWN)
*SHUTLAYER
*SHUTLAYER1
*AUTOLAYER *frequency*
*AUTOLAYER1 *frequency*
*WORKOVER (*pi_factor* *ntimes* wo_action)

action for injectors for monitored_constraint other than *BACKFLOW is one of:

*STOP
*SHUTIN (*REPEAT)

action when *BACKFLOW is monitored is one of:

*STOP
*SHUTIN (*REPEAT)
*AUTOLAYER *frequency*

action when *WHYSTAB or *RATSTAB is monitored is one of:

*STOP
*SHUTIN (*REPEAT)
*AUTOWELL *frequency*

action for producers when *MAX WHP or *MAX BHP is monitored is one of:

*STOP
*SHUTIN (*REPEAT)
*AUTOWELL *frequency*

No other actions are permitted.

DEFINITIONS:

*GOR

This subkeyword identifies a maximum gas-oil ratio (m^3/m^3 | scf/STB) monitor for a producer.

*WCUT

This subkeyword identifies a maximum water-cut (fraction) monitor for a producer. The water-cut is defined as (water_rate)/(total liquid rate); when no Intermediate Liquid stream is present for the separator (see the entry under *WELSEP) , total liquid rate = water rate + oil rate; when the Intermediate Liquid stream is present, total liquid rate = water rate + oil rate + INL rate.

*WGR

This subkeyword identifies a maximum water-gas ratio (m^3/m^3 | STB/scf) monitor for a producer.

*WWGR

This subkeyword identifies a maximum water-wet gas ratio (m^3/m^3 | STB/scf) monitor for a producer.

*GWGR

This subkeyword identifies a maximum gas-wet gas ratio (m^3/m^3 | scf/scf) monitor for a producer.

*BACKFLOW

This subkeyword identifies backflow monitoring. No value is required for this monitor; only an action is required. The default action is to print a message with no further action taken.

*WHYSTAB

This subkeyword identifies wellbore hydraulics stability monitoring. A real well can operate only above a certain rate at which the bottom-hole pressure remains in the stable region of the well-head pressure curve (a region in which the curve of WHP vs. BHP has a positive slope). It can only be used for wells for which a method of computing WHP has been introduced with the *PWELLBORE or *IWELLBORE keyword. No value is required for this monitor, but an action is required. The default action is to print a message with no further action taken.

*RATSTAB

This subkeyword identifies wellbore rate stability monitoring. A real well should operate only above a certain rate at which the bottom-hole pressure remains in the stable region of the tubing-intake or tubing-outflow curve (a region where the BHP vs. rate has a positive slope for producers or negative slope for injectors while WHP is held constant). For details, see Brill J. P. and Mukherjee H., “Multiphase Flow in Wells”, Monograph SPE Henry L. Doherty Series, Vol 17, pp. 78-97. It can only be used for wells for which the method of computing WHP is to use the hydraulic tables (*PTUBE1 or *ITUBE1) introduced with the *PWELLBORE or *IWELLBORE keyword. No value is required for this monitor, but an action is required. The default action is to print a message with no further action taken.

Please note that this monitor will be ignored if the well operating conditions are out of the ranges of the used hydraulic tables.

*MIN

This subkeyword indicates that a rate, BHP, WHP, or wellstream mole percent constraint is to be monitored as a minimum.

*MAX

This subkeyword indicates that a rate, BHP, WHP, or wellstream mole percent constraint is to be monitored as a maximum.

*STO

This subkeyword identifies a surface oil rate (m^3/day | Stb/day) constraint for a producer.

*STI

This subkeyword identifies a surface Intermediate Liquid stream rate (m^3/day | Stb/day) constraint for a producer. For more information on the Intermediate Liquid stream please see the entry in this manual for the *WELSEP keyword.

*STG

This subkeyword identifies a surface gas rate (m^3/day | scf/day) constraint.

*WTG

This subkeyword identifies a surface wet gas rate (m^3/day | scf/day) constraint.

*STW

This subkeyword identifies a surface water rate (m^3/day | Stb/day) constraint.

*STL

This subkeyword identifies a total surface liquid (oil + water + intermediate liquid) rate (m^3/day | Stb/day) constraint for a producer.

*MXX

This subkeyword identifies a well stream composition (mole percent) constraint for a producer. XX is the component number to be monitored.

*REC

This subkeyword identifies a volumetric recycle rate (m^3/day | scf/day) constraint for a recycling injector. It can ONLY be used as a MINIMUM (NOT maximum) constraint. If the cycling injector is attached to a group, then the production from the cycling producers connected to the same group (plus any make-up rate specified for this group) will be used to determine whether the constraint is violated. If no groups are specified, then production from all cycling producers will be summed and the make-up rate added. At least one recycling producer must be defined before this subkeyword can be used. See *CYCLING for more information on the recycling option.

*WHP

This subkeyword identifies a maximum (producers) or minimum (injectors) well-head pressure (kPa | psi) monitor. It can only be used for wells for which a method for computing WHP has been introduced with the *PWELLBORE or *IWELLBORE keywords.

*BHP

This subkeyword identifies a maximum (producers) or minimum (injectors) bottom-hole pressure (kPa | psi) monitor.

value

A real number specifying the monitored value.

*STOP

This subkeyword specifies that the simulation be terminated if the constraint is violated.

*SHUTIN

This subkeyword specifies that if the constraint is violated, the well should be shut in. This is the default remedial action for a constraint violation.

*REPEAT

This subkeyword of the *SHUTIN action specifies that the timestep should be repeated after the indicated *SHUTIN action is taken.

*AUTOWELL

Similar to *SHUTIN except that well productivity is checked periodically and the well is reopened automatically if the previously violated monitored ceases to be violated. The *AUTOWELL action is valid for the *MIN *STO, *MIN *STG, *GOR, *WCUT, *WGR, *MAX MXX, *MIN *MXX, *MAX *WHP, *MAX *BHP, *WHYSTAB and *RATSTAB monitors for producers only.

frequency

This is the frequency in days at which the checking which is part of the *AUTOWELL or *AUTOLAYER (or *AUTOLAYER1) actions should be performed in an attempt to reopen the well or layer.

*RECOMPLETE

Plug the most offending layer (the one with the largest value of a maximum constraint or the smallest value of a minimum constraint) and, if possible, open a currently plugged layer in the well. This layer may have been plugged through earlier constraint violation action or through specification of the *AUTO layer status with the *PERF keyword. *RECOMPLETE is a valid action for all allowed monitored constraints for producers.

*UP

Keyword specifying that the recompletion must be done in the upward direction from the most offending layer.

*DOWN

Keyword specifying that the recompletion must be done in the downward direction from the most offending layer.

*UP-OR-DOWN

Keyword specifying that the recompletion will be done upward from the most offending layer for WCUT, MIN STO, MIN STG, MIN WTG, MAX STW, MAX STL, WGR, and WWGR violations, and downward from the most offending layer for GOR, GWGR, MAX STG, MAX WTG, MAX MXX, MIN MXX, and BACKFLOW violations.

*SHUTLAYER

Plug the most offending layer (which has the highest/lowest value of the monitored variable) and continue the simulation.

*SHUTLAYER1

Plug the most offending layer (which, when shut, causes the well to have the lowest/highest value of the monitored variable) and continue the simulation. *SHUTLAYER1 has the same effect as *SHUTLAYER for a rate type monitor (e.g. STO), but will be different for a ratio type monitor (e.g. GOR).

*AUTOLAYER

Similar to *SHUTLAYER except that the plugged layer is checked periodically and if the previously violated monitored constraint ceases to be violated, the layer is reopened automatically. *AUTOLAYER is valid only for the *GOR, *WCUT, *WGR, *WWGR, *GWGR, and *BACKFLOW monitored constraints. When *AUTOLAYER is the action for the *BACKFLOW monitor, a layer is opened at the checking time only if at least one layer of the well has remained open; i.e., for *BACKFLOW the *AUTOLAYER action will not reopen a well which has been shut because all of its layers have been plugged. For the other monitors, *AUTOLAYER may reopen a well. *AUTOLAYER may be applied to both producers and injectors for *BACKFLOW monitor.

*AUTOLAYER1

Similar to *AUTOLAYER but corresponding to *SHUTLAYER1.

*WORKOVER

Increases the well productivity indices for all layers of the well by a factor *pi_factor* and continues the simulation. This workover is applied a maximum of ntimes in the simulation. The *WORKOVER action is valid only for the *MIN *STO, *MIN *STG, *MIN STI, and *MIN *WTG monitored constraints.

pi_factor

A real number specifying the improvement in all the layer productivities for the well:

New_PI = Old_PI * *pi_factor*.

The entry of *pi_factor*, *ntimes*, and wo_action after *WORKOVER is optional; the recommended procedure is to use the *WORKOVER-DATA keyword to enter the workover parameters; please consult the manual page for *WORKOVER-DATA.

ntimes

The maximum number of workovers allowed before wo_action is taken instead of the productivity index increment. The entry of *pi_factor*, *ntimes*, and wo_action after *WORKOVER is optional; the recommended procedure is to use the *WORKOVER-DATA keyword to enter the workover parameters; please consult the manual page for *WORKOVER-DATA.

wo_action

The action to be taken if the monitor is violated after the maximum number of workovers has been exhausted. wo_action can be one of the following: *STOP, *SHUTIN or *SHUTIN *REPEAT. The default is *SHUTIN.

The entry of *pi_factor*, *ntimes*, and wo_action after *WORKOVER is optional; the recommended procedure is to use the *WORKOVER-DATA keyword to enter the workover parameters; please consult the manual page for *WORKOVER-DATA.

DEFAULTS:

Optional keywords. No defaults.

The default action is *SHUTIN. See the manual page for *WORKOVER-DATA for the default values of *pi_factor*, *ntimes*, and wo_action.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group, and must follow the *PRODUCER, *CYCLPROD or *INJECTOR.

EXPLANATION:

The keyword *MONITOR defines monitored well constraints, i.e. constraints which are not to be set as targets if violated but which cause a specified remedial action to be taken.

Some quantities can be monitored only for certain well types. For instance, *GOR, *WCUT, *WGR, *WWGR, *GWGR and *MXX and the *WTG and *STI monitors are used only for producing wells, while the *MIN *REC monitor is valid for gas cycling injectors only.

*BACKFLOW, *WHYSTAB and *RATSTAB can be monitored for either producers or injectors, and they are the only monitors that do not require a value.

For example, to produce a well at a maximum rate of 500 m^3/day, with a minimum bottom-hole pressure at 500 kPa and with monitoring constraints of GOR at 25000 m^3/m^3, water-cut at 98% and monitoring for backflow, the input looks like this:

```
*OPERATE *MAX *STO   500.00
*OPERATE *MIN *BHP   500.00 *CONT *REPEAT
*MONITOR *GOR      25000.00 *STOP
*MONITOR *WCUT   0 .98      *STOP
*MONITOR *BACKFLOW          *AUTOLAYER  10.0
```

If, while the primary constraint is active, the bottom-hole pressure falls below 500 kPa then the bottom-hole pressure constraint becomes the operating constraint.

If, while the primary constraint is active, the bottom-hole pressure constraint, the GOR constraint, and the backflow monitor are violated simultaneously, then the most serious of the specified actions for the three violated monitors will be taken. In this case, the simulation is terminated, since *STOP is specified for a GOR violation.

A well is considered to be operating normally if, in all layers,

for an injector Pbh + head > P(block)
for a producer Pbh + head < P(block)

If the above do not hold in all layers, then the well is considered to backflow.

The acceptable range of values for gas-oil ratios is:

	SI m^3/m^3	**Field** scf/STB
min	0.0	0.0
max	1.0E+20	5.615E+20

The acceptable range of values for water-gas ratios is:

	SI m^3/m^3	Field STB/scf
min	0.0	0.0

Alter Primary Well Operating Constraint Value (Optional)

***ALTER**

PURPOSE:

*ALTER allows modification of only the primary operating constraint for the well(s) whose names or numbers are listed. The primary operating constraint is the FIRST constraint entered for the well using the *OPERATE keyword.

FORMAT:

*ALTER *well_list*
values

DEFINITIONS:

well_list
A list of 'well_names' or well_numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

values
One number for each well identified by *well_list* specifying the new value of the primary operating constraint. Values must appear on a new line immediately after the *ALTER line.

DEFAULTS:

Optional keywords. No defaults.

CONDITIONS:

*ALTER must be located in the WELL AND RECURRENT DATA keyword group, and may appear anywhere in this keyword group following the initial *OPERATE declarations for all of the wells in the *ALTER list. All of the wells read in the list must have already had the well type defined as *PRODUCER or *INJECTOR. If a listed well has not yet had its type defined, an error message is issued and the run is terminated.

EXPLANATION:

This optional keyword is used to alter the primary operating constraint for a well or set of wells without having to redefine all of the additional operating constraints. It is an effective method of altering values when performing a history match.

*ALTER, followed by a non-zero value, also opens a well if the well has been shut in by a previous action or if the well has been initially defined as a shut in well.

When *ALTER is encountered in a data set, the simulator checks if the primary constraint with the new value becomes the most restrictive well constraint. If so, the well is switched to the primary constraint. If not, the new value is entered for the primary constraint but the well is switched to (or continues to run on) the currently most restrictive constraint.

If a primary constraint value of zero is specified using *ALTER for an open well then that well is shut in. If a non-zero value is specified for a shut in well then that well is opened.

*ALTER has essentially the same effect as *TARGET, except that *TARGET can apply to constraints other than the primary constraint.

The *TARGET keyword is a more flexible version of the *ALTER keyword. The *ALTER keyword allows for the modification of only the primary operating constraint (the FIRST operating constraint specified for a well). The *TARGET keyword, on the other hand, allows for the specification of any operating constraint.

Examples:

```
*PRODUCER 1
*OPERATE *MAX *OIL 500.00
:
*ALTER 1
      750
```

The *ALTER keyword may also look like this when several wells have been defined:

```
*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
*WELL 3 'Producer 3'
*WELL 4 'Injector 1'
:
*PRODUCER 1
*OPERATE *MAX *OIL 500.00
*PRODUCER 2
*OPERATE *MAX *OIL 750.00
*PRODUCER 3
*OPERATE *MAX *BHP 2500.0

*INJECTOR 4
:
*TIME 1200.
** At a later date, want to adjust the operating
** constraint values.
      ** well_numbers
*ALTER  1:2    3
** values
      2*1000.0  800.0
```

Alter Well Constraint Value (Optional) *TARGET

PURPOSE:

*TARGET allows modification of any previously specified well constraint value or the specification of a new constraint type and value for well(s).

FORMAT:

*TARGET *constraint_type well_list*
value_list

DEFINITIONS:

constraint_type
A constraint type that is valid for *OPERATE for wells in *well_list*. See *OPERATE for the full list and explanations of constraint types.

well_list
One or more well numbers or quoted well names to specify the wells to which this alteration applies. The *well_list* may be on the same line as the *TARGET keyword. If more than one line is required for the desired well list, a separate *TARGET keyword must be used. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

value_list
List of values, one for each well in the order specified by *well_list*. The first value must appear on the line immediately following *well_list*. Values must NOT appear on the same line as *well_list*. Values refer to the base limits for the well cutback constraints.

DEFAULTS:

Optional keywords. No defaults.

CONDITIONS:

*TARGET must be located in the WELL AND RECURRENT DATA keyword group, and may appear anywhere in this keyword group following the initial *OPERATE declaration. All wells appearing in the well list must already have had their type defined with *PRODUCER, *INJECTOR, or *CYCLPROD. If any listed well has not yet had its type defined, an error message is issued and the run is terminated.

EXPLANATION:

This optional keyword is used to alter a constraint value for wells without having to redefine all of the additional operating constraints. It is an effective method of altering constraints when performing a history match.

*TARGET also opens a well if the well has been shut in by a previous action or if the well has been initially defined as a shut in well. When *TARGET is encountered in a data set, the simulator checks if the altered constraint with the new value becomes the most restrictive

well constraint. If so, the well is switched to this new constraint type. If not, the new value is entered for the constraint but the well is switched to (or continues to run on) the currently most restrictive constraint.

The target type must be valid for *OPERATE for wells in *well_list*.

The *TARGET keyword is a more flexible version of the *ALTER keyword. The *ALTER keyword allows for the modification of only the primary operating constraint (the FIRST operating constraint specified for a well). The *TARGET keyword, on the other hand, allows the specification of any operating constraint.

Examples:

```
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
:
*TARGET *STO ** This alters the target constraint in
1            ** the *TARGET keyword above, i.e.,
750          ** *STO.
```

The *TARGET keyword may also look like this when several wells have been defined:

```
*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
*WELL 3 'Producer 3'
*WELL 4 'Injector 1'
:
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 2
*OPERATE *MAX *STO 750.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 3
*OPERATE *MIN *BHP 2500.0
*MONITOR *MIN *STO 10.00 *SHUTIN

*INJECTOR 4
*OPERATE *MAX *STW 100.0
*MONITOR *MIN *STW 10.00 *SHUTIN
:
*TIME 1200.
** At a later date, want to adjust the target
** constraint values.

*TARGET *STO
** well_numbers
     1:2
** values
   2*1000.0

*TARGET *BHP
** well_numbers
     3
** values
    800.0
```

```
*TARGET *STW
** well_numbers
     4
** values
     50.0
```

Operating Constraints in History Matching Mode (Optional)

***OPERATE-HIST**

PURPOSE:

*OPERATE-HIST indicates that well operating constraints defined by *OPERATE are under the special history-matching mode and the constraint values entered for surface phase rates serve as the observed flow rates.

FORMAT:

*OPERATE-HIST (*well_list*) (*type*) (*sector_name*)
type = *RSV | *STO | *STG | *STW | *STI | *STL | *OFF

DEFINITIONS:

well_list

A list of well names or well numbers to which this special mode applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. The history matching mode will apply to ALL wells if *well_list* is not input, including wells that have not been defined.

type

Subkeyword to define the effective constraint type that the wells will run on for history matching. It must be chosen from one of the following: *RSV, *STO, *STG, *STW, *STI, *STL and *OFF. When *RSV is specified, the wells will be put on the total reservoir fluid rate constraint whose value is calculated internally at the reference condition based on the observed surface flow rates. For *STO, *STG, *STW, *STI, or *STL the wells will be put on the corresponding surface oil, gas, water, liquid(oil + water) or intermediate liquid rate constraint, without checking of constraint violations among them. An error message is issued if the chosen phase is not specified by *OPERATE. The special history-matching mode will be turned off when *OFF is entered at any well change time. The default for *type* is *RSV.

sector_name

Sector name must be 16 characters maximum and must have already been defined in the RESERVOIR DESCRIPTION section in the input data. If *sector_name* presents and is on the same line as *well_list*, one of the *con_type* subkeywords must also present, as a separator between *well_list* and *sector_name*. Only one sector name is allowed per keyword. The sector's hydrocarbon pore-volume weighted average pressure and block-volume weighted average temperature (if applicable) will be used as the reference condition for converting *RSV. If the *sector_name* is omitted, the average well-block pressure mobility-weighted at the reservoir condition, and the well-block volume averaged temperature (if applicable) will be used instead.

DEFAULTS:

Optional keyword. Default is not to operate wells in the special history-matching mode.

CONDITIONS:

*OPERATE-HIST must be located in the WELL AND RECURRENT DATA keyword group, and may appear anywhere in this keyword group after the well names are available.

EXPLANATION:

*OPERATE-HIST enables the well to run in the special history-matching mode. The observed surface flow rates (*STO, *STG, *STW, *STI) are specified through keyword *OPERATE as regular operating constraints. Zero flow rates are assumed for unspecified surface streams. For the default *RSV, the specified surface phase rate constraints are converted to the total reservoir fluid rate (*BHF) at the beginning of each time step, corresponding to the reference condition as chosen by the user. The wells will be running on this total reservoir fluid rate as long as the current most restrictive constraint is one of the surface phase constraints. *OPERATE-HIST applies to both producers and injectors. It also opens a previously shut-in well if non-zero value is specified as any of the observed flow rates.

This option is useful in history matching runs. The calculated surface phase rates may not honor the specified flow rates individually if the mobility ratios have not been fully matched. This control mode is useful for making the well produce/inject the correct amount of voidage from/to the reservoir or sectors before the mobility ratios are fully matched. Therefore the rate of pressure changes should be approximately correct.

Examples:

Producer #1 will be running on an ‘equivalent’ *BHF constraint as long as any of the specified surface phase constraints (*STO, *STG, *STW) remains most restrictive. *OPERATE-HIST ceases to have any effect when the *BHP constraint is violated.

```
*PRODUCER 1
**Input observed flow rates:
*OPERATE *MAX *STO 500.0
*OPERATE *MAX *STG 100000.0
*OPERATE *MAX *STW 1.0
*OPERATE *MIN *BHP 500.0

*OPERATE-HIST 1 *RSV  **Run well #1 for history-matching

**or

*OPERATE-HIST 1 *RSV ‘FIELD’   **Change reference condition

*TIME 1200.0
**Turn off history-matching for well #1:
*OPERATE-HIST 1 *OFF
```

Alter Observed Flow Rates for History-Matching (Optional)

***ALTER-HIST**

PURPOSE:

*ALTER-HIST allows modification of the observed flow rates for wells running in the special history-matching mode.

FORMAT:

*ALTER-HIST *well_list*
values

DEFINITIONS:

well_list

A list of well names or well numbers to which this alteration of flow rate values applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

values

Observed flow rate values. Numbers and orders must match the surface phase rate constraints as defined by keyword *OPERATE for each well appearing in the *well_list*.

DEFAULTS:

Optional keywords. No defaults.

CONDITIONS:

*ALTER-HIST must be located in the WELL AND RECURRENT DATA keyword group, and may appear anywhere in this keyword group following the initial *OPERATE declarations for all of the wells in the *ALTER-HIST list. All of the wells read in the list must have already had the well type defined as *PRODUCER or *INJECTOR. If a listed well has not yet had its type defined, or does not have the special history-matching mode as specified by *OPERATE-HIST, an error message is issued and the run is terminated.

EXPLANATION:

This optional keyword is used to alter the observed flow rates for a well or set of wells running in the history-matching mode without having to redefine all of the additional operating constraints. It is an effective method of altering values when performing a history match.

*ALTER-HIST, followed by a non-zero value of any of the defined surface phase constraints (*STO, *STG, *STW, *STI), opens a well if the well has been shut in. If zeros are entered for all the defined surface phase rates using *ALTER-HIST, it shuts in a well if the well has been open.

Keyword *ALTER-HIST is different than keyword *ALTER since the latter only alters values of the primary\first well constraint which is not necessarily a surface phase rate.

Examples:

```
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*OPERATE *MAX *STG 100000.0
*OPERATE *MAX *STW 1.0
*OPERATE *MIN *BHP 500.0

*OPERATE-HIST 1 *RSV  **Run history-matching for well #1
:
*ALTER-HIST 1
**    STO   STG      STW
      750.0 120000.0 2.0
```

The *ALTER-HIST keyword may also look like this when several wells have been defined:

```
*TIME 0.0

*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
*WELL 3 'Producer 3'
*WELL 4 'Injector 1'
:
*PRODUCER 1
*OPERATE *MIN *BHP 300.0
*OPERATE *MAX *STO 400.0              **1
*OPERATE *MAX *STG 50000.0            **1
*PRODUCER 2
*OPERATE *MAX *STG 50000.0            **2
*OPERATE *MIN *BHP 300.0
*OPERATE *MAX *STW 1.0                **2
*INJECTOR 3
*OPERATE *MAX *STG 70000.0            **3


*OPERATE-HIST 1:3  **Run history-matching for wells #1-#3
:
:
*TIME 1200.0
**To adjust the observed flow rates at a later time.

*ALTER-HIST  1:3  **Must match *OPERATE of each listed well
**    STO-1 STG-1    STG-2   STW-2   STG-3
      401.0 50001.0  50001.0 1.1     70001.0
```

Well Head Method (Optional) *HEAD-METHOD

PURPOSE:

*HEAD-METHOD is used to identify the head method to be used.

FORMAT:

*HEAD-METHOD (*well_list*)
*GRAVITY | *ZERO-HEAD |
*GRAV-FRIC (*HEADRROUGH *rrough*) (*HYRADIUS *hedrad*)

DEFINITIONS:

well_list
A list of well names or well numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*GRAVITY
The head between well layers is calculated based on mobility weighted densities.

*ZERO-HEAD
The head between well layers is set to zero. This may be used, for instance, to model a submersible pump where the liquid column height in the annulus is small compared to the perforated length of the well and the gas density at reservoir conditions is small.

*GRAV-FRIC
The head between well layers is calculated using a correlation which takes account of fluid densities, frictional effects, and kinetic energy effects.

*HEADRROUGH *rrough*
Subkeyword of *GRAV-FRIC introducing input of the value of relative roughness to be used for the wellbore frictional head calculation. Default value of *rrough* is 0.0001 if *HEADRROUGH does not appear in the line. Dimensionless.

*HYRADIUS *hedrad*
Subkeyword of *GRAV-FRIC introducing input of the value of hydraulic radius (m | ft) to be used for the wellbore frictional head calculation. Default value of *hedrad* is taken from the well radius supplied by the *GEOMETRY keyword or the *PERF keyword via *RW.

DEFAULTS:

For GEM, the default head method is *GRAV-FRIC.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword identifies the well head method to be used for a given list of wells specified by name or number.

The wellnum may be a single integer, a series of integers delimited by spaces or commas, or a range of integers defined in the form: imin:imax.

Example:

```
*WELL 1 'Producer1'
*WELL 2 'Injector1' *VERT 12 14
*WELL 3 'Prod. 54-233a'
*HEAD-METHOD 1 *ZERO-HEAD
*HEAD-METHOD 2 *GRAVITY
*HEAD-METHOD 3 *GRAV-FRIC *HYRADIUS 0.15
```

Well Head Iteration (Optional) *HEADITER

PURPOSE:

*HEADITER specifies that certain wells are to have the wellbore pressure differences between completions (heads) computed iteratively at the beginning of a timestep to achieve consistency between completion pressures and the computed heads. See also the manual page for the related keyword *BHPHEADINIT.

FORMAT:

*HEADITER (*well_list*) *ONE-STEP | *ITERATIVE

DEFINITIONS:

well_list
: A list of 'well_names' or well_numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*ONE-STEP
: The heads between well layers are computed without iteration to consistency. This is the default and was the only method available before the introduction of the *HEADITER keyword (*1-STEP is a permitted variant of this keyword).

*ITERATIVE
: The heads between well layers are computed iteratively to achieve consistency between the calculated pressure drops in the wellbore and the completion pressures. This method may be useful in eliminating rate oscillations from timestep to timestep in horizontal gas wells.

DEFAULTS:

For GEM, the default setting is *ONE-STEP.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

*HEADITER allows the user to direct that the head calculation done at the beginning of a timestep should be done iteratively to achieve consistency between the completion pressures and the computed pressure differences between completions. Lack of consistency may be manifested as oscillations of non-controlled rates from timestep to timestep.

Our experience has been that *HEADITER suffices to eliminate oscillations in horizontal wells in which only gas flows; the related keyword *BHPHEADINIT may be necessary for backflowing wells with significant depth differences between completions and some liquid flow.

If the wellbore pressure in completion k is denoted as p_{wk} and the surrounding grid block pressure as pbk, then the flow rate in the wellbore depends upon the difference ($p_{bk} - p_{wk}$). If a full set of p_{wk} values are chosen for a well, then the flow rates in all segments of the well are determined. From these flows, heads may be determined (see the manual page for the *HEAD-METHOD keyword); the head hk for completion k is the difference computed between the well's BHP and the pressure in completion k. In the absence of a procedure to enforce the equality of pwk (the pressure value used to compute the completion flow rates) and BHP + h_k (the pressure value resulting from these flow rates), there is no guarantee of consistency between these two values for the completion pressure. The *HEADITER procedure applies a Newtonian iteration to determine a set of hk values for which pwk-(BHP+h_k) = 0 is satisfied in all completions, for a given value of BHP.

The application of this method requires the solution of dense linear systems with order equal to the number of active completions in the well; for wells with 50 or more completions this iteration may become excessively time-consuming.

The iterations may not converge; in this case the heads resulting from the first step of the iteration (those that would have been obtained without invoking *HEADITER) are used.

Example:

```
*HEADITER 'Well1' *ITERATIVE
```

Simultaneous BHP and Well Head Iteration (Optional)

***BHPHEADINIT**

PURPOSE:

*BHPHEADINIT specifies that certain wells are to have the bottom-hole pressure set to meet the well's rate or WHP target and simultaneously have the wellbore pressure differences between completions (heads) computed iteratively to achieve consistency between completion pressures and the computed heads. See also the manual page for the related keyword *HEADITER. The frequency with which this iteration is performed is determined by the keyword *WELLINIT. Please see the manual page for *WELLINIT for information on setting this frequency.

FORMAT:

*BHPHEADINIT (*well_list*) *SEPARATE | *COMBINED

DEFINITIONS:

well_list

A list of 'well_names' or well_numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*SEPARATE

The heads are set as directed by the *HEAD-METHOD and *HEADITER keywords at the beginning of each timestep and left fixed throughout the timestep; the BHP is set to meet the well's rate or WHP target with a frequency dictated by the *WELLINIT keyword. This is the default and was the only method available before the introduction of the *BHPHEADINIT keyword.

*COMBINED

A simultaneous iteration sets the BHP to meet the well's rate or WHP target and computes heads between well layers to achieve consistency between the calculated pressure drops in the wellbore and the completion pressures. This method may be useful in eliminating rate oscillations from timestep to timestep in backflowing wells with significant depth differences between completions and some liquid flow.

DEFAULTS:

For GEM, the default setting is *SEPARATE.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. If it is intended that the simultaneous iteration be applied at every Newtonian iteration, then *WELLINIT *ITER must be specified for the well in question.

EXPLANATION:

*BHPHEADINIT allows the user to direct that the well initialization done to set the BHP to a value meeting the well's rate or WHP target should simultaneously set the heads to achieve consistency between the completion pressures and the computed pressure differences between completions. Lack of consistency may be manifested as oscillations of non-controlled rates. *BHPHEADINIT may be necessary for backflowing wells with significant depth differences between completions and some liquid flow; the related keyword *HEADITER usually suffices for horizontal wells flowing mainly gas. Usually neither of these is needed for a non-backflowing well (or a well with a fixed pattern of backflowing layers) with significant depth differences between completions.

When *SEPARATE is in effect, a well's BHP is adjusted iteratively until T(calculated)-T(specified) = 0, where T is the value of the well's target rate or WHP. When *COMBINED is in effect, the BHP and completion heads (see the discussion in the manual page for *HEADITER) are iterated until T(calculated)-T(specified) = 0 and p_{wk}-(BHP+h_k) = 0 in all completions k.

The application of this method requires the solution of dense linear systems with order equal to the number of active completions in the well plus 1; for wells with 50 or more completions this iteration may become excessively time-consuming.

The iterations may not converge; in this case the BHP and heads resulting from the first step of the iteration are used.

Example:

```
*BHPHEADINIT 'Well1' *COMBINED
```

Perforations in Inactive Blocks (Optional) *NULL-PERF

PURPOSE:

*NULL-PERF specifies how perforations in inactive (null or pinched out) grid blocks are to be handled.

FORMAT:

*NULL-PERF (*well_list*) *STOP-SIM | *CLOSED

DEFINITIONS:

well_list
: A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*STOP-SIM
: If the simulator detects an attempt to perforate a named well in an inactive grid block, an error message identifying the well and block is printed and the simulation is terminated.

*CLOSED
: If the simulator detects an attempt to perforate a named well in an inactive grid block, a warning message is printed and the perforation is given CLOSED status. With CLOSED status, no fluids flow from or to the reservoir in the layer, but the layer is retained in the well and enters the well's head calculation.

DEFAULTS:

The default is *CLOSED.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword specifies the treatment of attempts to perforate a well in an inactive (null or pinched out) grid block. The attempt may either be considered an error, resulting in the termination of the simulation, or may result in the layer's being perforated with a status of *CLOSED. A layer with closed status is allowed to be the reference layer for the well (the layer in which the well's bottomhole pressure is defined).

The wellnum may be a single integer, a series of integers delimited by spaces or commas, or a range of integers defined in the form: imin:imax.

Example:

```
*WELL 1 'Producer1'
*WELL 2 'Injector1'
*WELL 3 'Prod. 54-233a'
*NULL-PERF 1:2 *CLOSED
*NULL-PERF 3 *STOP-SIM
```

Well Backflow Model (Optional) *XFLOW-MODEL

PURPOSE:

*XFLOW-MODEL is used to identify the method used to model well backflow and crossflow in a specified set of wells.

FORMAT:

*XFLOW-MODEL (*well_list*)
(*FULLY-MIXED (*LAGGED) | *ZERO-FLOW)

DEFINITIONS:

well_list
A list of well names or well numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*FULLY-MIXED
This keyword indicates that if backflow or crossflow occurs in the wellbore, it is modelled using an assumption of complete mixing in a wellbore of zero volume (cf. Coats et al, SPE 29111, 1995). *FULLY-MIXED and *ZERO-FLOW are mutually exclusive options and cannot be applied simultaneously to the same well.

*LAGGED
This sub-keyword of *FULLY-MIXED indicates that derivatives of source terms in a grid block into which fluid is injected from a crossflowing well with respect to variables in grid blocks surrounding the layers in which fluid flows into the same well are not calculated (i.e. are set to zero internally). This is the default and is currently the only treatment available. This subkeyword may be omitted.

*ZERO-FLOW
This keyword indicates that backflowing layers should be handled by setting layer flows to zero. The layer is not explicitly shut and flow resumes if pressures change so that the layer again flows normally. This treatment is the default. *FULLY-MIXED and *ZERO-FLOW may not be applied simultaneously to the same well.

DEFAULTS:

The default is *ZERO-FLOW for all wells. Use of *XFLOW-MODEL with no occurrence of *FULLY-MIXED or *ZERO-FLOW has the effect of setting *FULLY-MIXED *LAGGED for the wells listed. If *FULLY-MIXED is used with no following subkeyword, then *FULLY-MIXED *LAGGED is assumed.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword specifies the well backflow model to be used for the listed wells. Wells 1,2, and 4 are to use the fully-mixed crossflow, and well 3 is to use the zero-flow treatment.

Example:

```
*WELL 1 'Producer1'
*WELL 2 'Injector1' *VERT 12 14
*WELL 3 'Prod. 54-233a'
*WELL 4 'Pozo 4'
*XFLOW-MODEL 1:2 4 *FULLY-MIXED
*XFLOW-MODEL 'Prod. 54-233a' *ZERO-FLOW
```

Data for Workover Action for Wells (Optional)

***WORKOVER-DATA**

PURPOSE

*WORKOVER-DATA introduces the input of parameters which control the *WORKOVER action taken when a producing well violates a minimum rate monitored constraint which has *WORKOVER specified as the action to be taken in the event that the constraint is violated. The *WORKOVER action is specified using the *MONITOR keyword. The workover parameters which can be entered using *WORKOVER-DATA can also be entered after the *WORKOVER action subkeyword of the *MONITOR keyword; *WORKOVER-DATA is somewhat more convenient to use and is meant to supplant parameter entry after *WORKOVER, which however is still supported.

FORMAT:

*WORKOVER-DATA *well_list*
pi_factor ntimes wo_action

DEFINITIONS:

well_list
A set of 'well_names' or well_numbers to which this specification of *WORKOVER parameters applies; see below. No error is flagged if an injector is included in the well list, but the *WORKOVER-DATA entries can be used only for producers, since no *WORKOVER action is available for injectors.

well_names
Any number of well names (in quotes) to specify the wells to which this specification of *WORKOVER parameters applies. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

well_numbers
Any number of integers, or ranges of integers to specify the well numbers to which this specification of *WORKOVER parameters applies.

pi_factor
A real number specifying the factor by which all the layer productivities for the well are multiplied:

New_PI = Old_PI * pi_factor.

This value must appear on the first line after the *WORKOVER-DATA line.

ntimes
An integer specifying the maximum number of workovers allowed before wo_action is taken instead of the productivity index increment. This value must appear on the first line after the *WORKOVER-DATA line.

wo_action

Subkeyword specifying the action to be taken if the monitor is violated after the maximum number of workovers has been exhausted. wo_action can be one of the following: *STOP, *SHUTIN or *SHUTIN *REPEAT. The default is *SHUTIN. This subkeyword must appear on the first line after the *WORKOVER-DATA line.

*STOP

This subkeyword specifies that if a monitored constraint with the *WORKOVER action is violated, and the maximum number of workovers for the well has been reached, the simulation should be terminated.

*SHUTIN

This subkeyword specifies that if a monitored constraint with the *WORKOVER action is violated, and the maximum number of workovers for the well has been reached, the well should be shut and the simulation should advance to the next timestep. This is the default remedial action for a *WORKOVER constraint violation after the maximum number of workovers has been applied.

*REPEAT

This subkeyword of the *SHUTIN action specifies that if a monitored constraint with the *WORKOVER action is violated, and the maximum number of workovers for the well has been reached, the well should be shut and the timestep should be repeated.

DEFAULTS:

If a *WORKOVER action is specified for a violation of a monitored producer constraint under the *MONITOR keyword, but pi_factor, ntimes, and wo_action have not been specified for the well either after the *WORKOVER subkeyword or with the *WORKOVER-DATA keyword, the defaults are pi_factor = 1.2, ntimes = 10, wo_action = *SHUTIN.

If pi_factor, ntimes, and wo_action have been specified for a well and later new values are entered for the well either using *WORKOVER-DATA of the *WORKOVER subkeyword of *MONITOR, the later values replace the earlier.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group, and must follow the definition (via the *WELL keyword) of the wells listed in well_list.

Only a single set of pi_factor ntimes wo_action may be entered with a single *WORKOVER-DATA line; these will be applied to all wells in the list.

All well names or well numbers in well_list must appear on the same line as *WORKOVER-DATA; if more wells must be listed than can fit on the line, a second *WORKOVER-DATA line must be entered.

EXPLANATION:

The *WORKOVER action can be specified for violations of monitored minimum rate producing well constraints. Three parameters are used in applying a workover : the workover factor, the maximum number of workovers to be applied to the well, and the alternative

action to be taken if the constraint is violated but the well's maximum number of workovers has already been applied.

The number of workovers for a well is counted independently of the identity of the constraint whose violation has called for the workover. For example, if a producer has the workover action specified both for a *MIN *STO monitor and for a *MIN *STI monitor, then violations of either of these constraints increases the number of workovers applied to the well. However, no more than one workover is applied during a single timestep; for example, if a producer has *WORKOVER specified for both *MIN *STO and *MIN STI and both are violated during the same timestep, the productivity indices for the well's completions will be multiplied by pi_factor only once and the counter numbering the workovers applied to the well will be incremented only by one. However if the *MIN *STO constraint is violated in one timestep and the *MIN *STI constraint in the next, two workovers will be performed and counted toward the maximum ntimes.

Example:

```
*WORKOVER-DATA 'PROD*'
     1.3 12 *STOP
```

The above pair of lines assigns a productivity index factor of 1.3, a maximum number of workovers of 12, and an action after maximum workovers of termination of simulation for all wells whose names begin with the four characters PROD . The workover will only be applied if *WORKOVER is specified as a producer minimum rate monitored constraint violation action under the *MONITOR keyword.

Resetting Well Operating Constraint after Value Change (Optional) *MRC-RESET

PURPOSE:

*MRC-RESET allows the user to specify on a well-by- well basis that a well should or should not be set to its Most Restrictive Constraint after a change in the value of an operating constraint (through data read under the *OPERATE, *ALTER, or *TARGET keywords).

FORMAT:

*MRC-RESET (*well_list*) (*RESET | *NO-RESET)

DEFINITIONS:

well_list
: A list of 'well_names' or well_numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

RESET
: Specifies that a determination of the most restrictive well operating constraint should be done for all wells in *well_list* after a constraint value is changed through data and that the currently most restrictive constraint should be set as the current operating constraint for the wells before the next timestep is carried out.

NO-RESET
: Specifies that no operating constraint change should be made for the wells in *well_list* after a change in constraint values through data entry. Changes in operating constraint may occur after the next timestep as a result of constraint violations.

DEFAULTS:

Optional keyword. All wells are set in the *RESET mode at the beginning of the simulation by default; any changes entered under *MRC-RESET are cumulative. If no subkeyword appears after *well_list*, *RESET is assumed.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. If a well list is included in the *MRC-RESET line, then the *MRC-RESET line must follow all of the *WELL lines which define the wells in the list.

EXPLANATION:

When operating constraint values are changed the type of the most restrictive constraint may also change; for example, when a rate constraint value is increased, the bottomhole pressure

may become the most restrictive constraint. If no checking is done for this shift in the most restrictive constraint, the simulator must converge a timestep on the original constraint and let the constraint switch be accomplished as a result of constraint violation. If this next timestep must, for example, be run at a very high rate, converging the timestep may be quite difficult. To circumvent this potential difficulty, the default is to check all wells which have undergone a change in constraint value through use of the *OPERATE, *ALTER, or *TARGET keywords to determine the most restrictive constraint and set the well on this most restrictive constraint before the next timestep. The *NO-RESET option is provided if, for some reason, the user wishes to over-ride this automatic constraint switch.

Example: The sequence

```
*MRC-RESET   'SPECIAL-PRODUCER' *NO-RESET
*TARGET *STO  500.0
```

would have the oil rate constraint value of well SPECIAL-PRODUCER set to 500 units per day and the operating constraint would stay as it is (presumably on the oil rate constraint), regardless of whether the oil rate constraint is the most restrictive constraint at the beginning of a timestep.

Gas Lift Option (Optional) *GLIFT

PURPOSE:

*GLIFT specifies lift gas rates for a given set of production wells, or the method by which the lift gas rate is to be calculated.

FORMAT:

*GLIFT (*RATE | *GLR | *OPT) *well_list*
values

DEFINITIONS:

well_list

One or more quoted well names to specify the wells to which this setting of lift gas rates or composition applies. These names must be on the same line as the *GLIFT keyword. All listed wells must be producers. If more than one line is required for the well list, then the *GLIFT keyword must be repeated. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

*RATE

When this subkeyword is specified, gas lift injection rates must be specified directly. When all of *RATE, *GLR and *OPT are missing this is the default. For this option the units of 'values' are (m^3/day | SCF/day).

*GLR

When this subkeyword is specified, gas lift injection rates are calculated explicitly using produced gas liquid ratios from the previous timestep. For this option the units of 'values' are (m^3/m^3 | SCF/STB).

*OPT

When this subkeyword is specified, gas lift injection rates are calculated automatically according to the specifications listed under the *GLOPT keyword. An error will occur if the *OPT subkeyword is specified for a well and the *GLOPT keyword is missing. For this option 'values' are not required, but MAY be entered, in which case they are interpreted as maximum lift gas rates for the wells specified by well_names or well_numbers. The units of the values are (m^3/day | SCF/day) when used with *OPT. When no values are entered, very large values (1.0d+15) are assigned as the maximum lift gas rates for the listed wells.

values

One number for each well identified by *well_list* specifying the new value of the lift gas rate or gas-liquid ratio. Values must appear on one or more new lines immediately following the *GLIFT keyword. Values must NOT appear on the same line as the *GLIFT keyword. If the *OPT option is specified, then these values are not required, but may be entered as described above. See *RATE, *GLR, and *OPT above for the units of the values. If the

angular well fraction wfrac entered by the user under the *GEOMETRY keyword is less than one, it is assumed that lift gas rates also correspond to the well fraction values used; i.e., if the well fraction is 0.25, then the lift gas rate input by the user (or value obtained by the internal optimization algorithm) is 0.25 the rate for the whole well.

DEFAULTS:

Optional keyword. If the keyword is missing then gas lift rates are zero. The default sub-option is *RATE; if the well list follows *GLIFT immediately, values following the list are interpreted as lift gas rates.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It may appear anywhere in this keyword group following the *PRODUCER keyword(s) for the well_numbers. If *GLIFT is specified for a particular well then *PWELLBORE must also be specified for that well. All wells appearing in the well list must already have had their type defined with *PRODUCER or *CYCLPROD. If any listed well has not yet had its type defined, or is an injector, an error message is issued and the run is terminated. The lift-gas composition is required when *GLIFT is used with *PWELLBORE *MODEL. By default it will be taken from the field lumped separator gas from the previous time step. It can also be specified for each individual producer on lift with keyword *INCOMPWL.

EXPLANATION:

*GLIFT allows for the specification of a gas lift injection rate (or the specification of the method by which this rate is computed) for any producing well. These rates can be modified at different

*DATE or *TIME keywords by using the *GLIFT keyword again. The lift gas rate is added to the producer's formation gas rate before calculating the wellbore pressure drop and well liquid production rates. Each well's tubing hydraulics table specified by the *PWELLBORE keyword and entered using the *PTUBE1 keyword should be constructed by the user to include large enough ranges of gas-oil ratios to model the effects of gas lift operations.

Gas lift rates can be specified in three ways:

1. Input lift gas rates directly using the *GLIFT *RATE keywords.
2. Input a gas-liquid producing ratio using the *GLIFT *GLR keywords. The lift gas injection rates will be calculated from the current timestep's production data. However, the lift gas rates will not be used for the calculation of production rates until the next timestep.

   Lift_gas_inj = GLR_input * Total_liquid_est –
   GOR_est * Oil_rate_est
3. If *GLIFT *OPT is specified, gas injection rates will be calculated automatically using parameters specified using the *GLOPT keyword. The values entered under *GLIFT *OPT, if any, are maximum lift gas rates allowed for the wells.

When *GLIFT is in effect, printouts of total well and field gas rates and gas cumulatives do not include the lift gas. Individual well layer rates also do not include lift gas. A separate

printout of lift gas rates and cumulatives as well as field totals for all wells producing under gas lift is given.

Please note that gas rates and cumulatives output to the simulation output file (.OUT) do not include lift gas.

This option has the following assumptions:

1. The gas lift enters the production stream at or above the top reservoir grid block.
2. The wellbore gas flow rate is the sum of the formation gas flow rate and the lift gas flow rate.

Example:

```
*GLIFT 1  3:5  7
1000.0  500.0  700.0  600.0  1200.0
- or -
*GLIFT 1  3:5  7
2500.  1500.  1800.  900.  1600.
- or -
*GLIFT *OPT 1  3:5  7
```

SEE ALSO: *GLCONTROL, *GLOPT

Gas Lift Control (Optional) *GLCONTROL

PURPOSE:

*GLCONTROL specifies gas lift controls for a given set of production wells.

FORMAT:

*GLCONTROL *well_list*
(*WHP *val*) (*BHP *val*) (*WCUT *val*) (*STO *val*) (*TABLE *val*)

DEFINITIONS:

well_list

One or more quoted well names to specify the wells to which this setting of lift gas rates or composition applies. These names must be on the same line as the *GLCONTROL keyword. All listed wells must be producers. If more than one line is required for *well_list*, then the *GLCONTROL keyword must be repeated. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

*WHP

When this subkeyword is specified, the well will be put on gas lift automatically when the well-head pressure drops below 'val' (kPa | psi).

*BHP

When this subkeyword is specified, the well will be put on gas lift automatically when the bottom-hole pressure drops below 'val' (kPa | psi).

*WCUT

When this subkeyword is specified, the well will be put on gas lift automatically when the water-cut exceeds 'val' (fraction).

*STO

When this subkeyword is specified, the well will be put on gas lift automatically when the oil production rate drops below 'val' (m^3/day | Stb/day).

*TABLE

With the specification of this subkeyword gas lift calculations will be done using the specified wellbore hydraulics table number (entered under *PTUBE1), when the well is switched to gas lift automatically. If the well is assigned zero gas lift injection rates, then the *PTUBE1 table number to be used will revert to the table number specified with the *PWELLBORE keyword.

val

Specified minimum or maximum switching value, or *PTUBE1 table number.

DEFAULTS:

Optional keyword. If the keyword is missing the gas-lift wells will be put on gas lift immediately following the *GLIFT keyword. The default wellbore hydraulics table is the table number specified with *PWELLBORE.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It may appear anywhere in this keyword group following the *GLIFT keyword(s) for the well_numbers. If *GLCONTROL is specified for a particular well then *GLIFT and *PWELLBORE must also be specified for that well. All listed wells must have had their types defined with *PRODUCER or *CYCLPROD and must have been previously defined as gas lift wells with the *GLIFT keyword; otherwise, an error message will be issued and the run will be terminated.

EXPLANATION:

*GLCONTROL allows for the specification of operating constraints for particular gas lift wells which will be checked at every timestep for violation. If at any time one of the constraints is violated, the well will be switched to gas lift. If a wellbore hydraulics table number is specified using *GLCONTROL, then this table will be used for wellbore hydraulics calculations when the well is switched to gas lift. These constraints can be modified at different *DATE or *TIME keywords by using the *GLCONTROL keyword again.

Once a well begins receiving lift gas through a control violation, it continues to receive lift gas even if the control condition ceases to be violated, unless the limit is reset by re-entering the *GLCONTROL keyword.

No checking is done to ensure that the bottom-hole injection pressure is high enough to open the gas lift valves. It is left to the user to determine where the lift gas will be injected into the wellbore. Lift gas injection locations must be built into the tubing performance tables input by the *PTUBE1 keyword. A different table must be constructed for each gas lift valve elevation.

Example:

```
*GLCONTROL 1  3:5  7
*WHP 100.  *WCUT .75  *STO 10.  *TABLE 2
```

SEE ALSO: *GLIFT, *GLOPT

Gas Lift Optimization (Optional) *GLOPT

PURPOSE:

*GLOPT indicates the use of an option to optimize oil production by automatic allocation of lift gas.

FORMAT:

*GLOPT *gc_fact consum_rate*
(*MAXGAS *max_gas_rate*)
(*RECYCLE '*group_name*' *recyc_frac make_up_vol max_rate*)
(*GLOPT-TIME *time_interval*)

DEFINITIONS:

gc_fact

A minimum gas-cost factor, below which gas lift injection is uneconomic. This value is actually the derivative of the oil rate vs. lift gas injection. A realistic input value will ensure that large volumes of lift gas are not allocated for small incremental oil volumes (m^3/m^3 | STB/SCF).

consum_rate

The compressor gas consumption rate per unit gas injected (m^3/m^3 | SCF/SCF). This amount is subtracted from the lift gas available before the lift gas is apportioned among candidate producers.

*MAXGAS

This subkeyword specifies the maximum amount of gas available for gas lift injection. This total amount of gas will be automatically allocated to each well on gas lift production. If this value is very large and a realistic value of gc_fact has been input, then all of the available lift gas will not be used and the values reported for total lift gas will be the maximum optimized values for the field.

max_gas_rate

Total amount of gas available for gas lift injection (m^3/day | SCF/day).

*RECYCLE

When this subkeyword is specified, the total produced gas from the named group is considered available for gas lift injection in that group, subject to revision due to specified maximum gas rates, recycling fraction, and/or compressor consumption. The total produced gas is the gas produced from the reservoir plus the gas produced from previous gas lift injection averaged over the previous day (assuming that it takes one day to process and re-inject gas). The effect is cumulative, and if the compressor consumption is small and the recycle fraction is close to one, the available lift gas tends to increase with time.

'*group_name*'

Group name from which the gas to be injected is produced.

recyc_frac

Recycle fraction of gas production from a group of wells. The available lift gas for a group, computed as explained under *RECYCLE, is reduced by this factor.

make_up_vol

Make-up volume of gas available for injection from some source outside the field being modelled (m^3/day | SCF/day).

max_rate

Compressor maximum gas injection rate. Total available lift gas rate is internally not allowed to exceed this value (m^3/day | SCF/day).

*GLOPT-TIME

This subkeyword specifies the time interval between gas-lift optimization calculations. In the time interval specified by *GLOPT-TIME the allocated amounts of lift-gas remain at their previous values.

time_interval

The time interval between optimizations specified using the *GLOPT-TIME keyword. The default is 182.5 days (six months).

DEFAULTS:

Optional keyword. If the keyword is missing the gas lift rates must be specified on the *GLIFT keyword. If gc_fact is zero or missing, then the default value is 0.00005 STB/SCF, or 0.00028m^3/m^3. The default value for consum_rate is 0.0 SCF/day (m^3/day). The default gas availability is *MAXGAS 1.0E+20, and there is no default for the compressor maximum injection rates. The default *GLOPT-TIME is 182.5 days (six months).

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It may appear anywhere in this group after the relevant 'group_name' has been defined. It will only be operational for wells which are currently on gas lift and have been targeted for optimization using the *GLIFT *OPT option.

EXPLANATION:

When this option is specified, gas lift injection rates will be automatically optimized to provide maximum oil production rates for the wells specified. This optimization is an iterative procedure which uses tubing flow performance curves given by the *PTUBE1 keyword. The values in the *GLOPT option can be modified at different *DATE and *TIME keywords by using the *GLOPT keyword again.

The optimization begins with the determination of the total lift gas available (TLGA) for injection (total gas - compressor gas consumed). The TLGA may be input either directly using *MAXGAS, or indirectly from the field's gas production rates using *RECYCLE. In either case the user may specify that a compressor consumes some of the available gas. This gas consumption is directly proportional to the total lift gas injected (TLGI) which may be different from TLGA.

An iterative procedure is now started to determine the optimum lift gas allocations to each well. The TLGA is divided into equal "packets" of gas to be allocated iteratively.

The following equation is used to optimize oil production for each "packet" of lift gas allocated, and is summed over the entire range of TLGA:

$$\text{SUM_over_packets}\left(\frac{d(\text{oil rate})}{d(\text{gas inj rate})}\right)\text{max}$$

The preceding summation requires a derivative of the oil rate with respect to gas injected, which is calculated from the tubing performance tables (*PTUBE1). During this derivative calculation, wells which require a gas "kick off" rate to start oil production (derivative of zero) are assumed initially to start oil production immediately with lift gas injection. Then, if for the particular "packet" of lift gas allocation a well requiring a "kick off" gas rate turns out to possess the maximum derivative, then enough lift gas will be assigned to that well to "kick off" it's production.

If the derivative in the above equation is less than the minimum gas cost factor (gc_fact) then the gas "packet" will not be assigned to that well. The gas "packet" allocation process will end when either the TLGA is reached, or when each well has reached its minimum derivative specified by gc_fact.

Example:

```
*GLOPT 0  0.19
- or -
*GLOPT .00002  0.05
*RECYCLE 'group1' 0.97  1.0E+6  1.0E+8
- or -
*GLOPT 0  0 *MAXGAS 1.0E+7
```

Well Separators (Conditional) *WELSEP

PURPOSE:

*WELSEP specifies the separators through which the produced fluids are assigned surface stream compositions, molar rates, and volumetric rates.

FORMAT:

```
*WELSEP  wellnum  *STAGE      pstage      tstage
                                :           :
```

-or-

```
*WELSEP  wellnum
   *STAGE  stage_identification_lines
   (*WET-GAS  stage_number)
   *STREAM-DEN  stream_density_method_lines
```

where the stage_identification_lines (one set per stage) have the form

```
(*EOS eos_set_no pstage tstage)          output_destination_lines
( *PLANT plant_table_no )
```

and the output_destination_lines have the form

```
(*LIQUID-TO)          (stage_no)
(*MIDDLE-TO)          ('stream_name')
(*VAPOR-TO )
```

and the stream_density_method_lines have the form

```
'stream_name'         (*EOS eos_set_no pstream tstream (root))
                      (*KEY den_table_no)
                      (*MIX den_set_no)
                      (*GASLAW zfac pstream tstream)
```

DEFINITIONS:

wellnum

Any number of integers, or a range of integers, to specify the well numbers of the wells to which the current separator applies. Note that a list of well names can NOT be input here.

*STAGE(S)

This subkeyword indicates the input of phase-split information for the stages. Both *STAGE and *STAGES are accepted.

pstage

A real number specifying the stage pressure at which a two-phase EOS flash is done (kPa | psi).

tstage

A real number specifying the stage temperature at which a two-phase EOS flash is done (deg C | deg F).

*EOS

When appearing in a stage identification line, *EOS indicates that the stage is represented by a two- phase EOS flash at pressure and temperature pstage, tstage. When appearing in a stream density method line, *EOS indicates that the stream density is to be determined by a single-phase EOS density calculation at pressure and temperature pstream, tstream.

eos_set_no

The number of the equation-of-state parameter set to be used for the stage flash or for the stream density calculation. This is the parameter set number entered for the EOS parameter set using the *EOSSET keyword in the COMPONENT PROPERTIES section of the data set.

*PLANT

This subkeyword indicates that the fluid separation in the stage is to be represented by a table of component distribution fractions entered using the *PLANT-TABLE keyword.

plant_table_no

The number of the distribution fraction table entered using the *PLANT-TABLE keyword which is to be used to represent the fluid separation in the current stage.

*LIQUID-TO

Subkeyword introducing the destination of the liquid output of the current separator stage. This subkeyword is valid for both the *EOS and the *PLANT stage types.

*MIDDLE-TO

Subkeyword introducing the destination of the middle output of the current separator stage. This subkeyword is valid ONLY for the *PLANT stage type and not for the *EOS stage type.

*VAPOR-TO

Subkeyword introducing the destination of the vapor output of the current separator stage. This subkeyword is valid for both the *EOS and the *PLANT stage types.

*WET-GAS

The integer following *WET-GAS is the stage whose inlet stream (composition and rate) will be taken as the wet gas stream. *WET-GAS need not appear in the longer format separator description and must not appear in the shorter format. If *WET-GAS is not entered, the inlet stream to stage 1 is taken as the wet gas stream. When the first (shorter) format is used, the inlet stream to stage 1 is always the wet gas stream.

stage_no

The stage of the current separator to which the stage output whose destination is being specified is fed. stage_no must be larger than the stage number of the current stage. The stage numbering is the same as the order in which the stages are read in the data lines.

stage_number

Integer specifying a particular stage of the current separator. The stage numbering is the same as the order in which the stages are read in the data lines. The value must be greater than or equal to one and must not exceed the number of stages defined for the separator.

stream_name

As a stage outlet destination, this character string must be one of OIL, INL, or GAS. It specifies the surface stream which is the destination of the particular output of the current stage, or the stream for which the density calculation is being specified. When used for the density calculation specification, WTG is also accepted as the stream_name. INL stands for the intermediate liquid stream, WTG for the wet gas stream.

pstream

When *EOS or *GASLAW is specified as the surface stream mass density calculation method, the single-phase EOS or gas law density calculation is done at pressure pstream (kPa | psi).

tstream

When *EOS or *GASLAW is specified as the surface stream mass density calculation method, a single-phase EOS or gas law density calculation is done at temperature tstream (deg C | deg F).

root

Optional subkeyword of *STREAM-DEN *EOS specifying which root of the cubic EOS is used to compute the stream density. Root may be *LIQUID, *VAPOR, or *GIBBS. When *LIQUID is specified the smaller compressibility root is used; when *VAPOR is specified the larger; and when *GIBBS is specified the root giving the lower molar Gibbs free energy is chosen. The default is *LIQUID for the OIL and INL streams and *VAPOR for the GAS and WTG streams.

zfac

When the *GASLAW density calculation method is used, the relation PV = ZRT is applied, and mass_density = molecular_weight*P/ZRT. zfac (dimensionless) gives the value of the compressibility factor Z to be used in the density calculations.

*KEY

This subkeyword specifies that the named surface stream will have its mass density interpolated as a function of a key-component mole fraction from a table entered using the *DENSTR-KEY keyword.

*MIX

This subkeyword specifies that the named surface stream will have its mass density computed with a mixing rule for molar volume using component mass densities entered using the *DENSTR-MIX keyword.

*GASLAW

This subkeyword specifies that the named surface stream will have its mass density computed using the gas law relation PV = ZRT, with mass_density = molecular_weight*P/ZRT. Here Z is a user-entered compressibility factor, P is a user-entered pressure, and T is the absolute equivalent of a user-entered temperature.

den_table_no

The number of the density table entered using the *DENSTR-KEY keyword which is to be used in the mass density calculation for the current stream.

den_set_no

The number of the component mass density set entered using the *DENSTR-MIX keyword which is to be used in the mass density calculation for the current stream.

DEFAULTS:

Conditional keyword. The default separator conditions are those specified by *SEPARATOR in the INITIAL CONDITION keyword group. The WELSEP lines must be entered in one of the two formats shown above. If the second (longer) form is chosen, all of the indicated subkeywords, numbers, and/or character strings must be entered or a violation will be identified by the input routines and the run will be terminated.

When the first (shorter) form is used, the stages are all of the EOS type. The destination of the liquid output of stage n is the feed of stage n+1 or the OIL stream if stage n is the last stage; the destination of the vapor output of all stages is the GAS stream. The inlet to stage 1 is the WTG (wet gas) stream. The OIL, GAS, and WTG streams are the only surface streams when the shorter form is used, with a single-phase EOS mass density calculation done for OIL and GAS streams at the pressure and temperature of the last stage using the last (highest index) set of EOS parameters entered. The WTG density by default is calculated by *GASLAW with zfac = 1 at the same pressure and temperature.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

Each producer must be connected to one separator. The default separator conditions are those specified by *SEPARATOR in the INITIAL CONDITION keyword group.

For injectors, separator specification is not required. A single-phase EOS calculation is always used to relate injector surface rates to injected molar rates.

*SEPARATOR is used to calculate the original fluid-in-place during initialization and *WELSEP is used to calculate the individual surface well rates during the simulation.

Note that if *WELSEP for all producers are not the same as *SEPARATOR, there will be inconsistencies between the calculated recoveries and the original fluid-in-place.

Example 1:

```
*WELSEP 1:3,5 *STAGES 500.0  100.0
                      101.3   60.0
```

Wells 1,2,3 and 5 are connected to a two stage separator. Note that the stock tank conditions are assumed to be those specified by the last separator stage. The inlet stream to stage 1 is by default the wet gas stream

Example 2:

The following two sets of lines specify the same separator for the same wells, the first set using the first (shorter) form and the second the longer form.

```
*WELSEP 1:3 *STAGE 100.    120.
                    14.69   60.
-----------------------------------------------------
*WELSEP 1:3 *STAGE *EOS 2  100.   120. *LIQUID-TO 2
                                       *VAPOR-TO  'GAS'
                    *EOS 2   14.69  60. *LIQUID-TO 'OIL'
                                       *VAPOR-TO  'GAS'
    *STREAM-DEN 'GAS' *EOS 2  14.69  60.
                'OIL' *EOS 2  14.69  60.
```

In the second set the *EOS after *STAGE specifies that the separation in the first stage is to be done by a two-phase EOS flash using EOS parameter set 2 at pressure 100 psi and temperature 120 degrees Fahrenheit, with the liquid from the flash feeding the second stage and the vapor going directly to the GAS surface stream. *EOS on the third line indicates that the separator's second stage uses an EOS flash with parameter set 2 at pressure 14.69 psi and temperature 60 degrees Fahrenheit, with the liquid from that flash going to the OIL surface stream and the vapor going to the GAS surface stream. The *STREAM-DEN keyword indicates (1) that the separator has only two stages, since neither *EOS nor *PLANT was encountered; and (2) that both the GAS and OIL streams are to have their mass densities determined by a single-phase EOS calculation using parameter set 2 at 14.69 psi and 60 deg F.

Example 3:

Two-stage separator with one EOS stage and one plant stage :

```
*WELSEP 1 *STAGES *EOS 2 150. 120. *LIQUID-TO 'OIL'
                                   *VAPOR-TO 2
                  *PLANT 1         *LIQUID-TO 'OIL'
                                   *MIDDLE-TO 'INL'
                                   *VAPOR-TO 'GAS'
*WET-GAS 2
  *STREAM-DEN  'OIL'  *KEY 1
               'INL'  *MIX 1
               'GAS'  *MIX 2
               'WTG'  *GASLAW 1.0  14.69  60.
```

These lines specify that the separator applies to well 1, that stage 1 uses EOS parameter set number 2 in a two-phase flash at 150. psi and 120. deg F., that the liquid output from this flash goes directly to the OIL surface stream, and that the vapor output becomes the feed to stage 2. Stage 2 uses distribution fractions in plant table number one to determine the fluid split; the liquid output from this split goes to the OIL stream, the middle output to the INL (intermediate liquid) stream, and the vapor output of the split to the GAS surface stream. The mass density of the OIL surface stream is interpolated as a function of a key-component mole fraction from stream density table number 1; the INL stream has its density computed using a mixing rule with component mass density set number 1, and the GAS stream has its mass density computed using a mixing rule with component mass density set number two. The inlet to stage 2 (the plant

stage) is interpreted as the wet gas stream. The wet gas mass density is computed using PV = ZRT with Z = 1.0, P = 14.69 psi, and T the Rankine equivalent of 60 degrees F.

Example 4:

A four-stage separator with 3 EOS stages and a single plant stage :

```
*WELSEP 15:20 *STAGE *EOS  2  150.   120. *LIQUID-TO 2
                                          *VAPOR-TO  3
                     *EOS  3   30.    80. *LIQUID-TO 'OIL'
                                          *VAPOR-TO   3
                     *EOS  4   14.69  60. *LIQUID-TO 'INL'
                                          *VAPOR-TO   4
                     *PLANT 1             *LIQUID-TO 'OIL'
                                          *MIDDLE-TO 'INL'
                                          *VAPOR-TO 'GAS'
*WET-GAS 4
*STREAM-DEN 'OIL'  *EOS  3  14.69  60.
            'INL'  *KEY  1
            'GAS'  *MIX  1
            'WTG' *GASLAW 1.0  14.69  60.
```

The inlet stream to the plant stage is interpreted as the wet gas stream.

Specification of Non-EOS Stream Density Calculations (Optional) *DENSTR-KEY, *DENSTR-MIX

PURPOSE

*DENSTR-KEY and *DENSTR-MIX introduce tables and component density sets respectively which allow the computation of the mass density of surface streams of separators without reference to a set of EOS parameters. These keywords allow entry of the tables and component mass density sets which are referred to under the *STREAM-DEN subkeyword of the *SEPARATOR and the *WELSEP keywords.

FORMAT:

```
*DENSTR-KEY  table_no
        *DENUNITS          (*LIQUID)
                           (*GAS)
        *KEYF  kf  *KEYL  kl
        *KEYFRAC y1 ... yn
        *STRDEN d1 ... dn
and

*DENSTR-MIX set_no
        *DENUNITS          (*LIQUID)
                           (*GAS)
        *COMP-DEN d(1) ... d(nc)
```

DEFINITIONS:

table_no

Integer which designates the table being entered as the table_no-th stream density table. This integer will in turn be used in data lines introduced by the *STREAM-DEN subkeyword of the *SEPARATOR and *WELSEP keywords to associate a separator surface stream with a method to compute its mass density. The tables must be read consecutively; e.g. if a table is read after table 4, its table_no must be 5 or a data error will be detected.

*DENUNITS

Subkeyword which introduces another subkeyword which specifies whether, when field units are used, the entered density values are expressed in lbm/ft3 or in lbm/bbl . If *DENUNITS does not appear or *DENUNITS appears with no following subkeyword, lbm/bbl is assumed. If SI units are used, then this subkeyword has no effect, and may be omitted.

*LIQUID

Subkeyword of *DENUNITS which specifies that, if field units are used, the entered density values are expressed in the unit system which is appropriate for liquid surface streams, namely lbm/bbl.

*OIL and *INL are also accepted in place of *LIQUID. Specifying *LIQUID under *DENUNITS does not restrict the use of the table or component density set to liquid surface streams; for a gas stream the appropriate unit conversion is done.

*GAS

Subkeyword of *DENUNITS which specifies that, if field units are used, the entered density values are expressed in the unit system which is appropriate for gas surface streams, namely lbm/ft3. *SOLVENT is also accepted in place of *GAS. Specifying *GAS under *DENUNITS does not restrict the use of the table or component density set to gas surface streams; for a liquid stream the appropriate unit conversion is done.

*KEYF

Subkeyword to introduce the hydrocarbon component number of the first component which is part of the "key component", which is in general actually a set of consecutively numbered components.

kf

Integer index of the first hydrocarbon component which is included in the "key component".

*KEYL

Subkeyword to introduce the hydrocarbon component number of the last component which is included in the "key component".

kl

Integer index of the last hydrocarbon component which is included in the "key component". kl must be greater than or equal to kf and the mole fractions of components kf, ..., kl in the plant feed are summed to obtain the key component mole fraction.

*KEYFRAC

Subkeyword which introduces the set of n values of the key component mole fraction to which the following input line of stream mass densities applies. There is no explicit upper limit to the integer n, since the tables are stored consecutively in a singly-subscripted array.

y1 ... yn

The n values of the key component mole fraction for which stream mass density values will be entered on the next data line. y1 ... yn must be a monotonically increasing list of dimensionless real numbers whose values are between zero and one. GEM determines the value of n by counting the number of real numbers which follow *KEYFRAC before *STRDEN is found. If n = 1 the stream mass density is independent of the key component mole fraction. If no mole fraction values are entered (n = 0) an error is detected and the run is terminated.

*STRDEN

Subkeyword which introduces the set of n values of the stream mass density.

d1 ... dn

The n values of stream mass density in (kg/m3 | lbm/bbl) or (kg/m3 | lbm/ft3) depending on what is entered under or defaulted for the *DENUNITS keyword as explained above. These values correspond in order to the values y1 ... yn entered on the preceding line. These values must be non-negative real numbers. GEM determines how many values of the stream mass density are entered by counting the real numbers entered after *STRDEN and before the next keyword; if this number is not equal to the n determined in reading y1 ... yn, an error is generated.

set_no

Integer which designates the component mass density set being entered as the set_no-th set. This integer will in turn be used in data lines introduced by the *STREAM-DEN subkeyword of the *SEPARATOR and *WELSEP keywords to associate a separator stream with a method to compute its mass density. The component mass density sets must be read consecutively; e.g. if a set is read after set 4, its set_no must be 5 or a data error will be generated. These component mass density sets are used ONLY for the mass density calculations for the designated streams; they do not affect any EOS calculations or reservoir transport equations of any kind.

*COMP-DEN

Subkeyword which introduces the set of nc component mass densities, where nc is the number of hydrocarbon components entered in the FLUID PROPERTIES keyword section.

d(1) ... d(nc)

The set of nc values of component mass density, expressed in (kg/m3 | lbm/ft3) or (kg/m3 | lbm/ft3) as set or defaulted under the *DENUNITS subkeyword as explained above. These values are used as described in the EXPLANATION section below to compute the separator stream mass density as a function of composition. These must be positive real values. If a zero is entered an error is generated.

DEFAULTS:

If table_no is not entered after *DENSTR-KEY, the current last table index is incremented by one and the incremented index is used. Similar remarks hold for *DENSTR-MIX and the set_no.

CONDITIONS:

This keyword may appear in either the INITIAL CONDITIONS keyword group or the WELL AND RECURRENT DATA keyword group. A density table or component density set cannot be modified once it has been read; a user wanting to change the density table or the component mass density set used by a given separator stream must enter a new table or set with number exceeding the currently last table or set number by one and redefine the stream density calculation to refer to the new table or set.

EXPLANATION:

The *STREAM-DEN subkeyword of the *SEPARATOR and *WELSEP keywords allows three different methods to be specified for computing the mass density of a separator's surface streams : (1) Single-phase EOS calculation, (2) interpolation in a density table as a function of a key-component mole fraction, or (3) from a mixing rule averaging component molar volumes which are computed from user-entered component mass densities.

Method (1) uses the EOS parameters entered as described in the FLUID PROPERTIES section of this manual and is not described further here.

When method (2) is specified, the mole fractions of components kf through kl in the separator stream are summed to give the stream key-component mole fraction ky (the molar rates of all components in the stream are assumed to be known). If there are consecutive tabular key-component mole fractions yi and yi+1 such that yi < ky < yi+1, then the density is interpolated linearly as

$$\text{density} = \frac{(\text{ky} - \text{yi})}{(\text{yi}+1 - \text{yi})} * \text{di}+1 + \frac{(\text{yi}+1 - \text{ky})}{(\text{yi}+1 - \text{yi})} * \text{di}.$$

If ky < y1 or ky > dn, no extrapolation is done and the stream mass density is set to d1 or dn, respectively. Data for the construction of these tables can be had either from laboratory chemical analysis and density measurements of the separator streams or from calculations done using an EOS phase behavior package, if a characterization of the fluid is available.

When method (3) is specified, the user-entered component mass densities are used to compute component molar volumes v(m) (m3/gmol | ft3/gmol),

v(m) = M(m)/d(m),

where M(m) is the molecular weight of component m (kg/gmol | lbm/gmol) and d(m) is the user-entered component mass density. The stream molar volume is then computed from a linear mixing rule,

V(stream) = Sum(m = 1,nc) x(m)*v(m) ,

where x(m) is the mole fraction of component m in the separator stream, and the mass density D(stream) (kg/m3 | lbm/ft3) is then calculated as

D(stream) = M(stream)/V(stream)

where M(stream) is the average molecular weight of the stream,

M(stream) = Sum(m = 1,nc) x(m)*M(m).

Unit conversion to barrels as the volume unit are then done if necessary.

A simple source of approximate component mass density data is to determine, either from laboratory data or from calculations done with a phase behavior package, the molar density of the stream in question (mol/m3 | mol/ft3) and then to assume that all of the components in the mixture have that same molar density. The mass density of each component is then found by multiplying this molar density by the molecular weight (kg/mol | lbm/mol) of the component. A final conversion to account for volume units may be necessary if the densities are to be input in units of lbm/bbl.

Example 1:

```
*DENSTR-KEY 3
   *KEYF 5 *KEYL 6

**              y1      y2      y3
   *KEYFRAC   0.0     0.05    0.1
**              d1      d2      d3
   *STRDEN   320.3   321.1   322.2
```

The mole fractions y1, y2, y3 are to be compared with sums of the mole fractions of the fifth and sixth hydrocarbon components. The densities d1, d2, and d3 are in lbm/bbl ; the table could be used for OIL or INL streams, both of which are measured in liquid units. This table could not be used for a GAS stream, as the units are not correct.

Example 2:

```
*DENSTR-MIX 2
**               C1    C2   C3-C4   C5-C6   C7-C9   C10+
   *COMP-DEN   7.47  14.0   23.8    36.9    53.2   105.5
```

If field units are used, the component mass densities in the above are expressed in lbm/ft3. If the resulting mixing rule is used for a GAS stream, the density values are returned directly with the units lbm/ft3; if the mixing rule is used for an OIL or INL stream, the density value is converted to lbm/bbl.

Table Defining Gas Plant (Optional) *PLANT-TABLE

PURPOSE:

*PLANT-TABLE introduces a table of distribution fractions which determine the amounts and compositions of the fluid outputs of a gas plant used in determining initial volumes of fluids at surface conditions in place in the reservoir.

A plant table in general consists of several sub-tables, with each sub-table containing distribution fractions for a different value of a key-component mole fraction. The plant is assumed to have three output streams, namely liquid, vapor, and middle. If fewer streams are wanted, the user should enter zeros as distribution fractions for the unwanted output for all components.

FORMAT:

```
*PLANT-TABLE  table_no
        *KEYF  kf  *KEYL  kl
        *KEYFRAC  key_component_mole_fraction
           Repeated distribution_fraction_lines
```

where the distribution_fraction_lines (one for each hydrocarbon component) have the form :

```
liquid_fraction  vapor_fraction middle_fraction
```

The m-th distribution_fraction_line for a given value of key_component_mole_fraction gives the distribution fractions for component m, m = 1, ... , nc, where nc is the number of hydrocarbon components entered in the FLUID PROPERTIES section.

DEFINITIONS:

table_no

Integer which designates the table being entered as the table_no-th plant table. This integer will in turn be used in data lines introduced by the *SEPARATOR and *WELSEP keywords to associate a separator stage with a table of distribution fractions. The tables must be read consecutively; e.g. if a table is read after table 4, its table_no must be 5 or a data error will be detected.

*KEYF

Subkeyword to introduce the hydrocarbon component number of the first component which is part of the "key component", which is in general actually a set of consecutively numbered components.

kf

Integer index of the first hydrocarbon component which is included in the "key component".

*KEYL

Subkeyword to introduce the hydrocarbon component number of the last component which is included in the "key component".

kl

Integer index of the last hydrocarbon component which is included in the "key component". kl must be greater than or equal to kf and the mole fractions of components kf, ..., kl in the plant feed are summed to obtain the key component mole fraction.

*KEYFRAC

Subkeyword which introduces the value of the key component mole fraction to which the following input lines of distribution fractions apply.

key_component_mole_fraction

A real number between zero and one inclusive which gives the value of the sum of the mole fractions of components kf through kl in the plant feed for which the following lines of distribution fractions hold exactly. In a given occurrence of the *PLANT-TABLE keyword, the key_component_mole_fractions must be entered in monotonically increasing order.

liquid_fraction

Real number which gives the mole (equivalently the mass) fraction of the current component which exits in the liquid output stream of the gas plant. The liquid_fraction, middle_fraction, and vapor_fraction for each component are normalized internally to sum to one. The value entered may be zero but must not be negative.

vapor_fraction

Real number which gives the mole (equivalently the mass) fraction of the current component which exits in the vapor output stream of the gas plant. The liquid_fraction, middle_fraction, and vapor_fraction for each component are normalized internally to sum to one. The value entered may be zero but must not be negative.

middle_fraction

Real number which gives the mole (equivalently the mass) fraction of the current component which exits in the middle output stream of the gas plant. The liquid_fraction, middle_fraction, and vapor_fraction for each component are normalized internally to sum to one. The value entered may be zero but must not be negative.

DEFAULTS:

If table_no is not entered after *PLANT-TABLE, the current last table index in incremented by one and the incremented index is used. If zeros are entered for all of liquid_fraction, middle_fraction, and vapor_fraction on one line of a plant table, then liquid_fraction = middle_fraction = vapor_fraction = 1/3 is defaulted.

CONDITIONS:

This keyword may appear in either the INITIAL CONDITIONS keyword group or the WELL AND RECURRENT DATA keyword group. A table cannot be modified once it has been read; a user wanting to change the plant table used by a given separator must enter a new table with number exceeding the currently last table number by one and redefine the separator to refer to the new table.

EXPLANATION:

*PLANT can be specified as the stage type for a surface separator defined either with the *SEPARATOR keyword in the INITIAL CONDITIONS section or the *WELSEP keyword in the WELL AND RECURRENT DATA section. In either case *PLANT is followed by a table number in which are to be found the distribution fractions which are to be used to describe the gas plant stage. These tables are entered by the user with the *PLANT-TABLE keyword.

The gas plant has a given feed composition and total molar rate, from which can be calculated feed molar rates for each component. The output molar rate of component m in the liquid gas plant output is then computed as

liquid_molar_rate(m) = feed_molar_rate(m)* liquid_fraction(m) ;

the expressions for the molar rate of component m in the other two outputs (middle and vapor) are exactly analogous. The liquid output is intended nominally to correspond to an oil, the middle to a natural gas liquid, and the vapor to a light gas output, but there is no restriction as to the use of these names and the user has free choice of the distribution fractions used for the outputs.

If the sum kx of the mole fractions of the components included in the "key component" lies between two of the tabular values the distribution fractions are determined by linear interpolation between the two bracketing mole fraction values k1 < kx < k2 :

$$\mathrm{frac}(\mathrm{kx}) = \frac{(\mathrm{kx}-\mathrm{k1})}{(\mathrm{k2}-\mathrm{k1})} * \mathrm{frac}(\mathrm{k2}) + \frac{(\mathrm{k2}-\mathrm{kx})}{(\mathrm{k2}-\mathrm{k1})} * \mathrm{frac}(\mathrm{k1}),$$

where frac stands for liquid_fraction, middle_fraction, or vapor_fraction. Note that this interpolation preserves the property that the fractions sum to one. If kx is smaller than the smallest key mole fraction entered after *KEYFRAC, then the distribution fractions for the smallest entered mole fraction are used without extrapolation. Similarly no extrapolation is done if the key component mole fraction in the feed exceeds the largest key component mole fraction entered after *KEYFRAC.

A possible source for the distribution fraction values is detailed EOS simulations of the gas plant using a phase behavior package. Such a study would be preliminary to constructing the GEM data set for the reservoir simulation.

Example:

```
*PLANT-TABLE 2
    *KEYF 5  *KEYL 6

    *KEYFRAC 0.0
** liq_frac vap_frac mid_frac
      0.1      0.7      0.2     ** Component 1
      0.12     0.6      0.28    ** Component 2
      0.17     0.4      0.43    ** Component 3
      0.3      0.25     0.45    ** Component 4
      0.5      0.1      0.4     ** Component 5
      0.9      0.01     0.09    ** Component 6
```

```
    *KEYFRAC 0.05
** liq_frac vap_frac mid_frac
      0.1      0.7      0.2     ** Component 1
      0.15     0.55     0.3     ** Component 2
      0.23     0.27     0.50    ** Component 3
      0.35     0.10     0.55    ** Component 4
      0.60     0.05     0.35    ** Component 5
      0.92     0.01     0.07    ** Component 6
```

All of the lines in the table above have liq_frac + mid_frac + vap_frac = 1; had this not held for some line, the liquid fraction would have been replaced internally with the normalized quantity

$$\text{liq_frac_new} = \frac{\text{liq_frac}}{(\text{liq_frac} + \text{mid_frac} + \text{vap_frac})}$$

and analogously for mid_frac_new and vap_frac_new. GEM concludes that exactly two different values of the key component mole fraction have been entered in this table when it encounters a keyword other than *KEYFRAC after the nc-th line following *KEYFRAC 0.05.

Standard Pressure, Temperature, and EOS Set (Optional)

***INJECTOR-P, *INJECTOR-T, *INJECTOR-EOS**

PURPOSE:

*INJECTOR-P and *INJECTOR-T specify the standard pressure and temperature used internally in GEM for converting surface rates to molar rates for hydrocarbon injection wells through a single-phase EOS calculation.

FORMAT:

```
*INJECTOR-P  pstd
*INJECTOR-T  tstd
*INJECTOR-EOS  istd
```

DEFINITIONS:

pstd
: The standard pressure (kPa | psi).

tstd
: The standard temperature (deg C | deg F).

istd
: The EOS set number to be used with pstd, tstd. This is the EOS set number entered using the *EOSSET keyword in the COMPONENT PROPERTIES section of the data set.

DEFAULTS:

Initially GEM internally sets pstd = 14.7 psia or 101.325 kPa and tstd = 60.0 deg F. or 15.56 deg C by default, and sets istd to the last (largest) EOS set entered.

If *SEPARATOR appears in the initial conditions data section AND the first (shorter) format is used for the *SEPARATOR data, or the second format is used and the last stage of the initialization separator is of the EOS type, then pstd is set to the pressure of the last stage of the initialization separator, tstd is set to the temperature of the last stage of the initialization separator, and istd is set to the EOS set number used for the last stage of the initialization separator. When the shorter *SEPARATOR format is used istd is left unchanged at the last (largest) EOS set number.

These settings can be overridden by using the *INJECTOR-P, *INJECTOR-T, and *INJECTOR-EOS keywords.

CONDITIONS:

These keywords may appear in either the INITIAL CONDITIONS keyword group or the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

The quantities pstd, tstd, and istd are used in the following ways :

1. For computing relative oil and gas volumes in initialization. This is a reported number only and does not affect calculation amounts in place.
2. For converting between surface and reservoir rates for injection wells (no separators are specified for injectors).
3. For converting between surface and molar rates for injection wells.
4. For converting between make-up gas surface and molar rates in gas recycling.

pstd and tstd are not necessarily used in computing the surface rates of producing wells; please see the explanations of the *STREAM-DEN subkeyword of the *WELSEP for more information on how producer surface rates are calculated. The intended principal use of these keywords is in setting injector conditions other than 14.7 psi and 60 deg F. when the last stage of the initialization separator is not of EOS type.

Example:

```
*INJECTOR-P 100.
*INJECTOR-T  15.
*INJECTOR-EOS 3
```

If SI units are used, this sets the standard pressure to 100.0 kPa and the standard temperature to 15.0 degrees Celsius. EOS set number 3 will be used for injection rate calculations.

Well Element Geometry (Optional) *GEOMETRY

PURPOSE:

*GEOMETRY specifies the well geometric characteristics to be used by the simulator to calculate the well index internally.

FORMAT:

*GEOMETRY (*I | *J | *K) *rw geofac wfrac skin*

DEFINITIONS:

*I, *J, *K
: This subkeyword specifies that the wellbore is parallel to the I, J or K axis.

rw
: A real number specifying the wellbore radius. This should be the distance from the center of the wellbore at which fluid flows from the surrounding grid block into the well. (m | ft)

geofac
: A real, dimensionless number specifying the geometric factor for the well element. This factor depends upon the placement of the well within the grid block and upon the placement of the grid block relative to the boundaries of the reservoir. Values for *geofac* are given in Figure A-1 in Appendix A of this manual. In that figure, *geofac* is referred to as "CC". This value is not used when the well effective radius is calculated using the Peaceman formula, although it is still required on the *GEOMETRY line.

wfrac
: A real number between 0. and 1. specifying the fraction of a circle that the well models. (Usually one; set to one half or one quarter if the well is at the edge or corner of the grid block on the grid boundary.) Dimensionless.

skin
: A real number specifying the well skin factor (dimensionless).

DEFAULTS:

Optional keyword. If *GEOMETRY is absent then *rw* = 0.0762 m (0.25 ft) and *skin* = 0, and *geofac* = 1 and *wfrac* = 1 correspond to a full well in the middle of a block. The default wellbore direction is parallel to the K axis.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

It must come before any of the well completion keywords that will calculate well index internally. All subsequent well completion keywords will use the last *GEOMETRY line to calculate the well index (unless *WI is specified by keyword *PERF for all the well perforations).

EXPLANATION:

The parameters entered under *GEOMETRY are used to compute flow rates of phases into wells as follows. The formula applied to compute the molar flow rate into the well in a particular layer for each phase is

$$q = wi \cdot \lambda \cdot (P_{block} - P_{well})$$

$$wi = 2\pi \cdot ff \cdot kh \cdot wfrac \,/\, [\ln(re/rw) + skin]$$

where λ is the phase mobility and P the pressures. *kh* is either directly entered by the user or computed as the average permeability in the plane normal to the wellbore times the block thickness in the direction of the wellbore. The well index is multiplied by a partial completion factor *ff* entered by *GEO(A). It can be used to account for partial completions through a grid block.

The wellbore radius (*rw*) and skin factor (*skin*) can be overridden on a layer basis by the *PERF or *PERFV keyword.

The well effective radius (*re*), unless given directly by the user under the *PERF or *PERFV keyword (see the manual page for the *PERF keyword), will be calculated as follows when *GEO or *KH is in force for the well index calculation:

For a vertical well (direction = K):

$$re = geofac \cdot [\ DX \cdot DY \,/\, (\pi \cdot wfrac)\ ]^{1/2}$$

For a horizontal well (direction = J):

$$re = geofac \cdot [\ DX \cdot DZ \,/\, (\pi \cdot wfrac)\ ]^{1/2}$$

For a horizontal well (direction = I):

$$re = geofac \cdot [\ DY \cdot DZ \,/\, (\pi \cdot wfrac)\ ]^{1/2}$$

where DX, DY and DZ are the *x*, *y* and *z* dimensions of the grid block in which this particular well layer is completed.

When *GEOA or *KHA is in force for the well index calculation, *re* is computed as follows without using *geofac* (Peaceman D.W., “Interpretation of Well-Block Pressures in Numerical Reservoir Simulation with Non-Square Grid Blocks and Anisotropic Permeability”, SPEJ, June 1983, pp. 531-543):

For a vertical well (direction = K):

$$re = 0.28\ (DX^2\, ky + DY^2\, kx)^{1/2} \,/\, (kx^{1/2} + ky^{1/2})$$

For a horizontal well (direction = J):

$$re = 0.28\ (DX^2\, kz + DZ^2\, kx)^{1/2} \,/\, (kx^{1/2} + kz^{1/2})$$

For a horizontal well (direction = I):

$$re = 0.28\ (DY^2\, kz + DZ^2\, ky)^{1/2} \,/\, (ky^{1/2} + kz^{1/2})$$

The Peaceman formula automatically takes account of anisotropy in permeability and of non-unit aspect ratios of grid block dimensions.

If *GEO is used, then *rw* must be equal to the well radius entered with *REFINE *HYBRID. Since the well is assumed to be in the centre of the grid block, *geofac* is assumed to be 0.50.

Hybrid grids are used to model near-well effects such as breakthrough more accurately, especially in producers.

The well radius (*rw*) entered under *GEOMETRY also serves as a default tubing radius for wellbore pressure drop calculations if none is entered under *PWELLBORE or *IWELLBORE.

For example, to define wells 1 and 2 with identical well geometry and a third well 3, with a larger radius, enter:

```
*WELL 1 'Producer'  ** define 4 wells
*WELL 2
*WELL 3
...

** Well geometry for well 1 and well 2
**             rw    geofac  wfrac  skin
*GEOMETRY *K  0.375  0.2488   1.0   0.0

*PERF *GEO 1        ** Use last *GEOMETRY (rw=0.375)
   2 2 4     1.     ** well 1 perforation
....

*PERF *GEO 2        ** Use last *GEOMETRY (rw=0.375)
   4 2 2     1.     ** well 2 perforation
...

** Well geometry for well 3
**             rw    geofac  wfrac  skin
*GEOMETRY *K   0.5   0.2488   1.0   0.0

*PERF *GEO 3        ** Use last *GEOMETRY (rw=0.5)
   1 1 4     1.     ** well 3 perforation
```

Location of Well Completions (Conditional) *PERF, *PERFV

PURPOSE:

*PERF specifies the location of the well completion grid blocks, well indices or parameters for well index calculations.

*PERFV is a short form of *PERF for vertical wells, where the horizontal locations have been specified by the *VERT subkeyword of the *WELL keyword.

FORMAT:

*PERF / *PERFV (*index_kywd*) *wn*
{locat.} {well index} ({rel. perm.}) ({status}) ({connection})
: : : : :

where
index_kywd = (*WI | *SLIMTUBE) (*DIR) (*GEO | *GEOA) (*KH | *KHA)
(*RE) (*RW) (*SKIN) (*TURB) (*QUAD | *PSEUDOP)

DEFINITIONS:

{locat.}

i_f j_f k_f (/ …(/ i_{rn} j_{rn} k_{rn})…)

{well index}

((*wi*) | (*dir*) | (*ff*) | (*kh*) | (*re*) | (*rw*) | (*skin*) | (*di*))

{rel. perm.}

(*setn*) (*swcon*) (*sgcrit*) (*sorw*) (*sorg*)

{status}

(*OPEN | *AUTO | *CLOSED)

{connection}

(*FLOW-TO *ily* | *FLOW-FROM *ily*) (*REFLAYER)

wn

For *PERF, it must be a single well sequence number or single well name enclosed in single quotes. For *PERFV, it can be a range of well sequence numbers or a list of well names with a limited wildcard facility. *wn* must be on the same line as the perforation keyword.

{locat.}

i_f j_f k_f (/ …(/ i_{rn} j_{rn} k_{rn})…)

(i_f, j_f, k_f) are integers or integer ranges specifying the grid block index of the fundamental grid in the I, J, K direction (User Block Address), respectively, through which the well is completed through. (See explanation). For *PERFV, since i_f and j_f are already defined by the subkeyword *VERT of the *WELL keyword, only k_f is accepted.

Optional (i_{rn}, j_{rn}, k_{rn}) are similar to (i_f, j_f, k_f), but at the *n*'th level of refinement (if applicable) for the perforated grid block. Not applicable to *PERFV. At most one direction may have a range.

*WI - *wi*

Optional sub-keyword indicating that the corresponding column of {well index} is the well indices (*wi*) input directly for a particular well layer. If none of the subkeywords *WI, *GEO(A), *KH(A), *RE, *RW, *SKIN or *TURB is present then *WI is the default. If $wi > 0$ is set for a completion, the input of other {well index} columns, if specified, does not have any effect for this perforation. Production and injection for the layer for each phase at reservoir conditions is found from:

$$q = wi \cdot \lambda \cdot (P_{block} - P_{well})$$

where the units for *wi* are (md-m | md-ft).

For production the mobility (λ) is a particular phase mobility and phase rates are summed to give total layer rates; for injection mobility is the sum of the water, gas and oil mobilities.

wi must be a positive number and if *wi* = '–' (a dash) is set for a particular completion, its well index will be calculated internally:

$$wi = 2\pi \cdot ff \cdot kh \cdot wfrac \,/\, [\ln(re/rw) + skin]$$

as explained below and also in the manual page for the *GEOMETRY keyword. It is allowable to specify well indices for certain perforations ($wi > 0$) and the other perforations to have the well indices calculated according to the radial flow model (*wi* = '–') within the same well. All the variables required for calculating the well indices (*ff*, *kh*, *re*, *rw*, *skin*, *di*), if not specified directly by the user, will be obtained either from the defaults or calculated internally.

*DIR - *dir*

Optional sub-keyword indicating that the corresponding column of {well index} is the layer direction (*dir*) for internal well index calculations. Perforations are parallel to one of the local coordinate axes (I, J, K) but can vary from layer to layer. *dir* is an integer ranging from [1 -3] where I = 1, J = 2 and K = 3. If *DIR is absent or specified as *dir* = '–' for a particular layer, the direction specified by *GEOMETRY prevails. An equivalent way of specifying varying perforation directions along local axes is to use *LAYERIJK.

*GEO(A) - *ff*

Optional sub-keyword indicating that the corresponding column of {well index} is the partial completion factor (*ff*) for internal well index calculations. *GEO differs from *GEOA in the formula used for the well effective radius (if *RE is absent or specified as *re* = '–'), as explained in the manual page for the *GEOMETRY keyword. *GEOA uses the Peaceman formula. *GEO and *GEOA are mutually exclusive for the same well and have the same effect if

re > 0 is specified for a layer. Defaulted to 1, *ff* is dimensionless and must be a positive number.

*KH(A) - *kh*

Optional sub-keyword indicating that the corresponding column of {well index} is the *kh* variable for internal well index calculations. *kh* is defined as the average permeability in the plane normal to the wellbore times the block thickness in the wellbore direction. *KH and *KHA differ in the formula used for the well effective radius (if *RE is absent or specified as *re* = '–'). *KHA uses the Peaceman formula. *KH and *KHA are mutually exclusive for the same well and have the same effect if *re* > 0 is specified for a layer.

The units for *kh* are (md-m | md-ft). *kh* must be a positive number and if *kh* = '–' is set for a particular layer, it will be calculated internally.

*RE - *re*

Optional sub-keyword indicating that the corresponding column of {well index} is the well effective radius (m | ft). If *RE is absent, or is specified for a well but set *re* = '–' for a particular layer, the effective radius for this layer will be calculated according to one of the formula as described in the *GEOMETRY keyword. *re* must be positive.

*RW - *rw*

Optional sub-keyword indicating that the corresponding column of {well index} is the wellbore radius (m | ft). *rw* is taken from the value specified by *GEOMETRY for the entire well if *RW is absent. *rw* must be a positive number.

*SKIN - *skin*

Optional sub-keyword indicating that the corresponding column of {well index} is the skin factor (dimensionless). *skin* is taken from the value specified by *GEOMETRY for the entire well if *SKIN is absent. *skin* can be any numbers.

*TURB - *di*

Optional sub-keyword indicating that the corresponding column of {well index} is the turbulent skin factors to be included when calculating well indices for gas phase flow into or out of the wellbore in each of the well's completions. The effect modelled is the non-Darcy flow occurring at high gas velocities as described by Forchheimer's equation, in which the pressure gradient is set equal to an expression with both linear and quadratic terms in velocity, rather than the linear term alone as in Darcy flow. The effect is modelled by adding a velocity-dependent term to the constant mechanical skin term. A positive value for the turbulent D-factor (*di*) must be entered in order to apply this option. *di* is defaulted to 0 and takes effect only when the well indices are calculated internally.

The entered value of *di* (day/m^3 | day/ft^3) is multiplied internally by the molar volume of an ideal gas at surface conditions (q_g) and then added to the constant skin term for the computation of the molar gas flow rate:

$skin' = skin + di \cdot q_g$

*SLIMTUBE - *wi*

Optional sub-keyword indicating that the run represents a slim tube with injection at one end and production from the other. The direction of the slim tube should be either the I or the J direction. A special well index calculation is performed, giving well index

$wi' = wi \cdot geofac \cdot wfrac \cdot K \cdot area$

where *geofac* and *wfrac* are the well geometric parameter and angular completion fraction as defined in the manual entry for *GEOMETRY, *wi* here is the dimensionless partial perforation factor (*ff*) usually set to 1, *K* is the perforated grid block permeability in the direction of the slimtube flow (for example, the I-direction permeability if the slimtube direction is *I), and *area* is the grid block area perpendicular to the flow, for example DJ·DK for an I-direction tube. The *SLIMTUBE option avoids some difficulties which arise in well index calculations due to the small grid block size in slimtubes relative to usual well dimensions and with differences in flow direction relative to normal wells.

*SLIMTUBE and *WI cannot coexist for the same well since they take the same column of {well index} table for input (the *wi* column).

*QUAD

Optional sub-keyword indicating that an expression quadratic in block and wellbore pressure is to be used in calculating gas phase flow rates into or out of the wellbore in each of the well's completions. *QUAD does not take any column of the {well index} table. The effect modelled is the change in gas properties as the pressure changes from the grid block value to the wellbore value. The molar gas flow rate is modelled as

$q = wi \cdot K_{rg} \cdot (\rho/\mu) \cdot (P_{block} - P_{well})$

The default calculation uses the gas viscosity (μ) and molar density (ρ) values applying at the grid block pressure P_{block}. When *QUAD is in force the grid block viscosity is still used but the gas molar density is approximated as

$\rho = 0.5\ \rho_{block} \cdot (P_{block} + P_{well}) / P_{block}$

yielding an expression for q proportional to $P^2_{block} - P^2_{well}$.

*QUAD takes affect only when the well indices are calculated internally.

*PSEUDOP

Optional sub-keyword indicating that the variation in both gas molar density and gas viscosity with pressure is to be taken into account in computing the gas flow rate into or out of the wellbore (see discussion under *QUAD above). *PSEUDOP does not take any column of the {well index} table. The expression used for the molar gas flow rate when *PSEUDOP is in force is

$q = wi \cdot K_{rg} \cdot (\rho/\mu)_{ave} \cdot (P_{block} - P_{well})$

with the average value for the quotient of gas molar density (ρ) and gas viscosity (μ) computed as

$$(\rho/\mu)_{ave} = 0.5\ [(\rho/\mu)_{well} + (\rho/\mu)_{block}]$$

with ρ evaluated directly from the equation of state at both pressures and the viscosities evaluated directly from the correlation at both pressures. *PSEUDOP takes effect only when the well indices are not specified by the user. *QUAD and *PSEUDOP are mutually exclusive for the same well.

setn

Relative permeability set to use when calculating the phase mobility in the well equation. If not present, the relative permeability for the grid block will be used.

This option can be used for partial completions where relative permeability endpoints may be different. For example, if a producer is completed only in the top quarter of the grid block, then the water will have to come up to the well (let's say $Sw = 0.67$) before water becomes mobile in the well.

If *setn* is used it must be the 1st column of the {rel. perm.} table.

swcon

Connate water saturation of the well layer to scale the relative permeability table given by *setn*. *swcon* corresponds to the first table entry in the *SWT table. If *swcon* is not present then the *swcon* of the relative permeability table number *setn* will be used.

If *swcon* is used it must be the 2nd column of the {rel. perm.} table.

sgcrit

Critical gas saturation of the well layer to scale the relative permeability table given by *setn*. *sgcrit* corresponds to the table entry in the *SGT table where Krg is zero. If *sgcrit* is not present then the *sgcrit* of the relative permeability table number *setn* will be used.

If *sgcrit* is used it must be the 3rd column of the {rel. perm.} table.

sorw

Residual oil saturation of the well layer to scale the relative permeability table given by *setn* for the water-oil relative permeability. *sorw* corresponds to the table entry in the *SWT table where Krow is zero. If *sorw* is not present then the *sorw* of the relative permeability table number *setn* will be used.

If *sorw* is used it must be the 4th column of the {rel. perm.} table.

sorg

Residual oil saturation of the well layer to scale the relative permeability table given by *setn* for the gas-oil relative permeability. *sorg* corresponds to the table entry in the *SGT or *SLT table where Krog is zero. If *sorg* is not present then the *sorg* of the relative permeability table number *setn* will be used.

If *sorg* is used it must be the 5th column of the {rel. perm.} table.

*OPEN

This subkeyword specifies that the well layer is open (perforated). This is the default.

*AUTO

This subkeyword specifies that the well layer is currently plugged but is a candidate for automatic recompletion. This automatic recompletion is specified by using the *RECOMPLETE action of the *MONITOR keyword.

*CLOSED

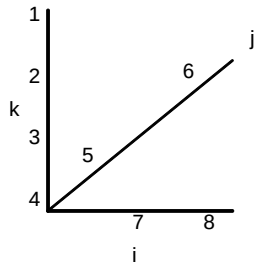
This subkeyword specifies that the well layer is a geometrical node point for the purpose of defining the well trajectory. The layer will remain plugged unless overridden by a new *PERF / *PERFV card selecting the layer status at that location as *OPEN or *AUTO. A closed layer may be the reference layer for a well. The pressure drop through a well segment with closed status is accounted for in computing the wellbore pressure differences between completions. No fluid flows from wellbore to reservoir or from reservoir to wellbore in a closed layer.

*FLOW-TO

This subkeyword is used to establish a "child to parent" relationship between the current layer and a layer ALREADY specified under the current *PERF card for a producer. For the first layer specified under the *PERF the parent is the surface. This default is applied if this keyword is missing for the first layer. For subsequent layers the default is the previous layer specified in the sequence. The index used to reference a given layer for the "child to parent" relationship is simply the count of the number of well layers entered to that point that the parent layer is specified under the current *PERF card. The range specification operator: can still appear, but only once per layer card. These points are best illustrated with an example of a dual lateral well with legs or "branches" parallel to the i and j axis:

```
*PERF *GEO 1
** i  j   k   ff   status   connection
   1  1  1:3  1.0           FLOW-TO 'SURFACE' ** 1-3
   1  1    4  1.0 *CLOSED   FLOW-TO     3     ** 4
   1  2    4  1.0           FLOW-TO     4     ** 5
   1  3    4  1.0           FLOW-TO     5     ** 6
   2  1    4  1.0           FLOW-TO     4     ** 7
   3  1    4  1.0           FLOW-TO     7     ** 8
```

This corresponds to the following geometry, assuming *KDIR *DOWN



The first line under the *PERF specifies layers 1-3. If the *FLOW-TO keyword was not specified for this line then *FLOW-TO 'SURFACE' would have been defaulted. The range in k direction is interpreted as well layer 1 connected to 2. Well layer 2 connected to 3. Layer flow is connected to layer 3. In general if the *FLOW-TO keyword is missing for layer N+1, then *FLOW-TO N is assumed. Therefore the *FLOW-TO keyword for layer 4 is not strictly required. Note also that layer 4 is assigned the status of *CLOSED. This node is defined simply for geometrical convenience, there can be no flow from it. In this case by default the reference layer for BHP calculation is the first layer. However any of the layers 1-8 could have been designated as reference layer with the keyword *REFLAYER.

ily

The index of the parent layer used to establish a "child to parent" relationship. The parent layer must already have been defined under the current *PERF card to be referenced. The character string 'SURFACE' can be used in place of *ily* for the first layer listed under *PERF.

*FLOW-FROM

This subkeyword is used to establish a "child to parent" relationship between the current layer and a layer ALREADY specified under the current *PERF card for an injector. Please see documentation under *FLOW-TO above. *FLOW-TO is also accepted for injectors with exactly the same effect that *FLOW-FROM would have in the same place.

*REFLAYER

By default the first layer specified under the current *PERF card is used to measure the wellbore flowing pressure. The user may select a different layer for this purpose by the use of this keyword. This keyword can only appear once per *PERF card.

DEFAULTS:

Defaults for subkeywords as given above. Each well must have at least one open perforation defined for it.

For entering the {well index} table, the user may put a dash ('–') at where a number is expected to indicate that the corresponding value will be taken from the defaults: internal calculation for *wi*, *kh*, *re*; *ff* = 1, *di* = 0; *dir*, *skin* and *rw* from *GEOMETRY.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

One of *PERF or *PERFV is required to apply to each well. The {connection} subkeywords *FLOW-TO, *FLOW-FROM and *REFLAYER are not applicable to *PERFV.

For a given well, the first layer defined on the most recent *PERF(V) line is the layer at which the well's bottom-hole pressure is calculated, and the layer from which the pressure drop to the surface is calculated when well head pressures are evaluated. If a layer other than that flowing to the surface is to be the reference layer, then the *PERF keyword should be

used, whose syntax is sufficiently flexible to allow separate specifications of the reference layer and the layer flowing to the surface.

The {rel. perm.} table specifies the relative permeability table set numbers and the available end points for well completions. Please refer to keyword *KRPERF for the full relative permeability data input for well completions.

For perforating deviated layers (see *LAYERXYZ), *wi* > 0 should not be specified for any deviated layer, and *kh* > 0 and *re* > 0 should not be specified for the same deviated layer at the same time. It is recommended to specify *GEO(A) only.

EXPLANATION:

The *PERF keyword specifies the grid blocks in which a well is completed. It can be used for horizontal or deviated wells, where the completions are not in a single vertical column of grid blocks, and can also be used for vertical wells.

The *PERFV keyword specifies the grid blocks in which a vertical well is completed. If several vertical wells are all completed in the same set of layers, they may all be defined with one *PERFV keyword.

The subkeywords of {well index} defines the input combination and sequence of the {well index} table for the well index calculations on a layer basis. The number of columns of {well index} table is in the range of [1 – 8]. The Peaceman formula for *re* will be adopted if either *GEOA or *KHA (or both) is specified.

If the perforation location integers identify an inactive grid block (for example, a null or pinched-out block), the user can choose one of two possible actions:

1. The perforation can be completed with a status of CLOSED, in which case the perforation will be included for the purposes of head calculation but no fluid will flow in the completion; or
2. The completion can be rejected by the simulator with a message identifying the suspect perforation so that data may be modified.

Action (1) is the default. Action (2) can be specified with the *NULL-PERF keyword.

Examples for *PERF:

```
*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
...
**            rw    geofac  wfrac  skin
*GEOMETRY *K 0.375  0.2488  1.0    0.0
*PERF 1   ** (default – *WI)
** if  jf   kf    wi
   16   8    4   1.56
17:20   8    6  12.40
   21   8    6    -    ** Internal wi with all defaults
*PERF *GEO 2
** if  jf   kf  ff  (setn)
   12   6  2:4  1.0    1
   13   6  5    0.5    2
```

```
*PERF *GEO *QUAD *TURB  1  ** No column for *QUAD
** if  jf   kf   ff   di       (setn)
   12   6   2:4  1.0  0.00002    1
   13   6   5    0.5  0.00004    2
*PERF *KH 'Producer 1'
** if  jf   kf   kh   (setn)
   12   6   2:4  100.    1
   13   6   5    500.    2
*PERF *KH *TURB *PSEUDOP  1  ** No column for *PSEUDOP
** if  jf   kf   kh   di
   12   6   2:4  1.0  0.00002
   13   6   5    0.5  0.00004
*PERF *SKIN *RE *KHA   *WI  2
7 5 3     -   -    - 362.8
7 4 3    15  18    -     -       ** default indicator (-)
7 3 3    13   - 1113     -
7 2 3    12   -    -     -
```

In the example right above, 4 columns of the {well index} table are defined for well #2:

The well index (*wi*) is directly supplied by the user for layer 7 5 3;

Well indices will be calculated internally for the rest of layers with *dir* = 3 (K), *rw* = 0.375 (from *GEOMETRY), *ff* = 1 (default - full completion) and *di* = 0 (default – no turbulence effect for gas flow):

layer 7 4 3: user-specified *skin* = 15 and *re* = 18; calculate *kh*;
layer 7 3 3: user-specified *skin* = 13 and *kh* = 1113; calculate *re* (Peaceman);
layer 7 2 3: user-specified *skin* = 12; calculate both *kh* and *re* (Peaceman).

Examples for *PERFV:

```
*WELL 1 'Producer 1' *VERT 12 16
*WELL 2 'Producer 2' *VERT 10  5
*WELL 3 'Producer 3' *VERT 21  3
*WELL 4 'Producer 4' *VERT 17 12
...
**            rw    geofac  wfrac  skin
*GEOMETRY *K 0.375  0.2488   1.0   0.0

*PERFV 1 4  ** (default - *WI)
** kf   wi
   2:4  1.56
   5    1.1
*PERFV *GEO 2:3
** kf   ff  (setn)
   2:4  1.0   1
   5    0.5   2
*PERFV *KH *TURB 'Producer 4'
** kf   kh   di     (setn)
   2:4  100. 0.0002   1
   5    500. 0.0004   2
```

Auto Thresholding for Well indices (Optional) *WIRANGE

PURPOSE:

*WIRANGE specifies the range of the constant geometric part of the well indices (WI) and takes proper actions if the well indices are out of the range.

FORMAT:

*WIRANGE (*well_list*)

type *v*1 *v*2

where *type* is any one of

*RANGE1 | *MEAN1 | *MEDIAN1 |
*RANGE2 | *MEAN2 | *MEDIAN2 | *OFF

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. *WIRANGE will apply to ALL wells if *well_list* is not input.

*RANGE1/2

This optional subkeyword specifies that the well indices WI must be within the range given directly by the two following values ($v1$, $v2$), $wi \in [v1, v2]$. For any wi that is above the maximum value ($v2$), it will be assigned to this value. For any wi that is below the minimum value ($v1$), it will be assigned to either this value (*RANGE1) or zero (*RANGE2). Please see *PERF *WI for the corresponding units required for $v1$ and $v2$.

*MEAN1/2

This optional subkeyword specifies that the well indices WI must be within the range calculated from the two following values ($v1$, $v2$), $wi \in [wi_m \cdot v1, wi_m \cdot v2]$, where wi_m is the arithmetic mean of the wi for all the well opening perforations. For any wi that is above the maximum value ($wi_m \cdot v2$), it will be assigned to this value. For any wi that is below the minimum value ($wi_m \cdot v1$), it will be assigned to either this value (*MEAN1) or zero (*MEAN2). $v1$ and $v2$ are fractions (dimensionless) in this case.

*MEDIAN1/2

Same as the above for *MEAN1/2 except that wi_m is the median of the wi for all the perforations.

*OFF

Turn off the well indices range regulation. $v1$ are $v2$ are not required and should not be input.

DEFAULTS:

Optional keyword. Default is not to add auto-thresholding to well indices.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. All wells appearing in the well list following *WIRANGE must already have been defined in *WELL lines. *v*1 and *v*2 are non-negative numbers and *v*2 cannot be smaller than *v*1. They must be present unless the subkeyword is *OFF. If *WIRANGE is present then any one of the subkywords *RANGE1…, or *OFF must immediately follow.

EXPLANATION:

Models that have a well index (direct user input or internally calculated) too large or too small may experience irregular rate and/or pressure behaviors causing difficulty in convergence. The extreme well indices are usually attributed to the abnormal permeabilities in the perforated well blocks or some tiny corners that a well trajectory cuts through the grid blocks. *WIRANGE regulates the well indices for each perforation so that all the active perforations will have the well indices within the desired interval. WI will be assigned to the maximum allowed value if it is exceeded. If WI is below the minimum allowed value, the user may decide to either trim the value to the minimum or simply close this perforation by giving $wi = 0$.

The well indices and well index statistics for all or selected wells may be output using keyword *WELPRN to help in specification of *WRANGE parameters. Please see the keyword description for further details.

Example:

```
DATE 1990 1 1
  *WELL 1 'Producer1'
  *WELL 2 'Producer2'
  *WELL 3 'Producer3'
  ......

**For well 1 & 2, wi will be cut-off if it is below 20%
**of the averaged wi or above 200% of the averaged wi.
  *WIRANGE 'Producer1'  'Producer2'
    *MEAN1  0.2  2.0

** Layers with wi below 10% of the median will be closed
  *WIRANGE 'Producer3'
    *MEDIAN2  0.1  1000
```

Well Geometry Parameters (Optional) *WELGEO

PURPOSE:

*WELGEO specifies the geometric characteristics for listed wells to be used by the simulator to calculate the well index internally.

FORMAT:

*WELGEO (*well_list*)
(*DIR (*I | *J | *K))
(*RAD *rw*)
(*GEOFAC *geofac*)
(*WFRAC *wfrac*)
(*SKIN *skin*)

DEFINITIONS:

well_list
One or more well numbers or quoted well names assigned with the *WELL keyword. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. *WELGEO will apply to ALL wells if *well_list* is not input.

*DIR
This optional subkeyword specifies that the wellbore is parallel to the I axis (*I), the J axis (*J) or the K axis (*K).

*RAD
If present, this subkeyword defines a positive real number *rw* for the wellbore radius. This should be the distance from the center of the wellbore at which fluid flows from the surrounding grid block into the well (m | ft). It also serves as a default tubing radius for wellbore pressure drop calculations if none is entered under *PWELLBORE or *IWELLBORE.

*GEOFAC
If present, this subkeyword defines a positive real (dimensionless) number *geofac* for the geometric factor for well element. This factor depends upon the placement of the well within the grid block and upon the placement of the grid block relative to the boundaries of the reservoir. *geofac* is ignored when the effective radius (*re*) is calculated using the Peaceman formula.

*WFRAC
If present, this subkeyword defines a positive real (dimensionless) number *wfrac* between 0 and 1 for the fraction of a circle that the well models. Usually one; set to one half or one quarter if the well is at the edge or corner of the grid block on the grid boundary.

*SKIN
If present, this subkeyword defines a real, dimensionless number *skin* for the well skin factor.

DEFAULTS:

Optional keyword. Default is to use the well parameters defined by keyword *GEOMETRY and assigned to wells through the subsequent perforation keywords (*PERF or *PERFV). If *GEOMETRY is also absent then *rw* = 0.0762 m (0.25 ft) and *skin* = 0.0, and *geofac* = 1.0 and *wfrac* = 1.0 correspond to a full well in the middle of a block. The default wellbore direction is parallel to the K axis.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Any conditions that apply to *GEOMETRY also apply to *WELGEO.

For listed wells, *WELGEO *DIR will override any previously defined layer-varying deviation directions by *PERF *DIR (or *LAYERIJK) or *LAYERXYZ; *WELGEO *RAD will override any previously defined layer-varying wellbore radius by *PERF *RW; *WELGEO *SKIN will override any previously defined layer-varying skin factors by *PERF *SKIN.

EXPLANATION:

Keyword *WELGEO provides an alternative way of setting the well element geometry parameters for a list of wells. The definitions and usages of these parameters are the same as those specified by keyword *GEOMETRY.

Values specified via *GEOMETRY are global and held in reserve, until overridden by another *GEOMETRY keyword. At the time a well is perforated via keyword *PERF or *PERV, it uses the current *GEOMETRY parameters to calculate well index.

In contrast, parameters specified by keyword *WELGEO take immediate effect for the listed wells. This action lets you override the *GEOMETRY parameters for specified wells at specified times. If a well is re-perforated at an even later time (after the *WELGEO keyword appears), the well index calculation will use parameters from *GEOMETRY again. Therefore, an override that you want to persist must be re-specified.

Example:

```
DATE 1990 1 1
  *WELL 1 'Producer1'
  *Well 2 'Producer2'
  *WELL 3 'Producer3'
  ......
  *PERF *GEO 1      **use default geometry for well 1
    2 2 4  1.
  *GEOMETRY *K 0.375 0.2488 1.0 0.0
  *PERF *GEO 2      **use values from *GEOMETRY for well 2
    4 2 2  1.
  *PERF *GEO 3
    1 1 4  1.       **use values from *GEOMETRY for well 3
*DATE 1901 1 1
  *WELGEO 1 2
     *SKIN -1.0     **Alter skin factor for well 1, 2
*DATE 1902 1 1
  *WELGEO *SKIN -1.2   **Alter skin factor for all wells
```

Geometric Data for Deviated Well Completions (Conditional)

***LAYERXYZ, *LAYERXYZ-METHOD**

PURPOSE:

*LAYERXYZ allows the user to supply geometric information specifying deviated perforations – perforations in which the wellbore direction is not parallel to one of the local coordinate axes. *LAYERXYZ-METHOD allows the user to switch the interpolation schemes for radius and permeability calculation along the deviated wellbore.

FORMAT:

```
*LAYERXYZ  'wname'
{locat.}          {deviated layer info.}
   :                        :
*LAYERXYZ-METHOD   (*DIRECT | *OLD)
```

DEFINITIONS:

wname

Single well name within single quotes specifying the well to which the following deviated layer specifications apply. No wild-carding is supported in this situation.

{locat.}

$i_f \;\; j_f \;\; k_f$ (/ …(/ $i_{rn} \;\; j_{rn} \;\; k_{rn}$)…)

i_f, j_f, k_f are integers or integer ranges of the fundamental grid block index in the I, J and K direction (User Block Address) specifying the layers of well '*wname*' to be treated as deviated layers. It is valid to name some of the well's layers and not others in the *LAYERXYZ lines, but layers not mentioned do not acquire the deviated status and will be treated as *I, *J, or *K perforations. Any layer named under *LAYERXYZ must already have been defined for the well using a *PERF or *PERFV statement.

Optional (i_{rn}, j_{rn}, k_{rn}) are similar to (i_f, j_f, k_f), but at the *n*'th level of refinement (if applicable) for the perforated grid block. At most one direction may have a range.

{deviated layer info.}

($x_1 \; y_1 \; z_1 \;\; x_2 \; y_2 \; z_2 \;\; h$ | *UNDEVIATED)

$x_1 \; y_1 \; z_1$

Coordinates of the "entry point" of the wellbore into the perforated grid block. (m | ft). See explanation.

$x_2 \; y_2 \; z_2$

Coordinates of the "exit point" of the wellbore from the perforated grid block. (m | ft). See explanation.

h

Perforated length of the deviated wellbore within the named grid block (m | ft). See explanation.

*UNDEVIATED

Subkeyword indicating that the layer is henceforth to be treated as an undeviated layer with the direction (*I,*J, or *K) which the layer had when initially perforated.

*DIRECT, *OLD

Subkeyword indicating which interpolation scheme is used for calculating effective wellbore radius and permeability of a deviated well layer. It applies to all wells having *LAYERXYZ. See EXPLANATION below.

*DIRECT	Directional weighting method (default, recommended).
*OLD	Cardinal weighting method. Before version 2011.10 this was the only method available.

DEFAULTS:

Layers named under *LAYERXYZ with geometric information (as opposed to the *UNDEVIATED subkeyword) are flagged as deviated within the simulator; undeviated is the default status assigned at the beginning of the run. The undeviated status can be re-imposed through the *UNDEVIATED subkeyword.

CONDITIONS:

The named layers must all have been previously created for well '*wname*' with *PERF or *PERFV lines. Unless the layer is designated as *UNDEVIATED, *WI ($wi > 0$) should not be specified for any deviated layer, and *KH(A) ($kh > 0$) and *RE ($re > 0$) should not be specified for the same deviated layer at the same time. It is recommended to specify *GEO(A) only. If geometric information is supplied, all seven of the real numbers $\{x_1, y_1, z_1, x_2, y_2, z_2, h\}$ must be specified or an error is generated. Not all of a well's layers need to be named under *LAYERXYZ; those omitted are treated as undeviated according to the original *PERF specification.

EXPLANATION:

This keyword specifies geometric information which allow the well indices to be calculated for perforations in which the wellbore direction does not parallel one of the grid block coordinate axes. The point coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2) must be expressed in the Cartesian coordinate system underlying the simulation grid. The entry point and exit point specify the direction of the wellbore; the perforation length (h) gives the actual perforation length through the named grid block. It is valid for h to exceed the distance between (x_1, y_1, z_1) and (x_2, y_2, z_2) in order to allow more freedom in matching well indices. However the distance between (x_1, y_1, z_1) and (x_2, y_2, z_2) must be positive; if (x_1, y_1, z_1) and (x_2, y_2, z_2) are the same point an error is generated. The identities of entry and exit points can be exchanged with no difference in the well index calculation. The *ff* factor entered under the *PERF or *PERFV line when the layer was created still apply to the deviated layer. If the user intends that h alone control the well index calculation, *ff* should be entered as 1.0 on the *PERF line for the layer.

The deviated well index is calculated as

$$wi = 2\pi \cdot ff \cdot wfrac \cdot K \cdot h \,/\, [\ln(re/rw) + skin]$$

The well angular fraction *wfrac*, the completion factor *ff*, well radius *rw* and skin term *skin* are entered on the relevant *GEOMETRY or *PERF lines. *h* is read directly in the *LAYERXYZ data.

The drainage radius *re*, if not supplied by the user on the *PERF lines, is computed from the information entered under *LAYERXYZ as follows when *GEO or *KH is in force. The projection of *re* along the D axis (D = I, J, or K) is calculated as

$$re(\mathrm{D}) = geofac \cdot [V \,/\, (\pi \cdot xh(\mathrm{D}) \cdot wfrac)]^{1/2},$$

where $V = xh(\mathrm{I}) \cdot xh(\mathrm{J}) \cdot xh(\mathrm{K})$ with $xh(\mathrm{D})$ being the grid block thickness in the direction D, and *geofac* is the geometric factor entered with the *GEOMETRY keyword. Please see keyword *GEOMETRY for *re*(D) calculations using the Peaceman formula when *GEOA or *KHA is in force.

Once *re*(D) has been calculated for the three directions (I, J, K), an interpolation to the deviated wellbore direction is done as follows. Let $\mathbf{u} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$ be a unit vector in the wellbore direction. It makes no difference whether **u** points with or against the fluid flow in the wellbore. Let $\mathbf{i} = (i_x, i_y, i_z)$, $\mathbf{j} = (j_x, j_y, j_z)$ and $\mathbf{k} = (k_x, k_y, k_z)$ be unit vectors pointing in the local I, J, and K directions respectively for the block in which the layer is perforated. These are determined from data entered for the block in the RESERVOIR DESCRIPTION section of the data set. Note that for corner-point grids (see the RESERVOIR DESCRIPTION section of this manual) these directions are not necessarily aligned with the underlying Cartesian axes of the grid (those in which the coordinates x_1, y_1, z_1 etc. are defined) nor are the vectors **i**, **j**, and **k** necessarily mutually orthogonal. The following two methods are supported.

Cardinal Weighting (original)

Define $\cos(\theta_i)$ as the dot product $\mathbf{u} \cdot \mathbf{i}$, and let $\cos(\theta_j)$ and $\cos(\theta_k)$ be similarly defined. Define $\sin^2(\theta_i) = 1 - \cos^2(\theta_i)$ and similarly for J and K. Then the interpolated value of *re* to apply in the direction of the wellbore is

$$re\,(\mathbf{u}) = [wt_i\, re^2(\mathrm{I}) + wt_j\, re^2(\mathrm{J}) + wt_k\, re^2(\mathrm{K})]^{1/2}$$

with

$$wt_i = \cos^2(\theta_i) \sin^2(\theta_j) \sin^2(\theta_k) \,/\, S$$

$$wt_j = \cos^2(\theta_j) \sin^2(\theta_k) \sin^2(\theta_i) \,/\, S$$

$$wt_k = \cos^2(\theta_k) \sin^2(\theta_i) \sin^2(\theta_j) \,/\, S$$

where *S* is the sum of the three trigonometric weighting factors in the numerator.

Cardinal Weighting was the only method available before version 2011.10.

Directional Weighting (default)

The wellbore direction $\mathbf{u} = (u_x, u_y, u_z)$ in the global orthogonal XYZ system is transformed into the local (potentially) non-orthogonal IJK system $\mathbf{u} = (u_i, u_j, u_k)$ by

$$i_x\, u_i + j_x\, u_j + k_x\, u_k = u_x$$

$$i_y\, u_i + j_y\, u_j + k_y\, u_k = u_y$$

$$i_z\, u_i + j_z\, u_j + k_z\, u_k = u_z$$

The determinant of the above 3 × 3 system measures the orthogonality of IJK, which cannot be zero for a valid grid block. The solved wellbore direction in IJK (u_i, u_j, u_k) is re-normalized and applied to the directional weighting of the given *re*(I), *re*(J) and *re*(K):

$$re\ (\mathbf{u}) = [u_i^2\, re^2(\mathrm{I}) + u_j^2\, re^2(\mathrm{J}) + u_k^2\, re^2(\mathrm{K})]^{1/2}$$

The Directional Weighting is the recommended (default) method after version 2011.10. One can switch it back to the original Cardinal Weighting for backward compatibility by keyword *LAYERXYZ-METHOD *OLD.

Unless *KH is specified by the user on the *PERF lines, the completion planar averaged permeability *K* is computed similarly except that *re*(I), *re*(J), and *re*(K) are replaced with

$$K(\mathrm{I}) = (K_j\, K_k)^{1/2}$$
$$K(\mathrm{J}) = (K_k\, K_i)^{1/2}$$
$$K(\mathrm{K}) = (K_i\, K_j)^{1/2}$$

Example 1:

```
*LAYERXYZ  'WELL-NNE17'
 65 23 5
** x1      y1      z1       x2      y2      z2    h
2287.49 1457.64 3949.09  2284.34 1460.23 3944.28  2.67
```

A completion for well 'WELL-NNE17' in block 65 23 5 must already have been created with a *PERF.. line for the above to be valid.

Example 2:

```
*LAYERXYZ  'WELL-MULTI-REF'
 16 48 11 / 1 1 2 / 2 2 1
** x1      y1      z1       x2      y2      z2      h
 102.11  493.74 2285.53   102.48  494.87 2284.13   2.67
```

In this example, the already defined completion of well 'WELL-MULTI-REF' in the multiply-refined grid block 16 48 11 / 1 1 2 / 2 2 1 is flagged as deviated and the seven values above are assigned to x_1, y_1, z_1, x_2, y_2, z_2, and *h*.

Simplified Geometric Data for Deviated Well Completions (Conditional) *LAYERIJK

PURPOSE:

*LAYERIJK allows the user to supply a layer direction for each layer. Perforations are parallel to one of the local coordinate axes but can vary from layer to layer. The directions defined on the *LAYERIJK keyword override the well direction specified on the *GEOMETRY keyword.

FORMAT:

```
*LAYERIJK  'wname'
{locat.}          {layer direction}
   :                     :
```

DEFINITIONS:

wname

Single well name within single quotes specifying the well to which the following deviated layer specifications apply. No wild-carding is supported in this situation.

{locat.}

$i_f \quad j_f \quad k_f$ (/ …(/ $i_{rn} \quad j_{rn} \quad k_{rn}$)…)

i_f, j_f, k_f are integers or integer ranges of the fundamental grid block index in the I, J and K direction (User Block Address) specifying the layers of well '*wname*' to be treated as deviated layers. It is valid to name some of the well's layers and not others in the *LAYERIJK lines. Layers not mentioned do not acquire the deviated status; layer direction is then defined by the currently active GEOMETRY keyword. Any layer named under *LAYERIJK must already have been defined for the well using a *PERF or *PERFV statement.

Optional (i_{rn}, j_{rn}, k_{rn}) are similar to (i_f, j_f, k_f), but at the *n*'th level of refinement (if applicable) for the perforated grid block. At most one direction may have a range.

{layer direction}

(*I | *J | *K | *UNDEVIATED)

*I, *J, *K

Signifies the layer is perforated in the local I, J or K direction.

*UNDEVIATED

Subkeyword indicating that the layer is henceforth to be treated as an undeviated layer with the direction (*I,*J, or *K) which the layer had when initially perforated (defined by active *GEOMETRY keyword).

DEFAULTS:

Layers named under *LAYERIJK with geometric information *I, *J or *K (as opposed to the *UNDEVIATED subkeyword) are flagged as deviated within the simulator; undeviated is the default status assigned at the beginning of the run. The undeviated status can be reimposed through the *UNDEVIATED subkeyword.

CONDITIONS:

The named layers must all have been previously created for well '*wname*'with *PERF or *PERFV lines. Not all of a well's layers need to be named under *LAYERIJK; those omitted are treated as undeviated according to the original *PERF specification.

EXPLANATION:

This keyword specifies geometric information that allows the well indices to be calculated for perforations in which the wellbore direction is parallel to one of the grid block coordinate axes but varies by layer. The *ff* factor entered under the *PERF or *PERFV line when the layer was created still apply to the deviated layer.

The deviated well index is calculated in a manner identical to the LAYERXYZ keyword with the following additional assumptions made. It is assumed a perforation enters a block at the center (barycenter) of one face perpendicular to the layer direction (*I, *J, or *K) and exits at the center (barycenter) of the other face. The length of the perforation is the length of the line connecting one face center to the other.

For example, a *LAYERIJK *I perforation would follow the path defined by connecting the center of the lower I face (the face connecting block I with block I-1) to the center of the upper I face (the face connecting the block I with the block I+1). The length of the connection is the distance between the centers of the lower I face and upper I face. Using the notation defined in the *LAYERXYZ keyword explanation, we also assume that *re*(u) equals *re*(I) and K(u) = K(I).

A *LAYERIJK *J or *LAYERIJK *K perforation would have its perforation length, effective radius, and well permeability calculated in an analogous fashion (references to I would be replaced by J or K in the above paragraph).

Example 1:

```
*LAYERIJK  'WELL-NNE17'
 65 23 5 *I
```

A completion for well 'WELL-NNE17' in block 65 23 5 must already have been created with a *PERF.. line for the above to be valid, the well perforation is parallel to the I direction

Example 2:

```
*LAYERIJK  'WELL-MULTI-REF'
 16 48 11 / 1 1 2 / 2 2 1 *J
```

In this example, the already defined completion of well 'WELL-MULTI-REF' in the multiply-refined grid block 16 48 11 / 1 1 2 / 2 2 1 is flagged as deviated, the well perforation is parallel to the J direction.

Relative Permeability Data for Well Completions (Optional)

***KRPERF**

PURPOSE:

*KRPERF specifies the relative permeability table set numbers and end points for well completions.

FORMAT:

```
*KRPERF        (wn)
{locat.}       {rel. perm. data}
   :                  :
```

DEFINITIONS:

wn

A single well sequence number or single well name enclosed in single quotes on the same line as *KRPERF. *wn* is optional and if it is absent, *KRPERF must immediately follow the sequence of layer lines introduced by the last *PERF(V) keyword for *wn* referring.

{locat.}

The layer location identification that have been previously defined by the *PERF(V) keyword. If *wn* is specified, the full grid block index in the I, J, K direction (User Block Address) must be provided. Otherwise, it follows the format of the preceding *PERF(V). Please see the appropriate manual page.

{rel. perm. data}

A set of one or more of the subkeywords *SETN, *BSWCON, *BSWCRIT, *BSORW, *BSOIRW, *BSGCON, *BSGCRIT, *BSORG, *BSOIRG and *BSLCON, each followed by a number (integer or real) giving the relative permeability table set number or end point value.

*SETN

Introduces the integer relative permeability table set number (set number) used to calculate the mobility entering the well index expression. If not specified, the relative permeability curve set number for the perforated grid block is used.

*BSWCON

Introduces the real number specifying the connate water saturation for the completion, which scales the Krow curve of the *SWT relative permeability table. *BSWCON is the smallest water saturation possible in the *SWT table. One minus *BSWCON is the maximum oil saturation possible in the *SWT table. If *BSWCON is not specified for the completion, then the connate water saturation used is obtained the last *BSWCON keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSWCON keyword from the currently active relative

permeability table as specified on the *RTYPE keyword or as respecified on the *KRPERF *SETN keyword.

*BSWCRIT

Introduces the real number specifying the critical water saturation for the completion, which scales the Krw curve of the *SWT relative permeability table. *BSWCRIT is the water saturation at which water first becomes mobile in the two phase oil-water table. If *BSWCRIT is not specified for the completion, then the critical water saturation used is obtained the last *BSWCRIT keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSWCRIT keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword. *BSWCRIT should be equal to or greater than *BSWCON.

*BSOIRW

Introduces the real number specifying the irreducible oil saturation for the completion, which scales the Krw curve of the *SWT relative permeability table. *BSOIRW is the smallest oil saturation in the two phase oil-water table, one minus *BSOIRW is the maximum water saturation in the table. If *BSOIRW is not specified for the completion, then the irreducible oil saturation used is obtained the last *BSOIRW keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSOIRW keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword.

*BSORW

Introduces the real number specifying the residual oil saturation for the completion, which scales the Krow curve of the *SWT relative permeability table. *SORW is the oil saturation in the two phase oil-water table at which oil first becomes mobile. If *BSORW is not specified for the completion, then the residual oil saturation used is obtained the last *BSORW keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSORW keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword. *BSORW should be equal to or greater than *BSOIRW.

*BSGCON

Introduces the real number specifying the connate gas saturation for the completion, which scales the Krog relative permeability curve of the *SGT table. *SGCON is the smallest gas saturation in the two phase gas-liquid relative permeability table. If *BSGCON is not specified for the completion, then the connate gas saturation used is obtained the last *BSGCON keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSGCON keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword.

*BSGCRIT

Introduces the real number specifying the critical gas saturation for the completion, which scales the Krg relative permeability curve of the *SGT table. *BSGCRIT is the gas saturation in the two phase gas-liquid relative permeability table at which gas first becomes mobile. If *BSGCRIT is not specified for the completion, then the critical gas saturation used is obtained the last *BSGCRIT keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSGCRIT keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword. *BSGCRIT should be equal to or greater than *BSGCON.

*BSLCON

Introduces the real number specifying the connate liquid saturation for the completion, which scales the Krog relative permeability curve of the *SGT table. *BSLCON is the smallest liquid saturation entry in the two phase gas-liquid relative permeability table. If *BSLCON is not specified for the completion, then the connate liquid saturation used is obtained the last *BSLCON keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSLCON keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword.

*BSORG

Introduces the real number specifying the residual oil saturation for the completion, which scales the Krog relative permeability curve of the *SGT table. *BSORG is the oil saturation in the two phase gas-liquid relative permeability table at which oil becomes mobile. If *BSORG is not specified for the completion, then the residual oil saturation used is obtained the last *BSORG keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSORG keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword.

*BSOIRG

Introduces the real number specifying the irreducible oil saturation for the completion, which scales the Krg relative permeability curve of the *SGT table. *BSOIRG is the smallest oil saturation in the two phase oil-gas table, one minus *BSOIRG minus *BSWCON is the maximum gasr saturation in the table. If *BSOIRG is not specified for the completion, then the connate liquid saturation used is obtained the last *BSOIRG keyword specified either from the rock fluid or recurrent data sections or in the absence of *BSOIRG keyword from the currently active relative permeability table as specified on the *RTYPE or as respecified on the *KRPERF *SETN keyword.

DEFAULTS:

*SETN, *BSWCON, *BSGCRIT, *BSORW, and *BSORG can be set either directly in the layer line after *PERF or *PERFV, or after *KRPERF. If no relative permeability data are

entered in the *PERF layer line, then defaults are set (or reset in the case of reperforation), and non-default values, if wanted, must be established or re-established with *KRPERF. The *BSWCRIT , *BSOIRW, *BSGCON and *BSLCON end points can only be set with the *KRPERF keyword.

When values of *SETN, *BSWCON, *BSGCRIT, *BSORG, and *BSORG are read in on the various *PERF cards, their values will be overridden by the values of these variables encountered on the *KRPERF card.

Please refer to the detailed discussion of relative permeability scaling given in the manual's Rock-Fluid Data section under the array keywords *BSWCON, *BSWCRIT, *BSORW, *BSOIRW, *BSGCON, *BSGCRIT, *BSLCON, *BSOIRG or *BSORG.

When the backward compatibility option *SCALING-OLD is used for a relative permeability set (see *RPT keyword). Only the variables *SETN, *BSWCRIT, *BSGCRIT, *BSORG, and *BSORG may be entered on the *KRPERF card. The values of the new variables are ignored.

We recommend that users employing *SCALING-OLD do not use the *KRPERF option.

The following rules are used to resolve inconsistencies in the end point data.

1. Critical water saturation is less than connate water saturation. Critical water saturation is reset to connate water saturation.
2. Irreducible oil saturation in the water-oil table is greater than residual oil. Critical oil saturation is reset to irreducible oil saturation.
3. Irreducible oil saturation in the gas-liquid table (Soirg) is greater than residual oil. Irreducible oil saturation is reset to critical oil saturation. Irreducible oil is calculated from *BSLCON and *BSWCON. Soirg equals *BSLCON minus *BSWCON.
4. Critical gas saturation is less than connate gas saturation. Critical gas saturation is reset to connate gas saturation.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group, after layers named in {location} have been defined by previous *PERF or *PERFV.

The *KRPERF keyword is optional, and when it appears there is no requirement that all perforated layers of the well must be named under *KRPERF; however, layers not named under *KRPERF receive the table set numbers and end point values set by default or set directly in the *PERF.. lines.

Keywords *BSLCON and *BSOIRG are mutually exclusive. If *SCALING_STARS option is turned then use keyword *BSOIRG otherwise use keyword *BSLCON.

EXPLANATION:

This keyword provides an alternative to entering completion relative permeability table set numbers and end points on the *PERF or *PERFV lines. In addition this keyword allows the user access to more end points. We recommend that users with new data sets enter all completion end points and relative permeability set numbers on the *KRPERF card.

Values for *SETN, *BSWCON, *BSORW, *BSORG, and *BSGCRIT may be entered either in the *PERF.. line or using the *KRPERF keyword; *BSWCRIT, *BSOIRW, *BSGCON, *BSOIRG and *BSLCON can be entered only by using the *KRPERF keyword.

This option can be used to represent effects such as partial completions where relative permeability end points may be altered. For example, if a producer is completed only in the top quarter of a grid block, then the water must cone up to the perforation (e.g. Sw = 0.67) before water can flow into the wellbore. So a water relative permeability set with a critical water saturation of 0.67 can be assigned to the partial completion using *SETN without affecting reservoir flow.

Example 1:

```
*WELL 1 'Producer 1'
...
**              rw   geofac  wfrac  skin
*GEOMETRY *K  0.375  0.2488  1.0    0.0

*PERF *GEO ‘Producer 1’
** if   jf  kf   ff    setn  swcon  sgcrit  sorw  sorg
   12   6   2:4  0.873  7    0.20    0.15   0.17  0.15

*ON-TIME 'Producer 1'
    0.9

*KRPERF 'Producer 1'
   12   6  2:3  *BSWCRIT  0.22  *BSOIRW  0.14
```

In this example, the completions of well 1 in layers 2 and 3 have the completion critical water saturation set to 0.22 and the completion irreducible oil saturation set to 0.14. Note that *SWCRIT exceeds the connate water saturation of 0.20 set in the layer perforation line, and that the value of the connate oil saturation (0.14) is less than the residual oil saturation on the water-oil curve (0.17).

Example 2:

```
*PERFV *KH ‘WELL*’
**  kf   ff
   1:4   1.0

*KRPERF
   1:3 *SETN  2      *BSWCRIT 0.15  *BSGCRIT 0.10
       *BSORW  0.12  *BSORG   0.10  *BSWCON 0.12
       *BSOIRW 0.10  *BSLCON  0.17  *BSGCON 0.00
    4  *SETN 3       *BSWCRIT 0.25
```

In this example, all wells with ‘WELL’ as the first four characters of their names are completed in layers 1 through 4 at the I and J blocks specified after *VERT in the defining *WELL line. If any well in this set does not have I and J indices specified, the *PERFV line is not valid. In the line following *PERFV, the defaults are set for all of the layer relative permeability variables (since they are not referred to). The first line following *KRPERF resets all nine relative permeability variable values to the entered values for layers 1:3 of the listed wells; for layer 4 a table set number (3) is provided and all other quantities except *BSWCRIT default. In layer 4, *BSWCRIT is being set high to reduce water production.

Pressure Gradients for Calculation of Pressure Differences Between Completions (Conditional) *LAYERGRAD

PURPOSE:

*LAYERGRAD allows the user to specify pressure gradients to be used in a static calculation of the pressure difference between adjacent completions within a wellbore. This keyword can be used, for example, to model pumped-off producing wells.

FORMAT:

```
*LAYERGRAD  'wname'
{location}      (pressure_gradient_value|*DEFAULT)
   :                              :
```

DEFINITIONS:

wname

Single well name within single quotes specifying the well to which the layer pressure gradient specifications apply. No wild-carding is supported in this situation.

{location}

if jf kf / ir1 jr1 kr1 { / … { / irn jrn krn} }

These triples of integers specify the layers of well 'wname' to be assigned user-specified head pressure gradients. It is valid to name some of well 'wname's layers and not others in the *LAYERGRAD lines, but layers not named do not acquire a special gradient value and have heads calculated in the normal way. Any layer named under *LAYERGRAD must already have been defined for the well using a *PERF or *PERFV statement. The gradient specified will be used for the head calculation between the named layer and the next heel-ward completion (the completion to which the named completion flows – see the discussion in the manual page for the *PERF keyword).

pressure_gradient_value

Non-negative real value (kPa/m | psi/ft) specifying the gradient to be used for the pressure difference calculation.

*DEFAULT

Specifies that the pressure difference between the named layer and the next heel-ward layer should be calculated normally, not using a specified pressure gradient value. The head calculation method in place can be specified using the *HEAD-METHOD keyword.

if

A single integer specifying the I direction grid block index of the fundamental grid (see explanation).

jf

A single integer specifying the J direction grid block index of the fundamental grid. (See explanation).

kf

A single integer specifying the K direction grid block index of the refined grid. (See explanation).

ir1 jr1 kr1

Three integers specifying the I, J, K direction grid block index of the first level of refined grid that the well is completed through.

irn jrn krn

Three integers specifying the I, J, K direction grid block index of the n'th level of refined grid that the well is completed through.

DEFAULTS:

Layers named under *LAYERGRAD with a gradient value (as opposed to the *DEFAULT subkeyword) are flagged as receiving special treatment during the layer head calculation. The default status is assigned to all layers at the beginning of the run. The default status can be re-imposed through the *DEFAULT subkeyword.

CONDITIONS:

The named layers must all have been previously created for well ‘wname’ with *PERF or *PERFV lines. Either a non-negative real number or the subkeyword *DEFAULT must follow the layer identification or an error is generated. Not all of a well’s layers need to be named under *LAYERGRAD; those omitted have heads calculated in the default manner.

EXPLANATION:

When no pressure gradient is specified for a layer, the pressure difference in the wellbore between one layer and the layer to which it flows is calculated either statically, using a pressure gradient depending on local fluid densities, or using a correlation which includes frictional effects. See the manual page for the *HEAD-METHOD keyword for a description of the head calculation methods available. When a head pressure gradient hgrad is specified with *LAYERGRAD, the pressure difference between the named layer and the next heel-ward completion in the well (which is unique) is calculated as

Delp = (P_next – P_named) = hgrad*(depth_next – depth_named)

See the manual entry for the *PERF keyword for an explanation of how the identity of the next heel-ward layer (the layer to which the named layer flows) is determined. The layer named first in the *PERF.. lines flows to the surface and no layer head is associated with the corresponding pressure difference; hence any specification of a head gradient for such a layer is ignored (but is valid).

The *LAYERGRAD keyword can be used to model pumped-off producing wells by specifying zero or gas-like pressure gradients between completions which are above the pump level.

Example 1:

```
*LAYERGRAD  'WELL-NNE17'
 65 23 5     0.5
 65 23 6     0.4
```

Completions for well 'WELL-NNE17' in blocks 65 23 5 and 65 23 6 must already have been created with a *PERF.. line for the above to be valid.

Example 2:

```
*LAYERGRAD  'PUMPED-WELL'
 16 48 11   0.
 16 48 12   0.
 16 48 13   0.
 16 48 14   0.
```

If well 'PUMPED-WELL' has completions in blocks 16 48 10:15, with layer 15 deepest, the above represent a pump located just above the perforation in block 16 48 14. The pressure difference in the wellbore between blocks 16 48 15 and 16 48 14 is calculated by the usual head method for the run, but all heads above this are set to zero, to simulate the absence of liquids above the completion in layer 14. Note that no gradient specification for the completion in block 16 48 10 is necessary because this layer flows to the surface.

User-Specified Reference Depth for Well BHP (Optional)

***BHPDEPTH**

PURPOSE:

*BHPDEPTH allows the user to specify a depth to which a well's bottomhole pressure is referred. When *BHPDEPTH has been specified for a well, the BHP for that well is in general not one of the wellbore completion pressures. The related keyword *BHPGRAD can be used to specify a pressure gradient used to determine the pressure difference between the reference depth and the reference completion.

FORMAT:

```
*BHPDEPTH  well_list     depth_values
```

DEFINITIONS:

well_list
: A set of 'well_names' or well_numbers; see below.

well_names
: Any number of well names (in quotes) to specify the wells to which this specification of reference depth applies. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

well_numbers
: Any number of integers, or integer ranges, to specify the wells to which this specification of reference depth applies.

depth_values
: A list consisting of either non-negative real numbers (m | ft) or the subkeyword *DEFAULT. If the depth_values list contains only a single entry then this entry will be applied to all wells in well_list; if there are more than a single entry in the depth_values list, the number of entries must match the number of wells in well_list and the first depth value is applied to the first well, etc. The numbers representing depths must contain a decimal point to identify them as real numbers and distinguish them from well numbers.

*DEFAULT
: When *DEFAULT occurs in the depth_value list, it restores the well to the default state in which the bottomhole pressure is the wellbore pressure in the reference completion.

DEFAULTS:

Optional keyword. If BHPDEPTH does not appear in the data set, all wells have bottomhole pressure equal to the wellbore pressure in the reference completion. The occurrence of BHPDEPTH does not affect the operation of any well not named in a well list following BHPDEPTH.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. All wells appearing in the well list following *BHPDEPTH must already have been defined in *WELL lines. The effect of multiple *BHPDEPTH lines is cumulative; that is, if *BHPDEPTH appears first followed by one well list and later followed by another, at the end of the second occurrence the wells in both lists will have the specified BHP reference depths. A well which has previously had a reference depth defined can be restored to having the BHP be the wellbore pressure in the reference completion by entering *DEFAULT for that well in another *BHPDEPTH line. If a well has one reference depth set in an earlier *BHPDEPTH line and then appears in a later *BHPDEPTH list with a different depth value, the more recently specified value overwrites the earlier value and is used in BHP calculations after its entry. The depth values must contain a decimal point to identify them as real numbers and distinguish them from well numbers. There is no restriction upon the distribution of the well list or depth value list over different lines in the data set; the depth values may begin on the same line as the last well identifier in the list and continue over as many lines as necessary. The well list too may be spread over more than a single line.

EXPLANATION:

When a BHP reference depth is entered for a well using *BHPDEPTH, the BHP differs from the wellbore pressure in the well's reference completion. The pressure difference is equal to

delp = g*rho*(ref_depth – completion_depth)

Here g is the gravitational acceleration and rho is a mobility-weighted fluid density characteristic of the reference completion. Using the *BHPGRAD keyword the user may enter a pressure gradient which replaces g*rho in the above formula. See the manual entry for the *PERF keyword for details of how the reference layer is specified.

Example: The sequence

```
*BHPDEPTH 'Prod1' 'Prod2' 'Inj1' 'Inj2'
1500.  1500.  1000.  1000.
```

Assigns BHP reference depths of 1500 units (feet or meters) to the two producers and of 1000 units to the two injectors. In this example the depth value list begins (but need not have begun) a new line of the data set.

User-Specified Pressure Gradient For Reference Depth for Well BHP (Optional) *BHPGRAD

PURPOSE:

*BHPGRAD allows the user to specify a pressure gradient which is used to compute the pressure difference between a well's reference completion and the reference depth to which the well's bottomhole pressure is referred. The entered gradient has no effect unless a reference depth for the well has been specified using *BHPDEPTH.

FORMAT:

*BHPGRAD *well_list* *gradient_values* | *DEFAULT

DEFINITIONS:

well_list

A list of 'well_names' or well_numbers to which this specification applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

gradient_values

A list consisting of either non-negative real numbers (kPa/m | psi/ft) or the subkeyword *DEFAULT. If the *gradient_values* list contains only a single entry then this entry will be applied to all wells in *well_list*. If there are more than a single entry in the *gradient_values* list, the number of entries must match the number of wells in *well_list* and the first gradient value is applied to the first well, etc. The numbers representing gradients must contain a decimal point to identify them as real numbers and distinguish them from well numbers.

*DEFAULT

When *DEFAULT occurs in the *gradient_value* list, it restores the well to the default state in which a mobility-weighted fluid density in the reference completion is used to determine the pressure difference between reference depth and the reference completion.

DEFAULTS:

Optional keyword. If BHPGRAD does not appear in the data set, all wells specified under a *BHPDEPTH list have the pressure difference between reference completion and reference depth calculated using a fluid density characteristic of the reference completion. The occurrence of BHPGRAD does not affect the operation of any well not named in a well list following BHPGRAD, and has no effect unless the well also occurs in a list following the *BHPDEPTH keyword.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. All wells appearing in the well list following *BHPGRAD must already have

been defined in *WELL lines. The effect of multiple *BHPGRAD lines is cumulative; that is, if *BHPGRAD appears first followed by one well list and later followed by another, at the end of the second occurrence the wells in both lists will have the specified BHP reference depths. A well which has previously had a reference gradient defined can be restored to the default calculation using a weighted density by entering *DEFAULT for that well in another *BHPGRAD line. If a well has one gradient set in an earlier *BHPGRAD line and then appears in a later *BHPGRAD list with a different gradient value, the more recently specified value overwrites the earlier value and is used in BHP calculations after its entry. The gradient values must contain a decimal point to identify them as real numbers and distinguish them from well numbers. There is no restriction upon the distribution of the well list or gradient value list over different lines in the data set; the gradient values may begin on the same line as the last well identifier in the list and continue over as many lines as necessary. The well list too may be spread over more than a single line.

EXPLANATION:

When a BHP pressure gradient is entered for a well using *BHPGRAD, the pressure difference between the reference completion and the reference depth is calculated as

delp = p_grad*(ref_depth – completion_depth)

See the manual entry for the *PERF keyword for details of how the reference layer is specified.

Example: The sequence

```
*BHPGRAD 'Prod1' 'Prod2' 'Inj1' 'Inj2'
0.5  0.5  0.4  0.4
```

assigns reference pressure gradients of 0.5 psi/ft to the two producers and of 0.4 psi/ft to the two injectors. In this example the gradient value list begins (but need not have begun) a new line of the data set.

Gas Recycling Option (Optional) *RECYCLE_GEM

PURPOSE:

*RECYCLE_GEM indicates the entering of data for the non- group controlled gas recycling option. If group control is used (i.e. if the keywords *GCONI or *GCONP appear anywhere in the data set) then this keyword cannot be used and recycling scenarios must be modelled using the *GCONI keyword.

FORMAT:

*RECYCLE_GEM	(*RECFRC)	*sep_comp_recyc_frac*
	(*RATE)	*comp_make_up_vol*

DEFINITIONS:

sep_comp_recyc_frac
A set of nc fractions for the nc components corresponding to the recycling mole or volume fraction (i.e. recycle the given fraction of the i-th component in the collected separator gases from all cycling producers).

comp_make_up_vol
A set of nc real numbers corresponding to the flow rates of the nc components in the make-up stream. This make-up rate refers to the whole field. (m^3/day | ft^3/day)

DEFAULTS:

Optional keywords.

The default for *sep_comp_recyc_frac* is one.

The default for *comp_make_up_vol* is zero.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. If group control is involved then use *RECYCLE as a subkeyword of *GCONI in the data deck. *RECYCLE_GEM must appear on the same line as *RECFRC or *RATE and before the subkeyword.

EXPLANATION:

The recycling option in GEM allows for the mimicking of certain processes in surface facilities such as the stripping of certain components and the addition of a make-up stream (see Appendix B, Figure 8).

Separator gas streams from all recycling producers, i.e. those with well type = *CYCLPROD, will first be mixed. For each component *sep_comp_recyc_frac* fraction in this gas stream will be recycled. This recycled stream is then mixed with a make-up stream with flow rates *comp_make_up_vol* for each component. The mixed stream then forms the injection fluid for wells with *INCOMP *CYCLING specifications.

Note: This primary keyword (*RECYCLE_GEM) was earlier referred to as *RECYCLE which is still supported but is deprecated.

If the constraint

*MONITOR *MIN *REC *val*

is applied to the injector, the constraint is said to be violated if the injection fluid flow rate (with the makeup stream included) falls below the constrained value, val. Thus if the sum of *f_makeup* is greater than *val*, the constraint will never be violated.

Example: Consider the following data segment,

```
...
*COMPNAME 'CO2' 'C1' 'C3' 'C6' 'C7+'
...
*RECYCLE_GEM *RECFRC 1.0    0.0  0.0  0.0  0.0
*RECYCLE_GEM *RATE   1.0E5  0.0  0.0  0.0  0.0
```

In this case, all hydrocarbons from the separator gases will be stripped, and the remaining carbon dioxide then recycled. To make-up for the losses, 100000.0 m^3/day of CO2 is added. The recycling stream will then consist purely of CO2.

If the following data are entered,

```
*RECYCLE_GEM *RECFRC 1.0    1.0  1.0  0.0  0.0
*RECYCLE_GEM *RATE   1.0E5  0.0  0.0  0.0  0.0
```

only the heavy ends in the separator gases are stripped out. The remaining gas stream will again be mixed with 100000.0 m^3/day of CO2 before reinjection. In this case, the reinjected fluid will not be pure CO2, but will vary according to the producing rates and fluids.

Group Production Constraints (Optional) *GCONP

PURPOSE:

*GCONP is used to specify group production controls.

FORMAT:

```
*GCONP  group_list
(*MAX)        (*STO)       value        (*STOP)
              (*STG)                    (*CONT)
              (*STW)                    (*SHUTALL)
              (*STL)                    (*SHUTMOWS)
              (*STI)                    (*SHUTMOW)
              (*WTG)                    (*SHUTMOL)
              (*BHF)                    (*SHUTMOLDOWN)
                                        (*SHUTMOLUP)
(*MIN)        (*MNP)       value        (*STOP)
              (*CPP)                    (*CONT)
(*GTARGET)    (*STO)       value
              (*STG)
              (*STW)
              (*STL)
              (*STI)
              (*WTG)
              (*BHF)
              (*MNP)
              (*CPP)
              (*NONE)
(*VREP)                    vrep_frac
(*RECYCLE)    (*GAS)       recyc_frac
              (*WATER)
(*PMAINT)     (*PMSECT)    'sector_name'
              (*PMTARG)    p_targ
              (*PMCOEF)    c1 c2 c3
```

DEFINITIONS:

group_list

Are the groups to which the following constraints apply. The wells that are connected to each group must already have been specified using the *WELL keyword. Production targets are apportioned using one of the available apportionment methods specified by *APPOR-METHOD.

*MAX

Specifies that the constraint is a maximum constraint. This value becomes a target for the group only as the result of a violation with the *CONT action.

*MIN

Specifies that the constraint is a minimum constraint. This value becomes a target for the group only as the result of a violation with the *CONT action. *STOP and *CONT are the only actions available for MIN constraints, and MNP (manifold pressure) and CPP (compressor or surface pressure) are the only group production constraints which accept the MIN designation.

*GTARGET

This subkeyword specifies a target production rate for the group. The specified stream rate is set as a target to be met by the group. If another constraint with action *CONT is violated, the rate target shifts to that constraint and the target set by the *GTARGET keyword ceases to have an effect. There is no action associated with *GTARGET since a target is not checked for violation.

*VREP

This subkeyword specifies a voidage replacement production target. This indicates that the production wells connected to this group produce an amount of the bottom-hole fluid in proportion to the total bottom-hole fluid injected into the reservoir by the injection wells connected to this group.

*RECYCLE

This subkeyword specifies a recycling production target. This indicates that the production wells connected to this group produce such that the phase injected by the injection wells connected to this group as specified by *GAS or *WATER is reproduced (recycled) out of the reservoir.

*PMAINT

This subkeyword specifies that the group production rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector (*PMSECT) at a desired level (*PMTARG).

*STO

This subkeyword identifies a surface oil rate (m^3/day | Stb/day) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

*STG

This subkeyword identifies a surface gas rate (m^3/day | SCF/day) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

*STW

This subkeyword identifies a surface water rate (m^3/day | Stb/day) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

*STL

This subkeyword identifies a surface liquid rate (oil + water + intermediate liquid) (m^3/day | Stb/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

*STI

This subkeyword identifies a surface intermediate liquid rate (m^3/day | Stb/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group. For more information on the intermediate liquid stream please see the description of the *WELSEP keyword in this manual.

*WTG

This subkeyword identifies a surface wet gas rate (m^3/day | SCF/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group. For more information on the wet gas stream please see the description of the *WELSEP keyword in this manual.

*BHF

This subkeyword identifies a bottomhole fluid rate (m^3/day | rbbl/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

*MNP

This subkeyword introduces a manifold pressure (kPa | psi) constraint. This may only be applied if the listed groups have all had production specified as going through a manifold with the *MANIFOLD keyword.

*CPP

This subkeyword introduces a compressor (surface) pressure (kPa | psi) constraint. This may only be applied if the listed groups have all had production specified as going through a manifold with the *MANIFOLD keyword.

*NONE

Remove any production target specified previously (including *VREP, *RECYCLE and *PMAINT). Existing maximum / minimum constraints are not affected.

*PMSECT

Introduces a single sector identified by the name ‘sector_name’ whose average hydrocarbon pore-volume pressure is to be maintained. ‘sector_name’ must be 16 characters maximum and must have already been defined in the RESERVOIR DESCRIPTION section in the input data. Defaulted to full ‘FIELD’.

*PMTARG

Introduces the targeted average hydrocarbon pore-volume pressure (kPa | psi) for the sector. The value p_targ must be a real number larger than one atmosphere. Defaulted to the current sector pressure.

*PMCOEF

Introduces the control coefficients used in the pressure control strategy. The values (c1, c2, c3) must be non-negative real numbers. Defaulted to the internally estimated values.

value

Constraint value -- see above for units.

vrep_frac

When the voidage replacement subkeyword is used (*VREP) the vrep_frac is the voidage replacement ratio. A ratio of 1.0 indicates that the bottom-hole fluid injected is completely produced.

recyc_frac

When the recycle subkeyword is used (*RECYCLE) the recyc_frac is the recycled fraction of the indicated surface stream, which is imposed as a group production target. A fraction of 1.0 indicates complete reproduction / recycling of the injected surface stream.

*STOP

Action subkeyword indicating that if the constraint cannot be met then the simulation should be stopped.

*CONT

Action subkeyword indicating that the simulation continues with the violated constraint becoming the target constraint. This is the default action if no action subkeyword is specified.

*SHUTALL

Action subkeyword indicating that if a maximum production rate is exceeded for a group, then all currently open wells in the group should be shut. Not available for MNP and CPP constraints.

*SHUTMOWS

Action subkeyword indicating that if a maximum production rate is exceeded for a group, then a list of prioritized most offending wells (MOWS -- the ones with the higher rates of the named surface stream) should be shut. Not available for MNP and CPP constraints.

*SHUTMOW

Action subkeyword indicating that if a maximum production rate is exceeded for a group, then the most offending well (MOW -- the one with the highest rate of the named surface stream) should be shut. Not available for MNP and CPP constraints.

*SHUTMOL

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending layer (MOL) in the most offending well (the one the highest rate of the named surface stream) should be shut. Not available for MNP and CPP constraints.

*SHUTMOLDOWN

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending layer (MOL) and the layers below it in the most offending well (the one with the highest rate of the named surface stream) should be shut. Not available for MNP and CPP constraints.

*SHUTMOLUP

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending layer (MOL) and the layers above it in the most offending well (the one with the highest rate of the named surface stream) should be shut. Not available for MNP and CPP constraints.

DEFAULTS:

Optional keyword. Default is no production constraints on any group. Default action is *CONT.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can be given any constraint values. Constraint types *GTARGET, *RECYCLE, *VREP and *PMAINT are exclusive in one timecard, and only the latest counts. If a group is assigned by *GCONP with the following dependent constraints: *VREP, *RECYCLE, or *PMAINT, such a group cannot be assigned by *GCONI with these constraints for any of its injection streams. Error message will be issued if the consistency of the dependent constraints is violated.

EXPLANATION:

*GCONP is used to specify constraints on how much fluid is produced in the group. *GCONP can also be used to specify voidage replacement, recycling or pressure maintenance targets (see explanations for keyword *GCONI). This allows group controls to adjust production rates in response to injection.

Example:

```
*GCONP  'Field'
*MAX  *STW  10000. *SHUTMOLDOWN
```

This specifies that group 'Field' has a maximum stock tank water production constraint, with the action upon violation being to shut the most offending layer of the most offending well (layer and well with highest water rate) and any lower perforations.

Example:

```
*GCONP 'Group-1'
*GTARGET *MNP 2800.
```

This sets a manifold pressure target of 2800. units for the group 'Group-1'. This group must have had its production assigned manifold treatment with the *MANIFOLD keyword. 'Group-1' must have wells directly attached to it.

A number of apportionment methods (e.g. guide rate, instantaneous potential, priority ranking, and internal guide rate) can be used to distribute a group production target among the contributing wells or groups through the keyword *APPOR-METHOD (which is described on a separate manual page).

Group Injection Constraints (Optional) *GCONI

PURPOSE:

*GCONI is used to specify group injection controls.

FORMAT:

```
*GCONI  group_list
(*MAX)          (*STG)          value          (*STOP)
                (*STW)                         (*CONT)
                (*BHG)
                (*BHW)
                (*GMP)
                (*WMP)
                (*GCP)
                (*WCP)
(*GTARGET)      (*STG)          (value)
                (*STW)          (*NONE)
                (*BHG)
                (*BHW)
                (*GMP)
                (*WMP)
                (*GCP)
                (*WCP)
 (*VREP)        (*GAS)          vrep_frac
                (*WATER)
                (*GMKUP)
                (*WMKUP)
(*VREFP)        (*GAS)          vrefp_frac
                (*WATER)
(*RECYCLE)      (*GAS)          recyc_frac
                (*WATER)        recyc_frac   make_up_volume
                (*RECFRC)       sep_comp_recyc_fracs {Number of entries should
                                equal the number of components}
                (*RATE)         comp_make_up_vols {Number of entries should equal
                                the number of components}
(*PMAINT)       (*GAS)          (*PMSECT)  'sector_name'
                (*WATER)        (*PMTARG)  p_targ
                                (*PMCOEF)  c1 c2 c3
                                (*PMMAXR) d1 d2
```

DEFINITIONS:

group_list

Are the groups to which the following constraints apply. The wells that are connected to each group must already have been specified using the *WELL

keyword. The injection targets are apportioned using one of the available apportionment methods specified by *APPOR-METHOD.

*MAX

Specifies that the constraint is a maximum constraint which is checked for violations. This value becomes an injection rate target for the group only as the result of a violation with the *CONT action.

*GTARGET

This subkeyword specifies a target injection rate for the group. The specified stream rate is set as a target to be met by the group. If another constraint with action *CONT is violated, the rate target shifts to that constraint and the target set by the *GTARGET keyword ceases to have an effect. There is no action associated with *GTARGET since a target is not checked for violation.

*STG

This subkeyword identifies a surface gas rate (m^3/day | SCF/day) maximum or target. Zero rates are allowed and have the same effect as shutting in all the gas injection wells connected to that group.

*STW

This subkeyword identifies a surface water rate (m^3/day | Stb/day) maximum or target. Zero rates are allowed and have the same effect as shutting in all the water injection wells connected to that group.

*BHG

This subkeyword identifies a reservoir gas rate (m^3/day | SCF/day) maximum or target. Zero rates are allowed and have the same effect as shutting in all the gas injection wells connected to that group.

*BHW

This subkeyword identifies a reservoir water rate (m^3/day | Stb/day) maximum or target. Zero rates are allowed and have the same effect as shutting in all the water injection wells connected to that group.

*GMP, *WMP

This subkeyword introduces a gas or water manifold pressure (kPa | psi) injection constraint. This subkeyword can only be entered if all of the listed groups have had gas or water injection identified as going through a manifold with the *MANIFOLD keyword.

*GCP, *WCP

This subkeyword introduces a gas or water compressor (surface) pressure (kPa | psi) injection constraint. This subkeyword can only be entered if all of the listed groups have had gas or water injection identified as going through a manifold with the *MANIFOLD keyword. Also, a hydraulics table for calculation of the gas or water manifold-to-surface pressure drop must have been identified for all of the listed groups with the *GPTABLE keyword.

*NONE

Remove any previously specified injection target (including *VREF, *VREFP, *RECYCLE and *PMAINT) for the indicated stream (gas or water). Existing maximum constraints are not affected.

*VREP

This subkeyword introduces a voidage fraction injection target. This indicates that the injection wells connected to this group inject such that the voidage created by the producers connected to this group is replaced. In this case *GAS or *WATER specifies which phase is to be injected to replace the voidage. If more than one phase is being injected to replace the voidage then there must be one *VREP keyword for each phase. These primary voidage replacement streams are handled independently. One make-up stream can be supplemented with *GMKUP or *WMKUP to meet a total voidage replacement fraction. One of *GAS, *WATER, *GMKUP, or *WMKUP must be present for each *VREP keyword.

*VREFP

This subkeyword introduces reference volume fraction injection target. This indicates that the injection wells connected to this group inject such that the volume of fluid created by the producers connected to this group is replaced at the same reference condition. The reference pressure is the field hydrocarbon pore-volume weighted average, and the reference temperature is the field bulk-volume weighted average. In this case *GAS or *WATER specifies which phase is to be injected to replace the reference volume. If more than one phase is being injected then there must be one *VREFP keyword for each phase. These reference volume replacement streams are handled independently.

*RECYCLE

This subkeyword introduces a recycling injection target. This indicates that the injection wells connected to this group inject such that the phase produced by the production wells connected to this group as specified by *GAS or *WATER is recycled (reinjected) into the reservoir at the same surface conditions specified for the injectors. Considerable flexibility has been added to the specification of recycling injection targets through the keywords *GPRODGROUP, *WPRODGROUP, *GSALESFRAC, *GSALESMAXR,*GFUELFRAC, *GFUELMAXR, *WMKMAXR, *WMKUPTO, *GMKMAXR, *GMKUPTO, *GMKCOMP, *WRECYMAXR, *GRECYMAXR, *GSALESMASK, *GFUELMASK, and *GRECYMASK. It is recommended that the *RECYCLE subkeyword of *GCONI be used to specify water and gas recycling target fractions and that all the rest of the recycling target specification be done through those keywords. For backward compatibility the sub-subkeywords *RECFRC and *RATE of *RECYCLE are still supported by GEM.

*PMAINT

This subkeyword specifies that the group injection rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector (*PMSECT) at a desired level (*PMTARG).

*GAS, *WATER

Specifies that the gas or water phase to that is to be injected for voidage replacement, reference volume replacement, recycle or pressure maintenance.

*GMKUP, *WMKUP

Specifies that gas or water phase is the make-up stream supplemented to meet the total voidage replacement fraction.

*PMSECT

Introduces a single sector identified by the name ‘sector_name’ whose average hydrocarbon pore-volume pressure is to be maintained. ‘sector_name’ must be 16 characters maximum and must have already been defined in the RESERVOIR DESCRIPTION section in the input data. Defaulted to full ‘FIELD’.

*PMTARG

Introduces the targeted average hydrocarbon pore-volume pressure (kPa | psi) for the sector. The value p_targ must be a real number larger than one atmosphere. Defaulted to the current sector pressure.

*PMCOEF

Introduces the control coefficients used in the pressure control strategy. The values (c1, c2, c3) must be non-negative real numbers. Defaulted to the internally estimated values.

*PMMAXR

Introduces the maximum surface rate of the specified injection stream that is available for the pressure maintenance. It is the summation of two sources: a fraction (d1) of the total produced (from the same group) and a make-up rate (d2). The values must be non-negative real numbers. The fraction d1 is nondimensional and defaulted to 0. The make-up rate d2 is (m3/day | Stb/day) for water and (m3/day | SCF/day) for gas, and is defaulted to 1.0e+20.

value

Constraint value. Units are specified with the individual constraint subkeywords.

vrep_frac

When the voidage replacement subkeyword is used (*VREP) the vrep_frac is the voidage replacement ratio. A ratio of 1.0 indicates complete voidage replacement by the specified phase. When several values are entered (one for each of several phases) then the vrep_frac entered for a non make-up phase is applied as a target for that phase independently of the other specified values.

vrefp_frac

When the reference volume replacement subkeyword is used (*VREFP) the vrefp_frac is the reference volume replacement ratio. A ratio of 1.0 indicates complete reference volume replacement by the specified phase. When several values are entered (one for each of several phases) then the

vrefp_frac entered for a phase is applied as a target for that phase independently of the other specified values.

recyc_frac

When the recycle subkeyword is used (*RECYCLE) the recyc_frac is the recycled fraction of the indicated surface stream, which is imposed as a group injection target. A fraction of 1.0 indicates complete recycling of the surface stream. If the group producing the recycled fluid is named as the producing group for several different re-injecting groups using the *GPRODGROUP or *WPRODGROUP keywords, then the apportionment of the producing group's recycled fluid among the reinjecting groups must be set with recyc_frac. For example, if it is meant that all of the producing group's fluid is to be reinjected, and three groups on recycling targets reinject the fluid, the recyc_frac of the three injecting groups should sum to 1.

make_up_volume

The amount of make-up water (m^3/day | BBL/day) to be injected with the recycled water. Note : GEM still supports this way of entering the water make-up volumetric rate, but an alternate and more flexible method of specifying water make-up rates has been introduced using the keywords *WMKMAXR and *WMKUPTO. Please see the pages of this manual describing that keyword.

*RECFRC

This subkeyword introduces the component recycling fractions. Note : this subkeyword is still supported by GEM, but an alternate method of entering component recycling fractions has been introduced using the *GRECYMASK keyword. Please see the page of this manual describing that keyword.

*RATE

This subkeyword introduces the component make-up rates for hydrocarbon injection. Note : this subkeyword is still supported by GEM, but an alternate and more flexible method of entering gas make-up rates has been introduced using the keywords *GMKCOMP, *GMKMAXR, and *GMKUPTO. Please see the pages of this manual describing these keywords.

sep_comp_recyc_fracs

Fractions reducing the pooled separator gas molar component injection rates determined from the gas component rates of the cycling producers and the specified gas recycling fraction. One value must be entered for each of the nc hydrocarbon components. Note : GEM still supports this method of entering the component recycling fractions, but an alternate entry method using the *GRECYMASK keyword has been introduced. Please see the page in this manual describing that keyword. The default value for all nc of the sep_comp_recyc_fracs is 1.0, corresponding to no component-based reduction of the molar injection rates.

comp_make_up_vols

These numbers determine the surface volumetric rate and the composition of the injected make-up hydrocarbon. nc numbers must be entered, where nc is the number of hydrocarbon components. The volumetric make-up rate is the sum of the nc numbers; the mole fraction of the m-th component in the make-up fluid is the m-th number divided by the sum of the nc numbers (m^3/d | ft^3/d). The default values for these volumetric rates are zero for all components. Note : this way of entering gas make-up rates is still supported by GEM, but an alternate and more flexible method of entering gas make-up rates has been introduced using the keywords *GMKCOMP, *GMKMAXR, and *GMKUPTO. Please see the pages of this manual describing these keywords.

*STOP

Action subkeyword indicating that if the constraint cannot be met then the simulation should be stopped.

*CONT

Action subkeyword indicating that the simulation continues with the violated constraint switched to target constraint.

DEFAULTS:

Optional keyword. Default is no injection constraints or targets for groups.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined, by appearing in the list directly following *GROUP in a *GROUP line or after the *ATTACHTO keyword on a *GROUP line, before it can be given any constraint values. Constraint types *GTARGET, *RECYCLE, *VREP and *PMAINT are exclusive in one timecard for the same injector stream (gas or water), and only the latest entry counts. If a group is assigned by *GCONI with the following dependent constraints: *VREP, *RECYCLE, or *PMAINT for any of the injection streams, such a group cannot be assigned by *GCONP with these constraints for group production. In addition, only one injection stream can be assigned with voidage makeup (*VREP *GMKUP/*WMKUP) or pressure maintenance (*PMAINT) for the same targeted group. Error message will be issued if the consistency of the dependent constraints is violated.

EXPLANATION:

*GCONI is used to specify maximum or target fluid injection rates for the group. *GCONI can also be used to specify voidage replacement, recycling or pressure maintenance targets. This allows group controls to adjust injection rates in response to production.

Examples :

```
*GCONI 'Group-1' 'Group-2'
*GTARGET *STG 3000000.
```

This sets stock tank gas injection targets of 3000000. units per day for Group-1 and Group-2.

```
*GCONI 'Group-1'
*GTARGET *STG 1000000.
*MAX  *GMP 2394.  *CONT
```

This sets a stock tank gas target for Group-1 of 1000000. units, and a maximum gas injection manifold pressure of 2394 units. If maintaining the targeted injection rate causes the maximum manifold pressure to be exceeded, the group will switch to a gas manifold pressure target and continue simulation. Group-1 must have had gas injection specified with a *MANIFOLD keyword and must have a hydraulics table identified for the manifold-to-compressor pressure drop with the *GPTABLE keyword.

Recycling: The following keywords supplement *GCONI and add flexibility to recycling target specification, making recycling data entry more straightforward: *GPRODGROUP, *WPRODGROUP, *GSALESFRAC, *GSALESMAXR, *GFUELFRAC, *GFUELMAXR, *WMKMAXR, *WMKUPTO, *GMKMAXR, *GMKUPTO, *GMKCOMP, *WRECYMAXR, *GRECYMAXR, *GSALESMASK, *GFUELMASK, and *GRECYMASK. Please see the pages in this manual describing these keywords.

A water recycling target volumetric rate is now calculated as follows in GEM:

water_injection_target = wrecyc + wmakeup
wmakeup = min(max_water_makeup, water_makeupto - wrecyc)
wrecyc = min(fwrecyc*wprod, max_water_recycling_rate)

In the above, max_water_makeup is the quantity entered after the water recycling fraction fwrecyc on the *RECYCLE *WATER line or directly using the *WMKMAXR keyword. water_makeupto is a target injection rate which will be met unless a maximum make-up rate is exceeded. water_makeupto can be entered only using the *WMKUPTO keyword. wprod is the water production rate of, by default, the group upon which the water recycling injection target is imposed. However wprod can be taken as the water production rate of another group named using the *WPRODGROUP keyword. If no make-up target rate water_makeupto has been entered, wmakeup is set equal to the specified maximum make-up rate and the min operation in the second equation is not performed.

Gas recycling can involve different treatments of different hydrocarbon components and the expressions are correspondingly more complicated. In addition, the user is able to specify rates at which gas for fuel and sales are subtracted from the total gas production rate before the recycling target injection rate is calculated. A molar recycling injection target for component m is calculated in GEM as follows:

qinj(m) = qrecyc(m) + qmake_up(m)
qmake_up(m) = zmake_up(m) * qmolarmake-up
qmolarmake-up = make-upvolrate * make-up_surface_molar_density
make-upvolrate = min(max_make-up_rate , make-upto - recyc_vol_rate)
recyc_vol_rate = min(frecyc*(prodrate-fuel-sales) , maxrecyclerate)
fuel = min(frfuel*prodrate , max_fuel_rate)
fuel(m) = qprod(m)*(fuel/prodrate)
sales = min(frsales*(prodrate-fuel) , max_sales_rate)
sales(m) = qprod(m)*(sales/prodrate)

qrecyc(m) = frecyc*recfrc(m)*qrecyc0(m)*factor
qrecyc0(m) = qprod(m)-fuelmask(m)*fuel(m)-salesmask(m)*sales(m)
factor = recycvolrate/(frecyc*(prodrate-fuel-sales))

In the above, prodrate is the volumetric rate of gas production from the group upon which the recycling injection target is imposed or a producing group named using the *GPRODGROUP keyword. zmake_up(m) is the composition of the make-up gas, entered either indirectly using the *RATE subkeyword of *RECYCLE or directly for the targeted group using the *GMKCOMP keyword. max_makeup_rate is the maximum gas make-up rate entered indirectly using the *RATE subkeyword of *RECYCLE or directly using the *GMKMAXR keyword. make_upto is a target gas injection rate which make-up gas will be injected to meet until the maximum make-up rate is reached. make_upto can be entered only using the *GMKUPTO keyword. If the user does not specify a value for make_upto, the min operation in the fourth equation above is not performed and the gas make-up volumetric rate is set equal to the maximum rate.

frecyc is the gas recycling fraction entered on the *RECYCLE *GAS line. frfuel is a maximum fraction of the gas production rate which can burned as fuel; it can be entered only using the *GFUELFRAC keyword for the producing group. max_fuel_rate is an upper limit to the volumetric fuel rate. It can be entered only with the *GFUELMAXR keyword for the producing group. If no maximum fuel rate is entered the min operation in the sixth equation is not performed. max_recycle_rate is a volumetric maximum recycling injection rate for the targeted group which can be entered only with the *GRECYMAXR keyword; if not entered, the min operation in the fifth equation is not performed. qprod(m) is the molar gas production rate of component m in the group upon which the reinjection target is imposed or the producing group named using *GPRODGROUP.

frsales is the maximum fraction of the produced gas remaining after the fuel burn which can be diverted to sales. It can be entered only with the *GSALESFRAC keyword for the producing group. max_sales_rate is a upper limit to the rate of diversion of produced gas to sales; it can be entered with the *GSALESMAXR keyword for the producing group.

recfrc(m) is the component mask for gas reinjection; these numbers can be entered on the *RECYCLE *RECFRC line or with the *GRECYMASK keyword for the targeted group. The component mask reduces the nominal component rate; typically, component mask values are 1 or 0 for a component recycled or not. The default component mask values are 1 for all components. fuelmask(m) is the fuel component mask entered with the *GFUELMASK keyword for the producing group and salesmask(m) is the sales gas component mask entered with the *GSALESMASK keyword for the producing group.

Example: This is an example using voidage replacement. Half the voidage is replaced with water injection and half with gas injection. Note that voidage replacement is done at reservoir conditions and consequently surface rates will not agree with the ratio of 50%, even though the reservoir rates will be in a ratio of 50%.

```
*GCONI 'Group1'
  *VREP *SOLVENT 0.5
  *VREP *GAS     0.5
```

Example: Gas recycling. All separator gas from recycling producers in Group1 is re-injected through cycling injectors in Group1. The group injection target rate is apportioned among the group's open cycling injectors.

```
*GCONI 'Group1'
*RECYCLE *GAS 1.0
```

Example: Gas recycling. Injected fluid contains only components 1:3, in the same molar ratios as in the pooled separator gas from recycling producers in Group1. The volumetric rate of injection is 80% of the cycled gas production rate. Here nc = 6.

```
*GCONI 'Group1'
*RECYCLE *GAS 0.8
*RECYCLE *RECFRC 1.0  1.0  1.0  0.0  0.0  0.0
```

Example: Gas recycling using fuel, sales, and make-up rates.

```
*GPRODGROUP ‘Group1’ *FROM ‘Group2’
*GRECYMASK ‘Group1’ 1.0  1.0  1.0  0.0  0.0  0.0
*GMKMAXR ‘Group1’ 1.0D+7
*GMKUPTO ‘Group1’ 3.0D+7
*GFUELMAXR ‘Group2’ 5.0D+6
*GSALESMAXR ‘Group2’ 5.0D+6
*GMKCOMP ‘Group1’ 0.99  0.01  0.0  0.0  0.0  0.0
*GCONI 'Group1'
*RECYCLE *GAS 0.8
```

A number of apportionment methods (e.g. guide rate, instantaneous potential, priority ranking, and internal guide rate) can be used to distribute a group injection target among the contributing wells or groups through the keyword *APPOR-METHOD (which is described on a separate manual page).

Pressure Maintenance: *PMAINT specifies one special group control which instructs a group or groups to adjust the production rates (at reservoir condition) or injection rates (at the surface condition) in order to maintain the average hydrocarbon pore-volume pressure in a particular region (sector) at a desired level. The group runs essentially on a production (or injection) target rate Qtarg of bottom-hole (or surface) fluid in reference to that of the group injection (or production), Qref:

$$Q_{targ} = Q_{ref} \pm \Delta Q$$

where ΔQ is the bottom-hole (or surface) correction rate needed to reach or maintain a certain pressure. The sign ± designates for production (+) or injection (–) target. ΔQ has been set to zero for normal voidage replacement constraints for production control (*VREP under *GCONP) or reference volume replacement constraints for injection control (*VREFP under *GCONI) at the hydrocarbon pore-volume weighted average pressure and the bulk-volume weighted average temperature of the designated sector. However, setting $\Delta Q = 0$ may not exactly maintain the pressure due to the involvement of compressible fluids, and the different reservoir conditions under which Q_{targ} and Q_{ref} are calculated.

With the pressure control strategy, the correction volumetric rate is calculated as

$$\Delta Q = c_1\left(\left(P - P_{targ}\right) + c_2\left(\frac{dP}{dt}\right) + c_3\frac{1}{\Delta t}\int_{t-\Delta t}^{t}\left(P - P_{targ}\right)dt\right)$$

where P is the sector pressure at the current time t, and Δt is the integration interval over the last several timesteps. The empirical constants c_1, c_2 and c_3 penalize/minimize the object terms of (1) the difference between current pressure and the target, (2) the pressure derivative, and (3) the time-averaged pressure difference.

There is little to guide the choice of values of c_1, c_2, c_3. Heuristically, c_1 is a damping term which determines how quickly P will move toward P_targ, and controls the overall accuracy. Larger values of c_1 give a quicker approach to P_targ but may lead to pressure oscillations or even divergence if it is set too high. As P gets close to P_targ, the derivative term becomes significant and acts to decrease the magnitude of the time derivative of pressure to avoid overshooting. The larger c_2 is the less likely is overshoot, but the approach to P_targ is slowed. c_3 acts to prevent long-term persistence of pressures either above or below the target. Larger values of c_3 more quickly overcome differences from P_targ, at the possible cost of some oscillatory behavior.

The Well Management estimates and sets the above constants internally at the beginning of pressure maintenance if the subkeyword *PMCOEF is absent after *PMAINT, by assuming that the pressure follows an exponential decline to the set target under pure depletion of an isothermal point reservoir. It should be noted that there is no guarantee that such a rough estimate may work most favorably in general situations. Nevertheless, the user is recommended to run the simulator with the internally-set coefficients. By judging the pressure convergence behavior in terms of fluctuation and/or shifting, optimal values might then be achieved by tuning the internally-set values at possibly the same orders and input through subkeyword *PMCOEF. Currently the internal calculation of c_1, c_2 and c_3 does not take into account aquifer influx. If aquifer influx is strong it may be necessary for the user to optimize these coefficients.

Example:

```
*DATE 1985 1 1
     *GCONP ‘Grp-1’ *GTARGET *BHF 300.       ** (1)
     *GCONI ‘Grp-1’ *GTARGET *STW 150.       ** (2)
     *GCONI ‘Grp-1’ *GTARGET *STG 29000.     ** (3)

*DATE 1987 1 1
     *GCONI ‘Grp-1’ *PMAINT *GAS            ** (4)
```

Maintain sector ‘FIELD’ (default) average pressure at the current value as dated (default) by adjusting group ’Grp-1’ gas injection rate, using the internally-set control parameters (default). Line (3) is overridden by line (4). The production and injection rates are apportioned using the instantaneous potential method (default).

Fuel Consumption Fraction for Group Gas Recycling (Optional) *GFUELFRAC

PURPOSE:

*GFUELFRAC is used to specify a maximum fraction of the surface gas produced by a group which is available for consumption as fuel. Fuel consumption reduces the amount of gas available for re-injection by the same or another group. Gas recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *GFUELFRAC have no effect unless a group gas recycling target is currently in force.

FORMAT:

*GFUELFRAC group_name_list fuel_fractions

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will take the fuel fraction values specified after the list. The list need not be contained on a single line.

fuel_fractions
: A set of real values between 0 and 1 inclusive. They may all be zero. If only a single value is supplied, that value will be applied to all groups in the list. If more than one value is supplied, the number of fuel fractions must equal the number of group names in the list and the first fraction will be applied to the first group, etc. These fractions set, or set an upper limit for, the fraction of the volumetric rate of gas production which is consumed as fuel, and in this way affect the amount of gas available for re-injection under a gas recycling target. The fractions may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GFUELFRAC, then by default its fuel fraction value is zero and no gas is consumed as fuel unless a non-zero fixed fuel consumption rate is set with the *GFUELMAXR keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GFUELFRAC. Values outside the range [0,1] are rejected as being in error.

EXPLANATION:

For a full explanation of how the fuel fractions enter the calculation of a gas recycling injection target, please see the pages of this manual describing the GCONI keyword. It is important to note that the fuel fraction values are associated with the group producing the re-injected gas, and not with the group injecting it. If a reporting group is cited as the producing group with the GPRODGROUP keyword, then specify the name(s) of the reporting groups

with the GFUELFRAC keyword. Fuel consumption will be set at a volumetric rate of fuel_fraction*gas_production_rate unless a fixed maximum fuel rate has been set using the *GFUELMAXR keyword.

Example: Group1 acquires a fuel fraction of 0.10 and Group2 of 0.12.

```
*GFUELFRAC  'Group1' 'Group2'
               0.10    0.12
```

Here the fuel fraction values are, but need not have been, on a separate line from the group names.

Fuel Component Mask for Group Gas Recycling (Optional)

***GFUELMASK**

PURPOSE:

*GFUELMASK is used to specify component reduction factors for group fuel consumption, which reduces the amount of produced gas from a group available for reinjection by the same or another group. Gas recycling injection targets are specified for groups using the *GCONI keyword.

FORMAT:

*GFUELMASK group_name_list mask_values

DEFINITIONS:

group_name_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will take the single set of fuel component mask values specified after the list. The list need not be contained on a single line.

mask_values

A set of nc real values, where nc is the number of hydrocarbon components. The values must be between 0 and 1 inclusive, but they cannot all be zero. These fractions reduce the amount used for fuel selectively by component from the amount calculated volumetrically. Often the values are either 0 or 1, for components which are fully available for use as fuel or are not burned at all. The mask values can be, but need not be, on the same line as the last name in the group list. The mask values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GFUELMASK, then by default all of that group's mask values are 1.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GFUELMASK. Values outside the range [0,1] and a set of values summing to 0 are rejected as being in error.

EXPLANATION:

For a full explanation of how the fuel mask values enter the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the fuel mask values are associated with the group producing the re-injected gas, and not with the group injecting it. An initial molar fuel rate fuel(m) is calculated from the volumetric fuel rate and the composition of the produced gas. This is then reduced (or left unchanged) by multiplying by the fuel_mask_value(m). The reduced molar fuel rate is then subtracted from the total production rate of component m in surface gas.

Example: Group1 and Group2 acquire the specified fuel mask for a seven component hydrocarbon fluid description.

```
*GFUELMASK 'Group1’ ‘Group2'
   1.0  1.0  1.0  0.0  0.0  0.0  0.0
```

Here the fuel mask values are, but need not have been, on a separate line from the group names.

Maximum Fuel Consumption Rate for Group Gas Recycling (Optional) *GFUELMAXR

PURPOSE:

*GFUELMAXR is used to specify a maximum rate at which the surface gas produced by a group is made available for consumption as fuel. Fuel consumption reduces the amount of gas available for re-injection by the same or another group. Gas recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered using *GFUELMAXR have no effect unless a group gas recycling target is in force.

FORMAT:

*GFUELMAXR group_name_list max_fuel_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum fuel rates specified after the list. The list need not be contained on a single line.

max_fuel_rates
: A set of non-negative real values (m3/day | scf/day). Zero is accepted as a value but is interpreted as signifying that no maximum fuel rate is set. If only a single value is supplied, that value will be applied to all groups in the list. If more than one value is supplied, the number of maximum fuel rates must equal the number of group names in the list and the first rate will be applied to the first group, etc. These maximum rates set, or set an upper limit for, the volumetric rate of gas production consumed as fuel and in this way affect the amount of gas re-injected under a gas recycling target. The values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GFUELMAXR, then by default its maximum fuel rate is zero and no gas is consumed as fuel unless a non-zero fuel fraction is set with the *GFUELFRAC keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GFUELMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum fuel rates enter the calculation of a gas recycling injection target, please see the pages of this manual describing the GCONI keyword. It is important to note that the maximum fuel rate values are associated with the group producing the re-injected gas, and not with the group injecting it. If a reporting group is cited as the producing

group with the GPRODGROUP keyword, then specify the name(s) of the reporting groups with the GFUELMAXR keyword. Fuel consumption will be set at the entered maximum fuel rate unless a fuel fraction values has been set for the producing group *GFUELFRAC keyword. If both the fuel fraction and the maximum fuel rate are set to zero, then there is no fuel consumption. If only one of the fraction or maximum rate is entered then the entered quantity is applied. If both are entered then the smaller of the two fuel rates is used.

Example: Group1 acquires a maximum fuel rate of 3.0D+06 scf/day and Group2 of 3.5D+06 scf/day.

```
*GFUELMAXR  'Group1'    'Group2'
             3.0D+06     3.5D+06
```

Here the maximum fuel rates are, but need not have been, on a separate line from the group names.

Make-up Gas Composition for Group Gas Recycling (Optional) *GMKCOMP

PURPOSE:

*GMKCOMP is used to specify the composition of a gas injected as part of a group gas recycling target to supplement the recycled fluid. This information can also be specified in a somewhat less straightforward way using the *RECYCLE *RATE subkeywords of *GCONI when *RECYCLE *GAS is specified. The two methods of entering the make-up composition can over-write each other. Gas recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *GMKCOMP have no effect unless a group gas recycling injection target is currently in force.

FORMAT:

*GMKCOMP group_name_list mole_fractions

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will take the single set of mole fraction values specified after the list. The list need not be contained on a single line.

mole_fractions
: A set of nc real values, where nc is the number of hydrocarbon components. The values must be between 0 and 1 inclusive, but they cannot all be zero. These fractions specify the composition of make-up gas used to supplement gas re-injection. The mole fraction values can be, but need not be, on the same line as the last name in the group list. The mole fraction values may be spread over more than a single line.

DEFAULTS:

Optional keyword. No default. If a non-zero gas make-up rate is specified when no make-up composition has been specified, a terminating error and message are issued. Make-up rates are specified using the *RECYCLE *RATE subkeywords of *GCONI or the *GMKMAXR keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GMKCOMP. Values outside the range [0,1] and a set of values summing to 0 are rejected as being in error.

EXPLANATION:

For a full explanation of how the make-up composition enters the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the make-up composition is associated with the group re-injecting the gas, and not with the group producing it.

Example: Group1 and Group2 acquire the specified make-up gas composition for a seven-component hydrocarbon fluid description.

```
*GMKCOMP 'Group1’ ‘Group2'
   0.99  0.01  0.0  0.0  0.0  0.0  0.0
```

Here the fuel mask values are, but need not have been, on a separate line from the group names. If the entered values do not sum to 1 they will be normalized within the simulator.

Maximum Make-up Gas Rate for Group Gas Recycling (Optional) *GMKMAXR

PURPOSE:

*GMKMAXR is used to specify the maximum surface rate of make-up gas injected as part of a group gas recycling target to supplement the recycled fluid. This information can also be specified in a somewhat less straightforward way using the *RECYCLE *RATE subkeywords of *GCONI when *RECYCLE *GAS is specified. The two methods of entering the make-up rate can over-write each other. Gas recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *GMKMAXR have no effect unless a group gas recycling injection target is currently in force.

FORMAT:

*GMKMAXR group_name_list max_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum make-up rates specified after the list. The list need not be contained on a single line.

max_rates
: A set of non-negative maximum make-up rate values (m3/day | SCF/day). If only one rate value is entered it is applied to all of the groups listed. If more than one rate value is entered, then the number of rates entered must equal the number of groups in the list and the first rate is assigned to the first group, etc. Zero maximum rates may be entered, but they are interpreted as meaning that the group's make-up rate is controlled completely by the "make-up to" target entered using the *GMKUPTO keyword. If neither a maximum rate nor a "make-up to" target is specified, then the make-up rate is taken to be zero. The maximum rate values can be, but need not be, on the same line as the last name in the group list. The rate values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If not specified, make-up rate is controlled by target rates entered using the *GMKUPTO keyword; if no target rate is entered either, then no make-up gas is injected. Maximum make-up rates are also specified using the *RECYCLE *RATE subkeywords of *GCONI.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GMKMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum make-up rate enters the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the maximum make-up rate is associated with the group re-injecting the gas, and not with the group producing it.

Example: Group1 and Group2 both acquire the single specified maximum make-up gas injection rate.

```
*GMKMAXR 'Group1’ ‘Group2'  1.0D+7
```

Here the maximum rate value is, but need not have been, on the same line as the group names.

Gas Make-up Target for Group Gas Recycling (Optional)

***GMKUPTO**

PURPOSE:

*GMKUPTO is used to specify a total group surface gas injection rate (also referred to as a “make-up to” rate) which make-up gas, in general up to a certain maximum rate, will supplement the recycled gas to meet. Well constraint values may prevent the rate specified using *GMKUPTO from being met.

The maximum gas make-up rate can be specified either with the *GMKMAXR keyword or the *RECYCLE *RATE subkeywords of *GCONI. The total injection rate can be entered only with *GMKUPTO. The rates entered here have an effect only if a gas recycling injection target is in force. Gas recycling injection targets are specified for groups using the *GCONI keyword. The total injection rate specified here differs from a group STG target in that the target entered here will not be met if the make-up gas would have to exceed a specified maximum rate in order to meet the total rate.

FORMAT:

*GMKUPTO group_name_list total_gas_rates

DEFINITIONS:

group_name_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the total gas rates specified after the list. The list need not be contained on a single line.

total_gas_rates

A set of non-negative total injected gas rate values (m3/day | SCF/day). If only one rate value is entered it is applied to all of the groups listed. If more than one rate value is entered, then the number of rates entered must equal the number of groups in the list and the first rate is assigned to the first group, etc. Zero total rates may be entered, but they are interpreted as meaning that the group’s gas make-up rate is controlled completely by the maximum make-up rate entered using the *GMKMAXR keyword. If neither a maximum rate nor a “make-up to” target is specified, then the make-up rate is taken to be zero. The total gas rate values can be, but need not be, on the same line as the last name in the group list. The rate values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If not specified, the gas make-up rate is controlled by maximum rates entered using the *GMKMAXR keyword; if no maximum rate is entered either, then no make-up gas is injected.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GMKUPTO. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the total gas injection rate enters the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the “make-up to” rate is associated with the group re-injecting the gas, and not with the group producing it. A specified total injected gas rate has no effect unless a group gas recycling target is currently in force (see the *GCONI entry in this manual.). The group gas injection rate will be set to the total gas rate unless this total rate exceeds the injection rate of recycled gas by more than a maximum make-up rate, in which case the total gas injection rate is set to the recycling rate plus the maximum make-up rate.

Example: Group1 and Group2 both acquire the single specified total gas injection rate.

```
*GMKUPTO 'Group1’ ‘Group2'  5.0D+7
```

Here the total gas value is, but need not have been, on the same line as the group names.

Gas Producing Group for Group Recycling / Voidage Replacement (Optional) *GPRODGROUP

PURPOSE:

*GPRODGROUP is used to specify the group from which gas production rates and gas composition will be taken to calculate a gas recycling injection target or a voidage replacement by gas target specified for another group or groups using the *GCONI keyword.

FORMAT:

*GPRODGROUP group_name_list *FROM ‘producing_group_name'

DEFINITIONS:

group_name_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will have gas recycling injection targets or voidage replacement by gas targets based on gas production/total production from the named producing group. The list need not be contained on a single line.

*FROM

This subkeyword indicates that all groups named in the list preceding *FROM are to have their recycling injection targets or voidage replacement by gas targets calculated using the gas/total production from the single group with the name ‘producing_group_name’. *FROM may occur on the same line as the last group in the list but need not. It is valid for *FROM to be on a new line.

‘producing_group_name’

A single valid group name, enclosed in single quotes. This group determines gas rates and composition for recycling injection targets or total production rates for voidage replacement targets imposed upon the listed groups. This group name may be but need not be on the same line as the *FROM subkeyword. This group may be a named reporting group. Since reporting groups allow for fractional well membership, use of a reporting group as a contributing or nominating group for recycling or voidage replacement enables imposition of injection targets based on pattern type flood.

DEFAULTS:

Optional keyword. If a group’s name never appears in a group list following *GPRODGROUP, then by default that group’s own gas production rate and composition or groups total production rates are reservoir conditions are used to compute gas recycling targets or voidage replacement by gas targets imposed on the group.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GPRODGROUP.

The fraction of gas recycled to each group in the ‘group_name_list’ must sum to no more than 1.0. It is possible to reinject more gas than is produced from the “gas producing” group if this is not ensured (The fraction of gas recycled for each group is entered on its *GCONI keyword after the *RECYCLE *GAS sub-keyword).

EXPLANATION:

The imposition of a recycling rate target involves using the gas production rates and composition of one group to determine the gas injection rates and composition in the same or another group. Recycling targets can involve some additional parameters such as sales gas rate and fuel gas rate which are also associated with the producing group rather than with the injecting group. Please consult the pages of this manual describing the *GCONI keyword for a full explanation of the calculation of gas reinjection targets. Further information is also available from the manual pages for the other gas recycling keywords, which are *GSALESFRAC, *GSALESMAXR, *GFUELFRAC, *GFUELMAXR, *GMKMAXR, *GMKUPTO, *GMKCOMP, *GRECYMAXR, *GSALESMASK, *GFUELMASK, and *GRECYMASK.

Example: Group1 and Group2 have their gas recycling injection targets calculated using the gas produced by Group3.

```
*GPRODGROUP 'Group1’ ‘Group2' *FROM ‘Group3’
```

The imposition of a voidage replacement target involves using the total production rates at reservoir conditions of one group to determine the gas injection rates in the same or another group. Please consult the pages of this manual describing the *GCONI keyword for a full explanation of the calculation of gas targets based on voidage replacement. Use of reporting groups which are allow fractional membership can be used to design group injection targets predicated on recycling or voidage replacement where a given injector is assigned only a fraction of the total production of a producing well. For example consider a central injector INJ-1 flanked by four producers, PROD-1, PROD-2, PROD-3, PROD-4 and this pattern repeats with other 5 spot well arrangement. To assign a quarter of production to injection well in a 100% voidage replacement by gas scenario, create a group with just the injector and a reporting group with just the producers with one quarter membership, for example

```
*GROUP 'GRP-1' *ATTACHTO 'FIELD'
*WELL 1 'INJ-1'   *ATTACHTO 'GRP-1'
*WELL 2 ‘PROD-1’
*WELL 3 ‘PROD-2’
*WELL 4 ‘PROD-3’
*WELL 5 ‘PROD-4’
*GCONI   'GRP-1'
      *VREP *GAS   1.0     **Voidage replacement injection target
*REPORTING-GROUP 'RP-1' ‘PROD-1’ ‘PROD-2’ ‘PROD-3’ ‘PROD-4’
                                          0.25 0.25 0.25 0.25
*GPRODGROUP 'GRP-1' *FROM 'RP-1'

** Calculate gas target for GRP-1 based on total production of reporting
group ‘RP-1’ and   ** a voidage replacement fraction specified with
GCONI
```

Injection Group for Group Recycling/Voidage Replacement (Optional) *VRI_GROUP

PURPOSE:

*VRI_GROUP is used to specify the injection group from which gas/water/voidage injection rates will be taken to calculate a production recycling or voidage replacement target for a given group or groups. The actual recycling or voidage replacement target is specified using the keyword GCONP.

FORMAT:

*VRI_GROUP production_group_name_list *FROM injection_group_name'

DEFINITIONS:

Production_group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will have a production recycling or voidage replacement targets based on injection by the named injecting group. The list need not be contained on a single line.

*FROM
: This subkeyword indicates that all groups named in the list preceding *FROM are to have their production recycling targets or production voidage replacement by targets calculated using the injection rates by the single group with the name 'injection_group_name'. *FROM may occur on the same line as the last group in the list but need not. It is valid for *FROM to be on a new line.

'injection_group_name'
: A single valid group name, enclosed in single quotes. This group determines injection rates dictating production group recycling or production voidage replacement targets imposed upon the listed groups. This group name may be but need not be on the same line as the *FROM subkeyword. This group may be a named reporting group. Since reporting groups allow for fractional well membership, use of a reporting group as a contributing or nominating group for injection rate calculation enables imposition of production targets (recycling or voidage replacement) based on pattern type floods (for example for a reverse 5 spot pattern).

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *VRI_GROUP, then by default that group's own injection rates are used to compute production recycling or voidage replacement targets imposed on the group.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *VRI_GROUP.

EXPLANATION:

Example: Group1 and Group2 have their gas recycling production targets calculated using the gas injected by Group3.

```
*VRI_GROUP 'Group1' 'Group2' *FROM 'Group3'
```

The imposition of a group production recycling or voidage replacement target involves using the injection rates at reservoir conditions of one group to determine the production rates in the same or another group. Please consult the pages of this manual describing the *GCONP keyword for a full explanation of the calculation of production targets based on recycling or voidage replacement. Use of reporting groups which are allow fractional membership can be used to design group production targets predicated on injection rate replacement where a given producer is assigned only a fraction of the total injection of an injecting well. For example consider a central injector PROD-1 flanked by injectors, INJ-1, INJ-2, INJ-3, INJ-4 and this pattern repeats with other inverse 5 spot well arrangement. To assign a quarter of injection to production well in a 100% gas recycling scenario, create a group with just the producer and a reporting group with just the injectors with one quarter membership, for example

```
*GROUP 'GRP-1' *ATTACHTO 'FIELD'
*WELL 1 PROD-1'   *ATTACHTO 'GRP-1'
*WELL 2 'INJ-1'
*WELL 3 'INJ-2'
*WELL 4 'INJ-3'
*WELL 5 'INJ-4'
*GCONP   'GRP-1'
      *RECYCLE *GAS   1.0     **Voidage replacement injection target
*REPORTING-GROUP 'RP-1' 'INJ-1' 'INJ-2' 'INJ-3' 'INJ-4'
                                                 0.25 0.25 0.25 0.25
*VRI_GROUP 'GRP-1' *FROM 'RP-1'

** Calculate a production recycling gas target for GRP-1 based on total
injection of reporting ** group 'RP-1' and recycling fraction specified
with GCONP
```

Recycled Gas Component Mask for Group Gas Recycling (Optional) *GRECYMASK

PURPOSE:

*GRECYMASK is used to specify component reduction factors for group re-injection of produced gas remaining after fuel and sales requirements have been satisfied. Gas recycling injection targets are specified for groups using the *GCONI keyword. The *GRECYMASK keyword affects injection rates only if a group recycling injection target is currently in effect.

FORMAT:

*GRECYMASK group_name_list mask_values

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will take the single set of recycling component mask values specified after the list. The list need not be contained on a single line.

mask_values
: A set of nc real values, where nc is the number of hydrocarbon components. The values must be between 0 and 1 inclusive, but they cannot all be zero. These fractions reduce the amount reinjected selectively by component from the amount calculated volumetrically. Often the values are either 1 or 0, for components which are fully available for reinjection or are not reinjected at all. The mask values can be, but need not be, on the same line as the last name in the group list. The mask values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GRECYMASK, then by default all of that group's mask values are 1.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GRECYMASK. Values outside the range [0,1] or a set of values summing to 0 are rejected as being in error. The recycling mask can also be entered using the *RECYCLE *RECFRC subkeywords of the *GCONI keyword, and the two different ways of entering this mask can over-write each other.

EXPLANATION:

For a full explanation of how the recycling mask values enter the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the recycling mask values are associated with the group re-injecting the gas, and not with the group producing it. An initial molar re-injection rate qrecyc(m) is calculated for component m from the volumetric sales rate and the composition of the produced gas and the specified global re-injection fraction. This is then reduced (or left

unchanged) by multiplying by the recyc_mask_value(m). The reduced molar recycling rate is then set as the group re-injection target.

Example: Group1 and Group2 acquire the specified recycling mask for a five-component hydrocarbon fluid description.

```
*GRECYMASK 'Group1’ ‘Group2' 1.0 1.0 0.0 0.0 0.0
```

Here the recycling mask values are, but need not have been, on the same line as are the group names. Both groups acquire the single five-number component mask.

Maximum Re-injection Rate for Group Gas Recycling (Optional) *GRECYMAXR

PURPOSE:

*GRECYMAXR is used to specify a maximum rate at which gas is re-injected under a group recycling target. Gas recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *GRECYMAXR have no effect unless a group gas recycling injection target is currently in force.

FORMAT:

*GRECYMAXR group_name_list max_recyc_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum recycling rates specified after the list. The list need not be contained on a single line.

max_recyc_rates
: A set of non-negative real values (m3/day | scf/day). Zero is accepted as a value but is interpreted as signifying that no maximum recycling rate is set. If only a single value is supplied, that value will be applied to all groups in the list. If more than one value is supplied, the number of maximum recycling rates must equal the number of group names in the list and the first rate will be applied to the first group, etc. These maximum rates set, or set an upper limit for, the volumetric rate at which produced gas is re-injected under a group gas recycling target. The values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GRECYMAXR, then by default its maximum gas recycling rate is set to zero, which is interpreted to mean that there is no fixed maximum imposed upon the recycling injection rate.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GRECYMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum recycling rates enter the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the maximum recycling rate values are associated with the group re-injecting the gas, and not with the group producing it. The recycling volumetric injection rate is set principally by the recycling fraction entered with the *RECYCLE *GAS

line following *GCONI, but if a maximum has been imposed with the *GRECYMAXR keyword, the re-injection target is not allowed to exceed this maximum.

Example: Group1 acquires a maximum recycling injection rate of 3.0D+07 scf/day and Group2 of 3.5D+07 scf/day.

```
*GRECYMAXR  'Group1'    'Group2'
              3.0D+07     3.5D+07
```

Here the maximum recycling rates are, but need not have been, on a separate line from the group names.

Sales Rate Fraction for Group Gas Recycling (Optional)

***GSALESFRAC**

PURPOSE:

*GSALESFRAC is used to specify a maximum fraction of the surface gas produced by a group (less the amount consumed as fuel) which is available for sale. Diversion to sales reduces the amount of gas available for re-injection by the same or another group. Gas recycling injection targets are specified for groups using the *GCONI keyword. Numbers entered under the *GSALESFRAC keyword have no effect unless a group gas recycling injection target is currently in force.

FORMAT:

*GSALESFRAC group_name_list sales_fractions

DEFINITIONS:

group_name_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the sales fractions specified after the list. The list need not be contained on a single line.

sales_fractions

A set of real values between 0 and 1 inclusive. They may all be zero. If only a single value is supplied, that value will be applied to all groups in the list. If more than one value is supplied, the number of sales fractions must equal the number of group names in the list and the first fraction will be applied to the first group, etc. These fractions set, or set an upper limit for, the fraction of the volumetric rate of produced gas, less the fuel consumption rate, which is re-injected under a gas recycling target. The fractions may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GSALESFRAC, then by default its sales fraction value is zero and no gas is diverted for sales unless a non-zero fixed sales rate is set with the *GSALESMAXR keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GSALESFRAC. Values outside the range [0,1] are rejected as being in error.

EXPLANATION:

For a full explanation of how sales fractions enter the calculation of a gas recycling injection target, please see the pages of this manual describing the GCONI keyword. It is important to note that the sales fraction values are associated with the group producing the re-injected gas, and not with the group injecting it. If a reporting group is cited as the producing group with

the GPRODGROUP keyword, then specify the name(s) of the reporting groups with the GSALESFRAC keyword. Sales will be set at a volumetric rate of

sales_fraction*gas_production_rate

unless a fixed maximum sales rate has been set using the *GSALESMAXR keyword.

Example: Group1 acquires a sales fraction of 0.20 and Group2 of 0.24.

```
*GSALESFRAC  'Group1' 'Group2'
               0.20    0.24
```

Here the sales fraction values are, but need not have been, on a separate line from the group names. Any group specifying Group1 or Group2 as its recycling production group with the *GPRODGROUP keyword will have the sales fractions set here used in calculating a gas recycling injection target set using the *GCONI keyword.

Sales Component Mask for Group Gas Recycling (Optional)

***GSALESMASK**

PURPOSE:

*GSALESMASK is used to specify component reduction factors for group gas diverted for sales, which reduces the amount of produced gas from a group available for reinjection by the same or another group. Gas recycling injection targets are specified for groups using the *GCONI keyword.

FORMAT:

*GSALESMASK group_name_list mask_values

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will take the single set of sales component mask values specified after the list. The list need not be contained on a single line.

mask_values
: A set of nc real values, where nc is the number of hydrocarbon components. The values must be between 0 and 1 inclusive, but they cannot all be zero. These fractions reduce the amount diverted to sales selectively by component from the amount calculated volumetrically. Often the values are either 1 or 0, for components which are fully available for sales or are not sold at all. The mask values can be, but need not be, on the same line as the last name in the group list. The mask values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GSALESMASK, then by default all of that group's mask values are 1.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GSALESMASK. Values outside the range [0,1] or a set of values summing to 0 are rejected as being in error.

EXPLANATION:

For a full explanation of how the sales mask values enter the calculation of a gas recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the sales mask values are associated with the group producing the re-injected gas, and not with the group injecting it. An initial molar sales rate sales(m) is calculated from the volumetric sales rate and the composition of the produced gas. This is then reduced (or left unchanged) by multiplying by the sales_mask_value(m). The reduced molar sales rate is then subtracted from the total production rate of component m in surface gas less the fuel burn rate for component m.

Example: Group1 and Group2 acquire the specified sales mask for a six-component hydrocarbon fluid description.

```
*GSALESMASK 'Group1’ ‘Group2'
   1.0  1.0  1.0  0.0  0.0  0.0
```

Here the sales mask values are, but need not have been, on a separate line from the group names. Both groups acquire the single six-number component mask.

Maximum Sales Rate for Group Gas Recycling (Optional)

***GSALESMAXR**

PURPOSE:

*GSALESMAXR is used to specify a maximum rate at which the surface gas produced by a group is made available for sales. Diversion to sales reduces the amount of gas available for re-injection by the same or another group. Gas recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *GSALESMAXR have no effect unless a group recycling target is currently in force.

FORMAT:

*GSALESMAXR group_name_list max_sales_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum sales rates specified after the list. The list need not be contained on a single line.

max_sales_rates
: A set of non-negative real values (m3/day | scf/day). Zero is accepted as a value but is interpreted as signifying that no maximum sales rate is set. If only a single value is supplied, that value will be applied to all groups in the list. If more than one value is supplied, the number of maximum sales rates must equal the number of group names in the list and the first rate will be applied to the first group, etc. These maximum rates set, or set an upper limit for, the volumetric rate of gas production diverted to sales and in this way affect the amount of gas re-injected under a gas recycling target. The values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *GSALESMAXR, then by default its maximum sales rate is zero and no gas is sold unless a non-zero sales fraction is set with the *GSALESFRAC keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *GSALESMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum sales rate enters the calculation of a gas recycling injection target, please see the pages of this manual describing the GCONI keyword. It is important to note that the maximum sales rate values are associated with the group producing the re-injected gas, and not with the group injecting it. If a reporting group is cited as the producing group with the GPRODGROUP keyword, then specify the name(s) of the

reporting groups with the GSALESMAXR keyword. Sales will be set at the entered maximum rate unless a sales fraction value has been set for the producing group with the *GFUELFRAC keyword. If both the sales fraction and the maximum sales rate are set to zero, then there are no sales. If only one of the fraction or maximum rate is entered then the entered quantity is applied. If both are entered then the smaller of the two sales rates is used.

Example: Group1 acquires a maximum sales rate of 6.0D+06 scf/day and Group2 of 7.0D+06 scf/day.

```
*GSALESMAXR  'Group1'    'Group2'
             6.0D+06     7.0D+06
```

Here the maximum sales rates are, but need not have been, on a separate line from the group names.

Maximum Water Make-up Rate for Group Water Recycling (Optional) *WMKMAXR

PURPOSE:

*WMKMAXR is used to specify the maximum surface rate of make-up water injected as part of a group water recycling target to supplement the recycled water. This information can also be specified using the *RECYCLE *WATER subkeywords of *GCONI. The two methods of entering the make-up rate can over-write each other. Water recycling injection targets are specified for groups using the *GCONI keyword.

FORMAT:

*WMKMAXR group_name_list max_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum make-up rates specified after the list. The list need not be contained on a single line.

max_rates
: A set of non-negative maximum water make-up rate values (m3/day | Stb/day). If only one rate value is entered it is applied to all of the groups listed. If more than one rate value is entered, then the number of rates entered must equal the number of groups in the list and the first rate is assigned to the first group, etc. Zero maximum rates may be entered, but they are interpreted as meaning that the group's water make-up rate is controlled completely by the "make-up to" target entered using the *WMKUPTO keyword. If neither a maximum rate nor a "make-up to" target is specified, then the make-up rate is taken to be zero. The maximum rate values can be, but need not be, on the same line as the last name in the group list. The rate values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If not specified, the water make-up rate is controlled by target rates entered using the *WMKUPTO keyword; if no target rate is entered either, then no make-up water is injected. Maximum make-up rates are also specified using the *RECYCLE *WATER subkeywords of *GCONI.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *WMKMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum water make-up rate enters the calculation of a water recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the maximum water make-up rate is associated with the group re-injecting the water, and not with the group producing it.

Example: Group1 and Group2 both acquire the single specified maximum make-up water injection rate.

```
*WMKMAXR 'Group1’ ‘Group2'  300.
```

Here the maximum rate value is, but need not have been, on the same line as the group names.

Water Make-up Target for Group Water Recycling (Optional)

***WMKUPTO**

PURPOSE:

*WMKUPTO is used to specify a total group water injection rate which make-up water, in general up to a certain maximum rate, will supplement the recycled water to meet. The maximum make-up rate can be specified either with the *WMKMAXR keyword or the *RECYCLE *WATER subkeywords of *GCONI. The total injection rate can be entered only with *WMKUPTO. The rates entered here have an effect only if a water recycling injection target is in force. Water recycling injection targets are specified for groups using the *GCONI keyword. The total rate specified here differs from a group STW target in that the target entered here will not be met if the make-up water would have to exceed a specified maximum rate in order to meet the total rate.

FORMAT:

*WMKUPTO group_name_list total_water_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the total water rates specified after the list. The list need not be contained on a single line.

total_water_rates
: A set of non-negative total injected water rate values (m3/day | Stb/day). If only one rate value is entered it is applied to all of the groups listed. If more than one rate value is entered, then the number of rates entered must equal the number of groups in the list and the first rate is assigned to the first group, etc. Zero total rates may be entered, but they are interpreted as meaning that the group's water make-up rate is controlled completely by the maximum make-up rate entered using the *WMKMAXR keyword. If neither a maximum rate nor a "make-up to" target is specified, then the make-up rate is taken to be zero. The total water rate values can be, but need not be, on the same line as the last name in the group list. The rate values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If not specified, the water make-up rate is controlled by maximum rates entered using the *WMKMAXR keyword; if no maximum rate is entered either, then no make-up water is injected.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *WMKUPTO. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the total water injection rate enters the calculation of a water recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the “make-up to” rate is associated with the group re-injecting the water, and not with the group producing it. A specified total injected water rate has no effect unless a group water recycling target is currently in force (see the *GCONI entry in this manual.). The group water injection rate will be set to the total water rate unless this total rate exceeds the injection rate of recycled water by more than a maximum make-up rate, in which case the total water injection rate is set to the recycling rate plus the maximum make-up rate.

Example: Group1 and Group2 both acquire the single specified total water injection rate.

```
*WMKUPTO 'Group1’ ‘Group2'  300.
```

Here the total water value is, but need not have been, on the same line as the group names.

Water Producing Group for Group Recycling (Optional) / Production Group for Voidage Replacement by Water Injection *WPRODGROUP

PURPOSE:

*WPRODGROUP is used to specify the group from which water production rate in case of recycling or total production rate in case of voidage replacement will be taken to calculate a water recycling or voidage replacement by water injection target specified for another group or groups using the *GCONI keyword.

FORMAT:

*WPRODGROUP group_name_list *FROM 'producing_group_name'

DEFINITIONS:

group_name_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will have water recycling injection or voidage replacement by water targets imposed based on water production in case of recycling or total production in case of voidage replacement from the named producing group. The list need not be contained on a single line.

*FROM

This subkeyword indicates that all groups named in the list preceding *FROM are to have their water recycling or voidage replacement by water injection targets calculated using the water production in case of recycling or total production in case of voidage replacement from the single group with the name 'producing_group_name'. *FROM may occur on the same line as the last group in the list but need not. It is valid for *FROM to be on a new line.

'producing_group_name'

A single valid group name, enclosed in single quotes. This group determines water rates for recycling injection or total volumes produced at reservoir conditions for voidage replacement by water injection targets imposed upon the listed groups. This group name may be but need not be on the same line as the *FROM subkeyword.

This nominating or contributing group can also be a reporting group. Reporting groups allow fractional membership of wells enabling targets based on pattern floods to be specified. Please refer to keywords GPRODGROUP for more details.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *WPRODGROUP, then by default that group's own water/total production rate is used to compute water recycling or voidage replacement by water targets imposed on the group.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *WPRODGROUP.

The fraction of water recycled to each group in the ‘group_name_list’ must sum to no more than 1.0. It is possible to reinject more water than is produced from the “water producing” group if this is not ensured (The fraction of water recycled for each group is entered on its *GCONI keyword after the *RECYCLE *WATER sub-keyword).
The nominating or contributing group can also be a reporting group. Reporting groups allow fractional membership of wells enabling targets based on pattern floods to be specified. Please refer to keywords GPRODGROUP for more details.

EXPLANATION:

The imposition of a recycling rate/voidage replacement target involves using the water production rates in case of recycling or total production in case of voidage replacement of one group to determine the water injection rates in the same group or another group. Please consult the pages of this manual describing the *GCONI keyword for a full explanation of the calculation of water reinjection/voidage replacement targets. In the case of recycling, further information is also available from the manual pages for the other water recycling keywords, which are *WMKMAXR, *WMKUPTO, and *WRECYMAXR

Example 1:

Group1 and Group2 have their water recycling injection targets calculated using the water produced by Group3.

```
*WPRODGROUP 'Group1’ ‘Group2' *FROM ‘Group3’
```

Example 2: Voidage Replacement by Pattern (voidage replacement via water injection)

The 4 producers which form the 5 spot pattern for each injector are in their own reporting group called 'rep-voidageabc'. Each of these producers is given a weight of 0.25, this fraction changes on the reservoir boundaries. The 'voidageabc' control group is then put on group injection control (*GCONI) with water injection to replace 100% of voidage produced. A *WPRODGROUP keyword links the injection well group to a reporting group (with the well weights accounted for).

```
*REPORTING-GROUP 'rep-voidageabc'  11    12    21    22
                                  0.25  0.25  0.25  0.25
*GCONI 'voidageabc'     *VREP  *WATER  1.0
*WPRODGROUP  'voidageabc'  *FROM 'rep-voidageabc'
```

Maximum Re-injection Rate for Group Water Recycling (Optional) *WRECYMAXR

PURPOSE:

*WRECYMAXR is used to specify a maximum rate at which water is re-injected under a group recycling target. Water recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *WRECYMAXR have no effect unless a group water recycling injection target is currently in force.

FORMAT:

*WRECYMAXR group_name_list max_recyc_rates

DEFINITIONS:

group_name_list
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum recycling rates specified after the list. The list need not be contained on a single line.

max_recyc_rates
: A set of non-negative real values (m3/day | Stb/day). Zero is accepted as a value but is interpreted as signifying that no maximum recycling rate is set. If only a single value is supplied, that value will be applied to all groups in the list. If more than one value is supplied, the number of maximum recycling rates must equal the number of group names in the list and the first rate will be applied to the first group, etc. These maximum rates set, or set an upper limit for, the volumetric rate at which produced water is re-injected under a group water recycling target. The values may be spread over more than a single line.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *WRECYMAXR, then by default its maximum water recycling rate is set to zero, which is interpreted to mean that there is no fixed maximum imposed upon the recycling injection rate.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *WRECYMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum recycling rates enter the calculation of a water recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the maximum recycling rate values are associated with the group re-injecting the water, and not with the group producing it. The recycling volumetric injection rate is set principally by the recycling fraction entered upon the *RECYCLE *WATER line

following *GCONI, but if a maximum has been imposed with the *WRECYMAXR keyword, the re-injection target is not allowed to exceed this maximum.

Example: Group1 acquires a maximum recycling injection rate of 400. Stb/day and Group2 of 450. Stb/day.

```
*WRECYMAXR  'Group1'    'Group2'
               400.        450.
```

Here the maximum recycling rates are, but need not have been, on a separate line from the group names.

Monitored Group Constraints (Optional) *GCONM

PURPOSE:

*GCONM is used to specify monitored group production constraints. Unlike the controls specified under *GCONP and *GCONI, the quantities specified under *GCONM cannot be assigned as group targets, and no action resulting in setting the violated value as a target is possible.

FORMAT:

```
*GCONM  'group_names'
*GOR            value       (*STOP)
*WCUT                       (*SHUTALL)
*WGR                        (*SHUTMOWS)
*MAX   *MXX                 (*SHUTMOW)
*GWGR                       (*SHUTMOL)
*WWGR                       (*SHUTMOLDOWN)
*MAXGAS                     (*SHUTMOLUP)
*MAXSTW
*MINOIL         value       (*STOP|*SHUTALL)
*MININL
*MINGAS
*MINWTG
*MINBHF
*MINREC         value       (*STOP|*SHUTALL|*SHUTCYCINJ)
```

DEFINITIONS:

'group_names'

Are the groups to which the following constraints apply. The wells that are connected to each group have already been specified using the *WELL keyword. The injection and production targets are met by using one of the available apportionment methods specified by *APPOR-METHOD.

*GOR

This subkeyword identifies a maximum gas-oil ratio (m3/m3 | scf/STB) monitor for group production. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint.

*WCUT

This subkeyword identifies a maximum water-cut (fraction) monitor for group production. The water-cut is defined as (water rate)/(total liquid rate), where total liquid rate = water rate + oil rate + intermediate liquid rate. In runs where no intermediate liquid stream is defined its rate is taken to be zero. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint.

*WGR

This subkeyword identifies a maximum water-gas ratio (m3/m3 | STB/scf) monitor for group production. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint.

*MAX *MXX

This keyword activates group production stream maximum compositional monitoring for the selected component. The XX stands for the component number to be monitored. The mole percent of the component is calculated taking into account the well stream composition and production rates of all producers connected to the group. If the calculated mole percent is greater than the value specified, the associated action is taken. Up to five components can be specified to be monitored per group, i.e. up to five *MXX strings can follow *MAX. The value specified for this constraint must be expressed as a mole percentage. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint.

*WWGR

This subkeyword identifies a maximum water-wet gas ratio (m3/m3 | STB/scf) monitor for group production. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint.

*GWGR

This subkeyword identifies a maximum gas-wet gas ratio (m3/m3 | scf/scf) monitor for group production. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint.

*MAXGAS

This subkeyword identifies a maximum gas rate (m3/day | SCF/day) monitor for group production. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint. The “most offending” well or layer for this constraint is deemed to be the one with the highest GOR rather than the one with the highest gas rate.

*MAXSTW

This subkeyword identifies a maximum water rate (m3/day | SCF/day) monitor for group production. The *STOP, *SHUTALL, *SHUTMOW(S), *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are allowed for this monitored constraint. The “most offending” well or layer for this constraint is deemed to be the one with the highest water cut, rather than the one with the highest water rate.

*MINOIL

This subkeyword identifies a minimum oil rate (m3/day | Stb/day) monitor for group production. Only the *STOP and *SHUTALL actions are allowed for this monitored constraint.

*MININL

This subkeyword identifies a minimum intermediate liquid rate (m3/day | Stb/day) monitor for group production. Only the *STOP and *SHUTALL actions are allowed for this monitored constraint. More information about the intermediate liquid stream is available in the section describing the *WELSEP keyword.

*MINGAS

This subkeyword identifies a minimum gas rate (m3/day | SCF/day) monitor. Only the *STOP and *SHUTALL actions are allowed for this monitored constraint.

*MINWTG

This subkeyword identifies a minimum wet gas rate (m3/day | SCF/day) monitor. Only the *STOP and *SHUTALL actions are allowed for this monitored constraint.

*MINBHF

This subkeyword identifies a minimum bottomhole fluid rate (m3/day | bbl/day) monitor. Only the *STOP and *SHUTALL actions are allowed for this monitored constraint.

*MINREC

This keyword specifies that if the total group gas rate available for recycling falls below the specified value (m3/day | SCF/day) then perform the remedial action, which can be one of *STOP, *SHUTALL or *SHUTCYCINJ. The total recycling rate includes the make-up volumes.

value

Constraint value -- units are given under *GOR, *WCUT, *WGR, *MAXGAS, *MAXSTW, *MINOIL, *MININL, *MINGAS, *MINWTG, *MINBHF, *MXX, and *MINREC above.

*STOP

Action subkeyword indicating that if the monitored constraint is violated for a group then the simulation should be stopped. This is the default action for all constraints if no action is entered explicitly.

*SHUTALL

Action subkeyword indicating that if a monitored constraint is violated for a group, then all currently open wells in the group should be shut.

*SHUTMOWS

This specifies that if a monitor is violated then a list of prioritized most offending wells (MOWS -- the ones with the greater GOR, WCUT, WGR, WWGR, GWGR, or mole percent values, depending on the monitor) should be shut.

*SHUTCYCINJ

Action subkeyword indicating that if a monitored constraint is violated for a group, then all currently open gas cycling injectors in the group should be shut.

*SHUTMOL

Action subkeyword which specifies that if a monitor is violated then the most offending layer (MOL) in the most offending well (the one with the greatest GOR, WCUT, WGR, WWGR, GWGR, or mole percent) should be shut.

*SHUTMOLDOWN

Action subkeyword which specifies that if a monitored constraint is violated then the most offending layer (MOL) and the layers below it in the most offending well (the one with the greatest GOR, WCUT, WGR, WWGR, GWGR, or mole percent, depending on the monitor) should be shut.

*SHUTMOLUP

Action subkeyword which specifies that if a monitor is violated then the most offending layer (MOL –and the layers above it in the most offending well (the well with the greatest GOR, WCUT, WGR, WWGR, GWGR, or mole percent, depending on the monitor) should be shut.

*SHUTMOW

This specifies that if a monitor is violated then the most offending well (MOW -- the one with the greatest GOR, WCUT, WGR, WWGR, GWGR, or mole percent value, depending on the monitor) should be shut.

DEFAULTS:

Optional keyword. Default is no monitoring on groups. *STOP is the default action for all constraints.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can be given any monitored constraints.

EXPLANATION:

*GCONM is used to specify monitored constraints. Unlike the constraints which are read under *GCONP and *GCONI, the values entered under *GCONM cannot be applied as targets. Note: The *AUTO-OPEN action of the *MINOIL and *MINGAS monitors is no longer supported. Instead, automatic drilling of wells can be triggered when a group fails to meet a rate target by using a *GAPPOR *AUTODRILL *ON line for the group and specifying the well's status as *AUTODRILL. Please see the manual pages describing these keywords for more information.

Example: Automatic shutting of the most offending producer in the group when the mole percentage of component 1 in the group hydrocarbon production stream exceeds 50%, and automatic shutting of the most offending layer of the most offending well when the mole percentage of component 2 exceeds 80%.

```
*GCONM 'GROUP1'
   *MAX *M01 50. *SHUTMOW
        *M02 80. *SHUTMOL
```

Defining Group Production or Injection as Going Through a Manifold (Optional) *MANIFOLD

PURPOSE:

*MANIFOLD is used to specify that group production or injection of a stream is treated as going through a manifold. In most applications this refers to an undersea manifold, but the defining aspect of the manifold treatment is that all wells of one type (all producers, or all water injectors, etc.) in the group have the same pressure at the upper end of the tubing. A surface, or compressor, pressure may be defined for a manifold group. Manifold groups may operate under manifold and surface pressure constraints, but even when running on rate constraints the enforcement of equal manifold pressures upon the wells alters the group's performance relative to a non-manifold group.

Only a group having wells directly attached to it can have *MANIFOLD specified. Groups having other groups attached cannot have *MANIFOLD specified.

FORMAT:

*MANIFOLD (*PROD | *GASI | *WATI) group_list
(*ON | *OFF)

DEFINITIONS:

*PROD
: Indicates that group production should be treated as going through a manifold.

*GASI, *WATI
: Indicates that group gas or water injection should be treated as going through a manifold.

group_list
: Names of groups to which the current manifold specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in group_list must have been previously defined using the *GROUP keyword.

*ON
: Specifies that the manifold treatment is to be imposed.

*OFF
: Specifies that the manifold treatment is not to be imposed (or is to be removed).

DEFAULTS:

Optional keyword. Default for all groups and target types is *OFF.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group. The spacing of the subkeywords and group names is not important – they can all be on one line, or lines

may be broken arbitrarily. For example, *ON or *OFF may be on the same line as are the group names. Only one of the three possible target types may be specified in any one occurrence of the *MANIFOLD keyword. All wells operating through manifolds must have a means specified for computing tubing-head pressure, as the tubing-head pressure for these wells is interpreted as the manifold pressure. *MANIFOLD can only be specified for groups having wells attached directly to them.

Examples:

*MANIFOLD *PROD ‘Group-1’ *ON

This specifies that the production for group ‘Group-1’ is to be treated as going through a manifold. All producers attached to ‘Group-1’ must have a means specified for computing tubing head pressure through use of the *PWELLBORE keyword.

*MANIFOLD *GASI ‘Group-1’ ‘Group-2’
‘Group3’ *ON

This specifies that gas injection for ‘Group-1’, ‘Group-2’, and ‘Group-3’ should be treated as going through a manifold. All gas injectors in these groups must have a means specified for computing tubing head pressure through use of the *IWELLBORE keyword.

Pressure-Constraint Translation for Manifolds (Optional)

***PCON-MANIFOLD**

PURPOSE:

*PCON-MANIFOLD applies only to group streams treated as going through a manifold (see the pages for the *MANIFOLD keyword). When *PCON-MANIFOLD is specified, if the group is operating on a manifold or surface pressure constraint, that constraint is translated to a tubing-head pressure constraint and equal tubing-head pressure constraints are assigned to all wells attached to the manifold. Otherwise, the wells are assigned rates constraints which give the correct manifold or surface pressure at the start of the timestep. When *PCON-MANIFOLD is specified, group pressure constraints are usually honored more accurately, and higher-level rate constraints less accurately, than if it is not specified.

FORMAT:

*PCON-MANIFOLD (*PROD | *GASI | *WATI) group_list (*ON | *OFF)

DEFINITIONS:

*PROD
: Indicates that pressure translation should be applied for pressure constraints upon the production manifolds of the listed groups.

*GASI, *WATI
: Indicates that pressure translation should be applied for manifold group gas or water injection for the listed groups.

group_list
: Names of groups to which the current pressure translation specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in group_list must have been previously defined using the *GROUP keyword.

*ON
: Specifies that pressure translation is to be imposed.

*OFF
: Specifies that pressure translation is not to be imposed (or is to be removed).

DEFAULTS:

Optional keyword. Default for all groups and target types is *OFF.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group. *PCON-MANIFOLD can have an effect only if the specified group streams have been designated to be treated as going through a manifold through use of the *MANIFOLD keyword. It is not considered an error to specify *PCON-MANIFOLD when *MANIFOLD has not been specified, but pressure translation would then have no effect because a non-manifold group can never operate on a pressure constraint. The spacing of the subkeywords and group names is not

important – they can all be on one line, or lines may be broken arbitrarily. For example, the group names may be on the same line as the specification of stream type. Only one of the three possible target types may be specified in any one occurrence of the *PCON-MANIFOLD keyword. When a manifold for which *PCON-MANIFOLD is specified is running on a rate constraint rather than upon a pressure constraint, *PCON-MANIFOLD has no effect.

Examples:

*PCON-MANIFOLD *PROD ‘Group-1’ *ON

This specifies that pressure constraints for the production manifold for group ‘Group-1’ are to be translated into well tubing head pressure (manifold pressure) constraints.

*PCON-MANIFOLD *GASI ‘Group-1’ ‘Group-2’
‘Group3’ *ON

This specifies that pressure constraints for the gas injection manifolds for ‘Group-1’, ‘Group-2’, and ‘Group-3’ should be translated into well tubing-head pressure (manifold pressure) constraints.

Specification of Hydraulics Tables for Calculating Pressure Difference Between Manifold and Surface (Optional) *GPTABLE

PURPOSE:

*GPTABLE (Group Pressure Table) applies only to group targets treated as going through a manifold (see the pages for the *MANIFOLD keyword). *GPTABLE specifies the hydraulics table number to be used to compute the pressure drop between manifold and surface, for production and injection manifolds. The hydraulics table can be entered using the *PTUBE1 keyword for production manifolds or *ITUBE1 for injection manifolds.

FORMAT:

*GPTABLE (*PROD | *GASI | *WATI) group_list
table_numbers

DEFINITIONS:

*PROD
: Indicates that the pressure table numbers are being specified for production manifolds.

*GASI, *WATI
: Indicates that the pressure table numbers are being specified for gas or water injection manifolds.

group_list
: Names of groups to which the current hydraulic table number specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in group_list must have been previously defined using the *GROUP keyword.

table_numbers
: List of positive integers giving the hydraulics table numbers. The list may consist either of a single integer, in which case all of the listed groups are assigned this single hydraulics table number, or table_list must contain exactly as many entries as does the group_list, in which case the table numbers are assigned to the groups in order.

DEFAULTS:

Optional keyword. Default for all groups is no assigned hydraulics table number; in which case the surface (compressor) pressure is reported to be equal to the manifold pressure and no surface pressure constraint can be specified.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group Only one of the three possible target types may be specified in any one occurrence of the *GPTABLE keyword. The spacing of the subkeywords, group names, and table numbers is not important – they can all be on one line, or lines may be broken arbitrarily.

Examples:

```
*GPTABLE  *PROD ‘Group-1’ ‘Group-2’
1
```

This specifies that pressure drops between the production manifolds and surface for ‘Group-1’ and ‘Group-2’ will be calculated using the first hydraulics table.

```
*GPTABLE  *GASI ‘Group-1’ ‘Group-2’
‘Group-3’ 1  7  4
```

This specifies that the gas injection manifolds of ‘Group-1’, ‘Group-2’, and ‘Group-3’ should have pressure drops between compressor and manifold calculated using hydraulics tables 1, 7, and 4 respectively.

Manifold Depth for Calculating Pressure Difference Between Manifold and Surface (Optional) *GPHYDDEP

PURPOSE:

*GPHYDDEP applies only to group targets treated as going through a manifold (see the pages for the *MANIFOLD keyword). *GPHYDDEP specifies the depth used to scale the pressure drop linearly relative to the depth specified with the particular hydraulics table.

FORMAT:

*GPHYDDEP (*PROD | *GASI | *WATI) group_list
depths

DEFINITIONS:

*PROD
Indicates that the depths are being specified for production manifolds.

*GASI, *WATI
Indicates that the depths are being specified for gas or water injection manifolds.

group_list
Names of groups to which the current depth specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in group_list must have been previously defined using the *GROUP keyword.

depths
List of real numbers (m | ft) giving the manifold depths. The list may consist either of a single number, in which case all of the manifolds for the listed groups are assigned this single depth, or depths must contain exactly as many entries as does the group_list, in which case the depths are assigned to the groups in order.

DEFAULTS:

Optional keyword. Default for all manifolds is to use the depth specified with the hydraulics table used for the pressure drop calculation.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group Only one of the three possible target types may be specified in any one occurrence of the *GPHYDDEP keyword. The spacing of the subkeywords , group names, and depth values is not important – they can all be on one line, or lines may be broken arbitrarily.

Examples:

*GPHYDDEP *PROD ‘Group-1’ ‘Group-2’
3512.

This specifies that the production manifolds for ‘Group-1’ and ‘Group-2’ will be taken to be 3512 units below the surface when performing the pressure drop calculation between manifold and surface.

*GPHYDDEP *GASI ‘Group-1’ ‘Group-2’
‘Group-3’ 4564. 3256. 3987.

This specifies that the gas injection manifolds of ‘Group-1’, ‘Group-2’, and ‘Group-3’ are assigned depths of 4564, 3256, and 3987 units respectively to be used in scaling tabular calculations of the pressure drop between manifold and compressor.

Group Artificial Lift Quantity Value (Optional) *GROUPALQ

PURPOSE:

*GROUPALQ applies only to group targets treated as going through a manifold (see the pages for the *MANIFOLD keyword) and for which a manifold-to-surface hydraulics table has been specified using the *GPTABLE keyword. It introduces the value which will be used for the ALQ quantity in interpolations in a hydraulics table entered using the *PTUBE1 table format. For some choices of the ALQ variable, the variable value is arbitrary and can be specified freely in the data. In these situations the units of the ALQ quantity are not important so long as the assigned values and the tabular values are consistent.

FORMAT:

*GROUPALQ (*PROD | *GASI | *WATI) group_list
alq_values

DEFINITIONS:

*PROD
Indicates that the ALQ values are being specified for production manifolds.

*GASI, *WATI
Indicates that the ALQ values are being specified for gas or water injection manifolds. This subkeyword is accepted but hydraulics tables for injection manifold-to-surface pressure drops do not include the ALQ variable, so this specification does not have any effect in the simulation.

group_list
Names of groups to which the current ALQ specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in group_list must have been previously defined using the *GROUP keyword.

alq_values
List of real numbers giving the ALQ values. The list may consist either of a single number, in which case all of the manifolds for the listed groups are assigned this single value, or alq_values must contain exactly as many entries as does the group_list, in which case the ALQ values are assigned to the groups in order. The units are arbitrary but must correspond to the tabular units.

DEFAULTS:

Optional keyword. Default for all manifolds is an ALQ value of zero.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group Only one of the three possible target types may be specified in any one occurrence of the *GROUPALQ keyword. The spacing of the subkeywords, group names, and ALQ values is not important – they can all be on one line, or lines may be broken arbitrarily.

Examples:

```
*GROUPALQ *PROD ‘Group-1’ ‘Group-2’
1.
```

This assigns ALQ values of 1 for the production manifolds of ‘Group-1’ and ‘Group-2’.

```
*GROUPALQ  *PROD ‘Group-1’ ‘Group-2’
‘Group-3’ 1.  0.5  0.3
```

This specifies that the production manifolds of ‘Group-1’, ‘Group-2’, and ‘Group-3’ are assigned ALQ values of 1., 0.5, and 0.3 respectively to be used in tabular calculations of the pressure drop between manifold and surface.

Well Artificial Lift Quantity Value (Optional) *WELLALQ

PURPOSE:

*WELLALQ specifies an artificial lift quantity value to be used in the tabular tubing head pressure calculation for producers. It introduces the value which will be used for the ALQ quantity in interpolations in a hydraulics table entered using the *PTUBE1 table format. For some choices of the ALQ variable, the variable value is arbitrary and can be specified freely in the data. In these situations the units of the ALQ quantity are not important so long as the assigned values and the tabular values are consistent.

FORMAT:

```
*WELLALQ  well_list
 alq_values
```

DEFINITIONS:

well_list

Names or numbers of wells to which the current ALQ specification is to apply. Each well name is up to 40 characters long and enclosed in single quotes. All wells included in well_list must have been previously defined using the *WELL keyword. Both producers and injectors are valid as members of this list, but hydraulics tables for injectors do not include the ALQ variable, so specifying values for injectors will have no effect in the simulation.

alq_values

List of real numbers giving the ALQ values. The list may consist either of a single number, in which case all of the wells in well_list are assigned this single value, or alq_values must contain exactly as many entries as does the well_list, in which case the ALQ values are assigned to the wells in order. The units are arbitrary but must correspond to the tabular units.

DEFAULTS:

Optional keyword. Default for all wells is an ALQ value of zero.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group. The spacing of the well names and ALQ values is not important – they can all be on one line, or lines may be broken arbitrarily.

Examples:

```
*WELLALQ 'Well-1' 'Well-2'
1.
```

This assigns ALQ values of 1 for wells ‘Well-1’ and ‘Well-2’, in arbitrary units.

```
*WELLALQ  ‘Well-1’ ‘Well-2’
‘Well-3’ 1097.  976.5  1137.3
```

This assigns ALQ values of 1097, 976.5, and 1137.3, in arbitrary units, to ‘Well-1’, ‘Well-2’, and ‘Well-3’ respectively.

Priority List for Automatic Drilling of Wells (Optional) *DRILLQ *DRILLT *DRILLMXW

PURPOSE:

*DRILLQ allows the selection of algorithm for automatic opening of wells or the specification of the order (list by the user) in which wells must be drilled automatically.

*DRILLT allows the specification of a time delay between successive well drilling operations.

*DRILLMXW allows the specification of the maximum number of wells that can be drilled at any one time.

FORMAT:

```
*DRILLQ        *IPP | *IIP | *IPPQT |
               *IPPDRNINP|
               *IPPDRNINPQT
               Or
*DRILLQ         well_list
               Or
*DRILLQ        well list  *IPP | *IIP | *IPPQT
               *IPPDRNINP|
               *IPPDRNINPQT

*DRILLT        tinv_p  tinv_j
*DRILLMXW      ndrillmxwp ndrillmxwi
```

DEFINITIONS:

*IPP

Specifies that the instantaneous production potential of producing wells with status *AUTODRILL will be used to determine the order in which these wells will be drilled, with wells of higher potential opened first. The instantaneous production potential is taken to be the well's production rate at a bottomhole (BHP) pressure of 1 atm.

*IPPQT

Specifies that the instantaneous production potential of producing wells with status *AUTODRILL will be used to determine the order in which these wells will be drilled, with wells of higher potential opened first. *IPPQT can be used for producers only. The instantaneous production potential is taken to be the well's production rate at a bottomhole (BHP) pressure of 1 atm. Wells that cannot produce above a minimum threshold target rate will not be opened. This threshold is currently set at 1.0d-04 m3/d and can be changed using the keyword *DRILLQTREFP.

*IIP

Specifies that the instantaneous injection potential of injection wells with status *AUTODRILL will be used to determine the order in which these

wells will be drilled, with wells of higher potential being opened first. *IIP must be used for injectors. The instantaneous injection potential is taken to be the well's injection rate at a bottomhole pressure of 3500 atm.

*IPPDRNINP

Specifies that the product of instantaneous production potential and in place surface or reservoir volume of target phase will be used to determine the order in which producing wells with status *AUTODRILL will be drilled, with wells of higher instantaneous production potential times in place volume opened first. *IPPDRNINP can only be used for producers. The instantaneous production potential is taken to be the well's production rate at a BHP of 1 atm. The in place volume (surface or reservoir) of the target phase (oil, gas, bhf etc.) will be calculated by first determining which grid blocks either wholly or partially lie within the assigned drainage radius for each well. This allows the total moles and composition of each of the reservoir phases, oil, gas and water to be determined for each well. Surface volumes are determined by flashing to the surface each of these reservoir phases. A default drainage radius of 745 ft corresponding to a 40 acre spacing is used unless the user enters a value using the keyword *DRNRADW. This option is available for GEM only.

*IPPDRNINPQT

Specifies that the product of instantaneous production potential and in place surface or reservoir volume of target phase will be used to determine the order in which producing wells with status *AUTODRILL will be drilled, with wells of higher instantaneous production potential times in place volume opened first. *IPPDRNINPQT can only be used for producers. The instantaneous production potential is taken to be the well's production rate at a BHP of 1 atm. The in place volume (surface or reservoir) of the target phase (oil, gas, bhf etc.) will be calculated by first determining which grid blocks either wholly or partially lie within the assigned drainage radius for each well. This allows the total moles and composition of each of the reservoir phases, oil, gas and water to be determined for each well. Surface volumes are determined by flashing to the surface each of these reservoir phases. A default drainage radius of 745 ft corresponding to a 40 acre spacing is used unless the user enters a value using the keyword *DRNRADW. This option is available for GEM only. Wells that cannot produce above a minimum threshold target rate will not be opened. This threshold is currently set at 1.0d-04 m3/d and can be changed using the keyword *DRILLQTREFP.

well_list

A list of 'well_names' or well_numbers that give the priority in which wells will be drilled. The first well on the list will be drilled first and so on. Producers and injectors may be specified in the same list. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. Note: The well list may be as an option be followed by one of the keywords *IPP, *IIP, *IPPQT, *IPPDRNINP or *IPPDRNINPQT. Therefore the user may choose the order in

which a subset of all wells with autodrill status is drilled and allow the internal algorithm of choice to rank the remaining wells with autodrill status. The wells on the user specified list have a higher priority and will be drilled according to the order that they are listed in the queue. If a keyword specifying a ranking algorithm does not follow the well list, then only wells on the list can be drilled.

tinv_p tinv_j

Two non-negative real numbers (one for producers, and one for injectors) specifying the minimum time intervals in days between the wells being drilled due to the *AUTODRILL actions.

ndrillmxwp

A positive integer specifying the maximum number of production wells that can be drilled (or opened) at any one time. If a corresponding value of tinv_p is not entered with the keyword DRILLT, then all wells may be opened if necessary at every time step. If a tinv_p value is specified, then a maximum of ndrillmxwp wells can be opened every tinv_p days.

ndrillmxwi

A positive integer specifying the maximum number of injection wells that can be drilled (or opened) at any one time. If a corresponding value of tinv_i is not entered with the keyword DRILLT, then all wells may be opened if necessary at every time step. If a tinv_i value is specified, then a maximum of ndrillmxwi wells can be opened every tinv_i days.

DEFAULTS:

Optional keywords. Default is to use the instantaneous injection / production potential to determine drilling priority. There is no time delay for well drilling if *DRILLT is absent. The default value of ndrillmxwp is 1. The default value of ndrillmxwi is 1.

CONDITIONS:

*DRILLQ and *DRILLT and DRILLMXW must be located in the WELL AND RECURRENT DATA keyword group, and the listed wells must have been defined with the keyword *WELL. Wells that are listed but do not have autodrillable status nor subordinate to the targeted group will be ignored. DRILLQ, DRILLT and DRILLMXW must appear after the groups are defined. Groups are defined with the keyword GROUP. Therefore the keyword GROUP is a prerequisite for DRILLQ, DRILLT and DRILLMXW. If a positive value for tinv_p is not entered with keyword *DRILLT, then the value on ndrillmxw entered with DRILLMXW is not used and all eligible wells can be opened at any time. DRILLMXW does not apply to injection wells.

EXPLANATION:

DRILLQ is used to specify the priority in which wells that have *AUTODRILL as their status will be drilled. One may either use the instantaneous injection (IIP) / production potentials (IPP) / production potential multiplied by in place volume (IPPDRNINP) / production potentials (IPPQT) with minimum rate threshold / production potential multiplied by in place volume (IPPDRNINPQT) with minimum rate threshold or supply a list to give a drilling priority order.

The opening of *AUTODRILL wells is triggered during the apportionment of group rate targets if a group fails to be able to meet its rate target and if the group has had the *AUTODRILL feature turned on through the pair of lines

```
*GAPPOR 'group_name'
   *AUTODRILL *ON
```

Please consult the documentation for the *GAPPOR keyword for more information.

If a well_list is given then wells will be opened in the order in which they appear on the list with the minimum time delay specified by *DRILLT until the list is exhausted. A second well_list following *DRILLQ will reset the whole sequence of automatic drilling.

Example:

```
*GCONP 'Group1'
  *MAX  *STO  1000.0
  *DRILLQ  'well-4'  'well-5'  'well-6'
  *DRILLT  10.0  0.0  ** No delay for injectors
```

Drainage Radius of Wells for IPPDRINP based Autodrill Algorithm (Optional) *DRNRADW

PURPOSE:

*DRNRADW allows the user to enter a drainage radius for a production well or group of production wells. The drainage radius defines a reservoir volume with the well at the centre. A number of grid blocks will be either fully within the drainage radius or in the case that the drainage radius intersects a grid block, partially within the drainage radius. The drainage radius therefore allows the determination of a grid block list and subsequently in place amount of each phase, oil, gas and water at reservoir and surface conditions.

FORMAT:

```
*DRNRADW  (well_names) or ( well_numbers)
  values
```

DEFINITIONS:

well_names

Any number of well names (in quotes) to specify the wells to which this keyword applies. The well names must be on the same line as *DRNRADW and must end with the end of the line. If more wells than can be listed on one line must have DRNRADW specified, a new *DRNRADW line must be entered.

Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

well_numbers

Any number of integers, or ranges of integers specifying the well numbers to which this keyword applies. The list must be on the same line as *DRNRADW and must end with the end of the line. If more wells than can be listed on one line must have DRNRADW specified, a new *DRNRADW line must be entered.

values

Either a single real number or multiple real numbers designating the drainage radius. In the latter instance, the number of entries must equal exactly the same number of entries as the number of wells. If a single real number is entered in the list, it is applied to all wells in list. The values must appear on a new line.

DEFAULTS:

Optional keyword. If *DRNRADW does not appear in the data set, the value of 745 ft corresponding to a 40 acre spacing is used for all production wells.

CONDITIONS:

If it appears, this keyword must be located in the WELL AND RECURRENT DATA keyword group. It must appear AFTER (but not necessarily immediately after) the first *DATE line. All wells appearing in the well list must already have been defined with *WELL lines. The wells in the list may only be producers.

EXPLANATION:

The drainage radius allows in place amounts to be assigned to each well exclusively for the purpose of ranking wells in terms of auto drill priority. In contrast to ranking based on IPP (instantaneous production potential) an alternate criteria is too rank wells based on the product of maximum rate of the target phase multiplied by the in place amount of the target phase. This has the potential of selecting wells with both high productivity (high effective permeability) and a large volume of target phase (oil, gas) available to them. This is especially useful for coal bed methane (CBM) reservoirs where typically initially the reservoir is watered out. For more detail refer to the DRILLQ keyword.

Example:

```
*DRNRADW  'WELL1' 'WELL2'
 25.0 50.0
```

Implies a drainage radius of 25 ft | m will be assigned to WELL1 and a drainage radius of 50 ft | m for WELL2 depending on the unit system specified.

Alternate Phase for Ranking Wells on Auto Drill (Optional)

***DRILLALTP**

PURPOSE:

*DRILLALTP is used to specify an alternate phase for ranking (prioritizing) production wells which are designated as autodrill wells and either *IPP, *IPPQT, * IPPDRNINP or *IPPDRNINPQT method is selected as a means of ranking wells and assigning priority.

FORMAT:

*DRILLALTP 'group_names'
(*OIL|*STG|*STW|*INI|*STL|*WTG|*BHF)

DEFINITIONS:

group_names
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will use the alternate phase specified following the group names to assign priority to wells designated with the AUTODRILL status if necessary.

*STO | *STW | *STG | *INI | *STL | *WTG | *BHF
: One of the above options to select the alternate phase for priority ranking of wells with the *AUTODRILL status.

: *STO designates stock tank oil, *STW designates stock tank water, *STG designates stock tank gas, *INI designates intermediate surface liquid stream, *STL designates stock tank liquid (sum of oil, intermediate liquid and water), *WTG designates wet gas stream and *BHF designates bottom hole fluid (sum of oil, gas and water reservoir phases). If *INI is selected, the user has to have defined surface separation strategy which results in the creation of an intermediate liquid stream.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *DRILLALTP, then by default production wells with the AUTODRILL status belonging to that group will not be ranked based on maximum production potential of alternate phase (*IPP method) or product of maximum production potential of the alternate phase times the in place amount of the target phase (*IPPDRNINP method).

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can appear in the group list following *DRILLALTP.

EXPLANATION:

The primary list (list 1) consists of wells that have a maximum rate of the target (reference phase) phase greater than a minimum threshold value. This value is currently equal to 1.0E-4 m3/d. These wells are ranked from the largest rate to smallest rate. If an alternate phase is

entered then a secondary (list 2) consists of wells that cannot produce the target phase rate greater than the prescribed minimum rate but can produce the maximum rate of the alternate phase greater than the minimum value. For example for coal bed methane reservoirs the target phase is likely to be STG or BHF and alternate phase STW. The second list consists of wells ranked based on the maximum rate of the alternate phase. The premise is that it is better to open a well that can produce a large rate of the alternate phase than none of the target or primary phase. The wells in primary or list 1 must be opened first before wells on list 2 (alternate phase can be opened. If the *IPPDRNINP method is chosen with the *DRILLQ keyword, then wells with a positive value for the drainage radius (default is 745 ft) will gave a higher priority than those that are assigned a negative value for drainage radius by the user.

Table 1 below summarizes the potential lists and their priorities depending on which method is selected. Wells on list 1 have a higher priority that wells on list 2. Wells on list 2 have a higher priority than wells on list 3 etc. All wells in list 1 must be opened before any wells on list 2 can be opened etc.

All rates refer to instantaneous production potential, that is maximum rate at a BHP of 1 atm.

Description of variables in Table 1:

q_{tp}	-	target phase rate, that is rate at BHP of 1 atm (m3/d, ft3/d)
$q_{min,tp}$	-	minimum (threshold) rate target phase, set equal to 1.0E-04 m3/d (can be read in, DRILLQTREFP)
q_{ap}	-	alternative phase rate, that is rate at BHP of 1 atm (m3/d, ft^3/d)
$q_{min,ap}$	-	minimum (threshold) rate alternative phase, set equal to 1.0E-04 m3/d (can be read in DRILLQTALTP)
r_d	-	drainage radius (a value is assigned to each production well)
V_{inp}	-	In place volume of target phase for each well. Calculated on the basis of a drainage radius (m^3/scf)

Table 1: Ranking criteria for IPP based AUTODRILL options

	Ranking criterion				
	List #1 (Priority #1)	**List #2 (Priority #2) – Alternate phase**	**List #3 (Priority #3)**	**List #4 (Priority #4) Alternate phase**	**List #5 (Priority #5)**
*IPP (without alternate phase)			Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$		Target maximum phase rate
*IPPQT (without alternate phase)			Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$		Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$
*IPP (with alternate phase)			Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$	Alternate maximum phase rate, as long as $q_{ap} > q_{min,ap}$	Target maximum phase rate
*IPPQT (with alternate phase)			Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$	Alternate maximum phase rate, as long as $q_{ap} > q_{min,ap}$	Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$
*IPPDRNINP (without alternate phase)	Wells with $r_d > 0$ Target maximum phase rate multiplied by V_{inp}, as long as $q_{tp} > q_{min,tp}$		Wells with $r_d < 0$ Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$		Target maximum phase rate
*IPPDRNINPQT (without alternate phase)	Wells with $r_d > 0$ Target maximum phase rate multiplied by V_{inp}, as long as $q_{tp} > q_{min,tp}$		Wells with $r_d < 0$ Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$		Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$
*IPPDRNINP (with alternate phase)	Wells with $r_d > 0$ Target maximum phase rate multiplied by V_{inp}, as long as $q_{tp} > q_{min,tp}$	Alternate maximum phase rate multiplied by V_{inp}, as long as $q_{ap} > q_{min,ap}$	Wells with $r_d < 0$ Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$	Alternate maximum phase rate, as long as $q_{ap} > q_{min,ap}$	Target maximum phase rate
*IPPDRNINPQT (with alternate phase)	Wells with $r_d > 0$ Target maximum phase rate multiplied by V_{inp}, as long as $q_{tp} > q_{min,tp}$	Alternate maximum phase rate multiplied by V_{inp}, as long as $q_{ap} > q_{min,ap}$	Wells with $r_d < 0$ Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$	Alternate maximum phase rate, as long as $q_{ap} > q_{min,ap}$	Target maximum phase rate, as long as $q_{tp} > q_{min,tp}$

Rate Cutoff Thresholds for Auto Drill (Optional) *DRILLQTREFP *DRILLQTALTP

PURPOSE:

*DRILLQTREF and DRILLQTALTP indicate input of rate threshold cutoff's for automatic drilling of wells (wells with status AUTODRILL). If the volumetric flow rate of the phase is below the cutoff value then that well will not be drilled (opened).

FORMAT:

*DRILLQTREFP 'group_names' qtrefpval
*DRILLQTALTP 'group_names' qtaltpval

DEFINITIONS:

group_names
: A list of one or more valid group names, enclosed in single quotes. The groups named in this list will use the threshold rate specified to determine whether a well(s) designated to be opened by the autodrill algorithm will actually be opened.

qtrefpval
: A value for the reference (target) phase rate cut off. In SI units enter a value in units of m3/d and in field units enter a value in units of scf/d.

qtaltpval
: A value for the alternate phase rate cut off. In SI units enter a value in units of m3/d and in field units enter a value in units of scf/d.

DEFAULTS:

Optional keywords. The defaults are set based on the AUTODRILL option specified by the DRILLQ option

Option	Defaults (target and alternate phases)
IPP	1.0E-20 m3/d
IPPQT	1.0E-04 m3/d
IPPDRNINP	1.0E-20 m3/d
IPPDRNINPQT	1.0E-04 m3/d

CONDITIONS:

These keywords must be in the well and recurrent data section of the of the data file. They must be appear after the group hierarchy is specified since thresholds are assigned on the basis of user defined groups. The thresholds read in here will only be applied if the AUTODRILL feature is turned on with the one specific option IPP|IPPQT|IPPDRNINP or IPPDRNINPQT. For more information refer to the DRILLQ keyword.

EXPLANATION:

By default the autodrill algorithm will drill (open) a well even if it can produce at a very low rate as long as the assigned group target is not being met. The user can however choose to not open wells unless the maximum rate calculated based on the BHP of 1 atm exceeds a certain minimum threshold or cutoff. The values of these cutoff's can be specified with keywords DRILLREFP for the reference (target) phase and by DRILLALTP for the alternate phase.

Example:

```
*DRILLQTREFP 'Group-1' 4.0E-4
```

Assigns rate cutoff of 4.0E-04 m3/d in case of SI units and 4.0E-04 scf/d in case of field units

Group Apportionment Options (Optional) *GAPPOR

PURPOSE:

*GAPPOR introduces subkeywords which control how the apportionment of group target among contributing wells or subgroups is to be done. Currently *AUTODRILL is the only subkeyword supported.

FORMAT:

```
*GAPPOR        group_list
               *AUTODRILL (*ON | *OFF | stream identifier)
```

DEFINITIONS:

group_list

A list of group names to which the following constraints apply. The wells that are connected to each group have already been specified using the *WELL keyword. The injection and production targets are apportioned using one of the available apportionment methods specified by *APPOR-METHOD.

*AUTODRILL (*ON | *OFF | stream identifier)

The *AUTODRILL apportionment option is turned on or off. *ON specifies that if one of the listed groups is attempting to meet a rate target, and the group apportionment routine predicts that the contributing wells have insufficient injection or production potential to meet this target, any potentially contributing wells which are shut but have *AUTODRILL status may be opened automatically to meet the target. *OFF disables the option.

The stream identifier takes the form of *STO, *STG, *STW, *STI, *STL, *WTG and *BHF. They are the available group target streams as defined by group control keyword *GCONP under *GTARGET (refer to the explanation therein for details). The stream identifier has the same effect as *ON if the group is currently targeted to that stream; and has the same effect as *OFF if the group is running on a different targeted stream.

DEFAULTS:

Optional keyword. Default is no *AUTODRILL action in the apportionment. If *AUTODRILL appears with no following *ON or *OFF directive, *ON is assumed.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined before it can be assigned options under *GAPPOR.

EXPLANATION:

*GAPPOR is used to specify options used in the course of apportioning a group's rate target among the contributing wells.

The group control may direct a group to switch target (value and stream) if there is a maximum constraint (*GCONP *MAX) being violated. *AUTODRILL followed by the stream identifier ensures that auto-drillable wells can be opened only when the group is apportioned for the desired stream.

Example

The following line directs that if group GROUP1 is unable to meet its current rate target (if there is one), any wells connected directly or indirectly to GROUP1 which have AUTODRILL status should be opened in order until either no more AUTODRILL wells are left or until the target can be met. The order in which the wells are to be opened is determined by the *DRILLQ keyword; the default is to open the wells in decreasing order of instantaneous injection/production potential.

```
*GAPPOR 'GROUP1' *AUTODRILL *ON
```

Apportionment Method for Meeting Group Targets (Optional)

***APPOR-METHOD**

PURPOSE:

*APPOR-METHOD specifies the apportionment method to distribute the production or injection rates among contributing wells or groups for meeting the group target(s) defined by *GCONP or *GCONI.

FORMAT:

```
*APPOR-METHOD  (*PROD | *GASI | *WATI)  group_list
               (*IP | *GUIDE | *INGUIDE | *PRIOR)
```

DEFINITIONS:

*PROD

Indicates that group production targets should be apportioned with the specified method.

*GASI, *WATI

Indicates that group gas or water injection targets should be apportioned with the specified method.

group_list

A list of group names to which the apportionment method is applied. Wildcard characters may be used in the group names.

*IP

This sub-keyword specifies the use of instantaneous potential (IP), which is computed internally by the simulator, to determine the apportionment of production or injection targets among wells. This is default. Each contributing well is assigned a rate in proportion to its maximum rate for the surface stream for which the target is referenced. This has the effect that either the target can be met with all contributors being assigned rates in the same fraction of their maximum rate, or the target cannot be met and all contributors are assigned exactly their maximum rates.

*GUIDE

This sub-keyword specifies the use of user-supplied guide rates to determine the apportionment of production or injection targets among contributing wells or groups. Each contributing well or group receives a fraction of the target in proportion to its guide rate. If the assignment violates its maximum rate under the most restrictive constraint, such a well or group will be left out the apportionment with rate being cut back to the maximum rate. The reminder of the target is still distributed proportional to the guide rates among the remaining wells or groups whose assignments do not exceed the maximum rates. The guide rates of a proper surface reference stream are specified via keyword *GUIDEP for production or *GUIDEI for injection (refer to the explanation therein for details).

*INGUIDE

This sub-keyword specifies the use of internal guide rates to determine the apportionment of production or injection targets among contributing wells. The idea is very similar to the above *GUIDE, except that the guide rates are now generated internally by the simulator according to the user-supplied priority formula defined by keyword *PRIOR-FORM (refer to the explanation therein for details). The default is to use the instantaneous productivity or injectivity potential under a bottom-hole pressure constraint: 1 atm for production and 3500 atm for injection, if the priority formula is not present. Unlike the user-supplied guide rate (*GUIDE), the internally-generated guide rates are time dependent.

*PRIOR

This sub-keyword specifies the use of priority ranking to determine the apportionment of group targets among contributing wells. Wells are ranked and opened in order of decreasing priority until reaching a swing well. The user-supplied priority formulae and numerical controls are specified by keyword *PRIOR-FORM (refer to the explanation therein for details). The default is to use the instantaneous productivity or injectivity potential under a bottom-hole pressure constraint: 1 atm for production and 3500 atm for injection, if the priority formula is not present.

DEFAULTS:

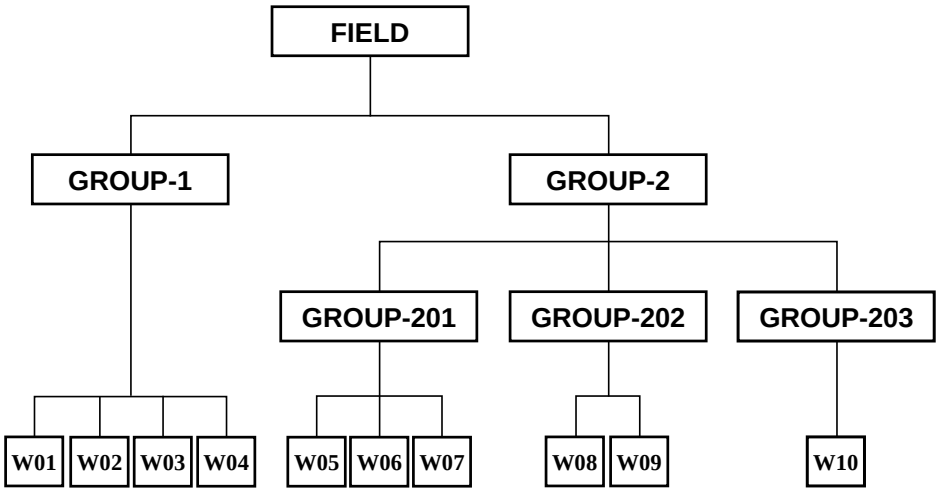
Optional keyword. Default is to use the internal guide rates (*INGUIDE) to determine rate distributions among wells to meet the group target rate.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It should appear after the group hierarchy is available. Defining an apportionment method for non-targeted groups or non-targeted injection streams does NOT have an effect. Furthermore, it does not have any effect to a production group or an injection group (with a particular stream) if the group is under a manifold control (see *MANIFOLD), since manifold itself is a special apportionment method.

EXPLANATION:

*APPOR-METHOD defines how to distribute a group target within its pool – the collection of all the contributing wells or groups. The pool for a targeted group consists of all its child groups and the wells directly attached. Groups and wells can only contribute to one particular target. They will be excluded from the pool created for a higher-level target if they are currently contributing to a subordinate target. The idea can be best demonstrated with the following group hierarchy sketch: if only the FIELD target is defined, all the groups and wells listed will contribute to it; however, if GROUP-2 is targeted additionally, then its child groups (GROUP-201, GROUP-202 and GROUP-203) and their wells (W05–W10) will contribute directly to the GROUP-2 target instead of the top-level FIELD target. The pool created originally for FIELD now consists of only GROUP-1 and its wells (W01–W04).



For a given group target, a proper apportionment method needs to be specified. Different apportionment method can in principal be applied\mixed to different group targets in a single run. CMG's multi-level well management module currently supports the following methods:

- Instantaneous potentials (*IP, default)
- Guide rates (*GUIDE)
- Internally generated guide rates (*INGUIDE)
- Priority ranking (*PRIOR)

For the above example, the FIELD target can be apportioned, say, with the instantaneous potential method (*IP) while the priority ranking method (*PRIOR) may be used for the targeted GROUP-2 in a single run.

Example

```
*GCONP  'FIELD'    *GTARGET  *STO  1000.
*GCONP  'GROUP-2'  *GTARGET  *STO  600.
*APPOR-METHOD  *PROD  'FIELD'    *IP
*APPOR-METHOD  *PROD  'GROUP-2'  *PRIOR
*PRIOR-FORM   *PROD 'Group-2'
   *PRIOR-NUMER   0.  1.  0.  0.  0.  0.  0.  0.
   *PRIOR-DENOM   0.  0.  1.  0.  0.  0.  0.  0.
```

In case that the GROUP-2 target is to be apportioned using the guide rates method (*GUIDE), the group rates for contributing groups GROUP-201, GROUP-202 and GROUP-203 and/or well rates for contributing wells W05 – W10 shall be supplied by *GUIDEP. Guide rates supplied for non-contributing groups\wells are ignored. On the other hand, instantaneous potential (IP) is defaulted as the guide rate if it has not been supplied by the user for a contributing group or well that is under the guide rates control.

```
*APPOR-METHOD  *PROD  'GROUP-2'   *GUIDE
*GUIDEP  *STO  'GROUP-201'  'GROUP-202'  'GROUP-203'
                 300.         200.         100.
**and/or
*GUIDEP  *STO  'W05'  *W06'  'W07'  'W08'  'W09'  'W10'
                100.
```

Backward Compatibility Notes:

The sub-keywords *IPP under *GCONP and *IIP under *GCONI in the previous versions, which set the *IP apportionment flag globally for all the group production or injection targets, now become obsolete. They will be ignored with reminding message if *APPOR-METHOD is encountered ahead in the datasets. However, the backward compatibility is still maintained and any old dataset containing *IPP or *IIP will run exactly the same since it shall not have the new keyword *APPOR-METHOD.

```
*GCONP *IPP = *APPOR-METHOD *PROD 'all_groups' *IP
*GCONI *IIP = *APPOR-METHOD *GASI 'all_groups' *IP
              *APPOR-METHOD *WATI 'all_groups' *IP
```

The group and/or well guide rates used in the guide-rate apportionment are supplied following keywords *GUIDEP and *GUIDEI as in the previous versions. However, they do not set the apportionment flag anymore. The guide-rate apportionment has to be invoked by entering the keyword *GUIDE under *APPOR-METHOD for either production or injection targets. The backward compatibility is still maintained by interpreting *GUIDEP and *GUIDEI as the 'old' ones if *APPOR-METHOD does not appear ahead in a dataset. Therefore, old datasets containing *GUIDEP and *GUIDEI will run exactly the same.

```
*GUIDEP_old 'ref stream' guide_rates
            = *APPOR-METHOD *PROD 'all_groups' *GUIDE
              *GUIDEP 'ref stream' guide_rates
*GUIDEI_old *STG guide_rates
            = *APPOR-METHOD *GASI 'all_groups' *GUIDE
              *GUIDEI *STG guide_rates
*GUIDEI_old *STW guide_rates
            = *APPOR-METHOD *WATI 'all_groups' *GUIDE
              *GUIDEI *STW guide_rates
```

Priority Formulae for Apportionment (Conditional)

***PRIOR-FORM**

PURPOSE:

*PRIOR-FORM defines the priority formulae and numerical control parameters for the priority ranking apportionment method or the internal guide rate method set by *APPOR-METHOD *PRIOR / *INGUIDE to meet group targets. It is aimed to produce a desired stream while minimizing the production of other/nuisance streams. It can also be used to meet injection target by which only the most "injectable" wells operate.

FORMAT:

```
*PRIOR-FORM     (*PROD | *GASI | *WATI)  group_list
                (*PRIOR-RATE     (*MRC | *BHP (bhp_val)) | *WBHP)
                (*PRIOR-CTRL     freq  tcr_min  trc_max)
                (*PRIOR-NUMER    A0 A1 … Anph)
                (*PRIOR-DENOM    B0 B1 … Bnph)
```

DEFINITIONS:

*PROD

Indicates that group production targets should be apportioned with the specified method.

*GASI, *WATI

Indicates that group gas or water injection targets should be apportioned with the specified method.

group_list

A list of group names to which the priority formula is applied. Wildcard characters may be used in the group names.

*PRIOR-RATE

This sub-keyword defines what type of rate is to be used in calculating the well priorities. *MRC indicates that the well priorities are to be calculated using the stream rates under the most restrictive condition. *BHP indicates that the well priorities are to be calculated using the production or injection potentials at a specified bottomhole pressure, *bhp_val*. If *bhp_value* does not appear after *BHP, the default value will be applied: 1 atmosphere for production targets and 3500 atmosphere for injection targets. Note that wells may become unproducible or uninjectable under a given *bhp_val*, and thus may be temporarily shutin by the group control. *WBHP indicates that the well priorities are to be calculated using the production or injection potentials at the values of well's own BHP constraint which may not necessarily be the most restrictive.

*PRIOR-CTRL

This sub-keyword introduces the numerical parameters that control the priority ranking apportionment. ‘freq’ is the minimum time interval in days between well priority calculations. ‘tcr_min’ is the lower limiting value of the ratio between the pool target and its capacity, above which the priority ranking apportionment will be applied. ‘tcr_max’ is the upper limiting value of the ratio between the pool target and its capacity, below which the priority ranking apportionment will be applied. *PRIOR-CTRL is not necessary to and thus has no effect on the internal guide rate apportionment.

*PRIOR-NUMER

This sub-keyword introduces the priority weighting coefficients for the numerator A_i (i = 0, nph). ‘nph’ is the number of surface production or injection streams allowed for group targets.

*PRIOR-DENOM

This sub-keyword introduces the priority weighting coefficients for the denominator B_i (i = 0, nph).

DEFAULTS:

Conditional keyword. If any of the sub-keywords (*PRIOR-CTRL, *PRIOR-RATE, *PRIOR-NUMER, *PRIOR-DENOM) is absent after *PRIOR-FORM, its default value(s) will be applied:

```
*PRIOR-RATE    *MRC
*PRIOR-CTRL    0.0  0.0  1.0
*PRIOR-NUMER   1.0  0.0  0.0 ...  0.0
*PRIOR-DENOM   1.0  0.0  0.0 ...  0.0
```

For subkeywords *PRIOR-NUMER and *PRIOR-DENOM, input of fewer number of the weighting coefficients (< 1+nph) is allowed. The truncated values remain as the default. If *PRIOR-FORM is not present, the instantaneous potentials of the proper phase at wells’ bottom-hole pressure of 1 atm for production or 3500 atm for injection are used.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group after the group hierarchy is available. Defining priority formulae for non-targeted groups or non-targeted injection streams does NOT have an effect.

EXPLANATION:

*PRIOR-FORM is used for either the priority ranking apportionment or the internal guide rate apportionment. In the priority ranking apportionment, wells contributing to a target are ranked in order of decreasing priority. When the pool target-capacity ratio falls between the threshold values [*tcr_min*, *tcr_max*], wells with highest priorities are assigned to operate at *tcr_max* fraction (upper limit) of their maximum rates according to the most restrictive constraints. If they have sufficient capacity to meet the group target, one 'swing' well will be identified such that the target will be exceeded if this well were also assigned *tcr_max* fraction of its maximum rate. Wells with lower priorities than that of the swing well are assigned *tcr_min* fraction (lower limit) of their max rates. The swing well is then assigned a makeup rate exactly equal to the deficit. Wells with lower priority, including the swing well,

are flagged so that they are not shut-in by well management in case of violating the minimum rate constraints.

The priority index for an individual well *iw* contributing to a targeted group *ig* is defined as

$$\text{Priority(iw)} = \frac{A_0(\text{ig}) + \sum_{i=1}^{\text{nph}} A_i(\text{ig})Q_i(\text{iw})}{B_0(\text{ig}) + \sum_{i=1}^{\text{nph}} B_i(\text{ig})Q_i(\text{iw})}$$

where A_i and B_i (i = 0, nph) are the weighting coefficients for the numerator and denominator, respectively, introduced by sub-keywords *PRIOR-NUMER and *PRIOR-DENOM. It is assumed that all wells contributing to a common group target should have the same priority definition. However, different targeted groups or different targeted injection streams may have different specifications of well priorities. Coefficients A_0 and B_0 are constants whilst the rest are the stream weighting coefficients to be multiplied correspondingly by Q_i, the stream rate under the most restrictive constraints (*MRC) or the production/injection potential at the given bottomhole pressure (*BHP or *WBHP) distinguished by sub-keyword *PRIOR-RATE. The allowable surface reference streams for the simulator are outlined in the table below. They are defined in detail elsewhere on the manual page for keywords *GUIDEP and *GUIDEI. All the weighting coefficients are non-negative real numbers and at least one A_i and one B_i must be non-zero.

GEM (Production: nph = 7)

i	CONST	STO	STG	STW	STI	STL	WTG	BHF
NUMER	A0	A1	A2	A3	A4	A5	A6	A7
DENOM	B0	B1	B2	B3	B4	B5	B5	B7

GEM (Injection: nph = 2)

i	CONST	STG	STW
NUMER	A0	A1	A2
DENOM	B0	B1	B2

The first value read by sub-keyword *PRIOR-CTRL, *freq*, indicates the elapsed minimum time in days between priority calculations. The default is to update the well priority at every timestep (*freq* = 0.0). It is designed to reduce in some cases the frequent alternating of the swing well among wells that may have very comparable priority values.

The last two values after sub-keyword *PRIOR-CTRL, *tcr_min* and *tcr_max*, are the threshold values for the pool target-capacity ratio (between 0-1), within which the ranking apportionment will be applied. The instantaneous potential (IP) apportionment will otherwise be adopted if the said ratio is out of the range [*tcr_min*, *tcr_max*]. Default is [0, 1], meaning that the priority ranking apportionment will always be in place regardless of the pool target-capacity ratio.

The upper limit *tcr_max* is designed to avoid the constant violation (and therefore switching) of the most restrictive constraints that might happen to some wells if they were forced to produce or inject at their maximum rates (*tcr_max* = 1). Another consideration is that if the pool has a very low capacity to meet a certain target (i.e. high pool target-capacity ratio), all contributing wells are required to operate at their maxima. The ranking apportionment may take the least effect because there is no swing well to choose.

The lower limit *tcr_min*, on the other hand, provides a flexible control over the wells with lower priorities than the swing well. These wells can operate at the *tcr_min* fraction of their maximum rates allowable instead of at the zero rate.

Example

Supply the priority formula for targeted groups ‘Group-1’ and Group-2’. As invoked by *APPOR-METHOD for the priority ranking method, wells will be opened in decreasing GOR when the pool target-capacity ratios fall between [0.0, 0.8]. Well priorities are evaluated using the *MRC rates (default) and recalculated at least every 30 days.

```
*APPOR-METHOD *PROD 'Group-1' 'Group-2' *PRIOR
*PRIOR-FORM   'Group-1' 'Group-2'
              **freq      tcr_min     tcr_max
   *PRIOR-CTRL   30.       0.0         0.8
              **CONST  STO  STG  STW  STI  STL  WTG  BHF
   *PRIOR-NUMER   0.    1.   0.   0.   0.   0.   0.   0.
   *PRIOR-DENOM   0.    0.   1.   0.   0.   0.   0.   0.
```

Wells can also be opened in order of decreasing WCUT :

```
*PRIOR-NUMER   0.    1.   0.   1.   0.   0.   0.   0.
*PRIOR-DENOM   0.    0.   0.   1.   0.   0.   0.   0.
```

or in order to minimize gas and water production as a whole. Proper weight for the gas is needed since rate values of gas production are generally higher by magnitudes than those of water:

```
*PRIOR-NUMER    0.    1.   0.   1.   0.   0.   0.   0.
*PRIOR-DENOM    0.    0.   0.01 1.   0.   0.   0.   0.
```

Guide Rates for Groups or Wells (Optional) *GUIDEP, *GUIDEI

PURPOSE:

*GUIDEP or *GUIDEI specifies the guide rates as needed if the guide rate apportionment method is set by *APPOR-METHOD *GUIDE to distribute the production or injection rates to groups or wells so as to meet the production or injection target.

FORMAT:

*GUIDEP	*STO	(group_list)	guide_rates
*GUIDEI	*STG	(well_list)	
	*STW		
	*STL		
	*STI		
	*WTG		
	*BHF		

DEFINITIONS:

*STO

This subkeyword indicates that the oil stream is the reference stream to which the guide rates apply; i.e., that the guide rate values entered should be interpreted as oil rates or as being proportional to oil rates. Rates should be entered as (m^3/day | Stb/day). Not valid for use with *GUIDEI.

*STG

This subkeyword indicates that the gas stream is the reference stream to which the guide rates apply; i.e., that the guide rate values entered should be interpreted as gas rates or as being proportional to gas rates. Rates should be entered in (m^3/day | SCF/day).

*STW

This subkeyword indicates that the water stream is the reference stream to which the guide rates apply; i.e., that the guide rate values entered should be interpreted as water rates or as being proportional to water rates. Rates should be entered in (m^3/day | Stb/day).

*STL

This subkeyword indicates that the guide rates apply to the total of the liquid stream rates (oil+water); i.e., that the guide rate values entered should be interpreted as total liquid rates or as being proportional to total liquid rates. Not valid for use with *GUIDEI. Rates should be entered in (m^3/day | Stb/day).

*STI

This subkeyword identifies that the intermediate liquid stream is the reference stream to which the guide rates apply; i.e. that the guide rate values entered should be interpreted as intermediate liquid rates or as being proportional to intermediate liquid rates. Rates should be entered in (m^3/day | Stb/day). Not valid for use with *GUIDEI.

*WTG

Identifies that guide rate values are wet gas rates. Rates should be entered in (m^3/day | SCF/day). Not valid for use with *GUIDEI.

*BHF

Identifies that guide rate values are Bottomhole fluid rates. Rates should be entered in (m^3/day | rbbl/day). Not valid for use with *GUIDEI.

group_list

A list of group names to which the guide rates are applied. No wildcard characters may be used in the group names.

well_list

A list of well names to which the guide rates are applied. A limited wildcard facility is available for lists of well names (but not for lists of group names). Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. well_names cannot be mixed with group_names in the same list.

guide_rates

A list of guide rates for the list of groups or wells. See the entries for the stream designation strings for the proper units.

DEFAULTS:

Optional keywords. Default is to use the internal guide rate apportionment method (*APPOR-METHOD *INGUIDE). *GUIDEP and *GUIDEI do not have an effect if the apportionment method is defined other than the guide rate method by *APPOR-METHOD *GUIDE.

CONDITIONS:

*GUIDEP and *GUIDEI must be located in the WELL AND RECURRENT DATA keyword group after the wells or groups have been defined.

EXPLANATION:

Guide rates are set for each of the wells specified in 'well_names' or each group specified by the 'group_names'. If the guide rates apportionment method (*APPOR-METHOD *GUIDE) is being used then all producers or injectors connected to that group must have their guide_rates specified with *GUIDEP or *GUIDEI. If no guide rates at all have been specified for the apportionment of the target of a particular group, then instantaneous injection/production potentials are used as the guide rates. Please consult the documentation for more information on the guide rate method under the keyword *APPOR-METHOD.

When *GUIDEI is used, the guide rates apply only for the target specified with the stream identifier; e.g., water injectors have their guide rates specified under *GUIDEI *STW.

When *GUIDEP is used, the guide rates apply for the listed wells or groups to all target types, with the proper conversion made for the ratios at which the wells produce the different streams. That is, the target is apportioned so that the wells produce the reference stream set with the *GUIDEP keyword in the ratios indicated by the rates entered with *GUIDEP. For example, if the following lines are entered when SI units are used,

```
*GUIDEP  *STO  ‘Well 1’ ‘Well 2’ ‘Well 3’
                100.     200.     300.
```

and an STG group target is to be apportioned among these producers, then if Well 1 has a GOR of 600., Well 2 a GOR of 300., and Well 3 a GOR of 200., then the STG guide rates used for the wells would be 60000, 60000, and 60000 respectively and the STG target would be distributed equally among the wells if no other constraints were violated in the process.

Example:

```
*GUIDEP  *STO  'GR-PA'  'GR-PB'  'GR-PC'
                300.00   100.00   400.00
*GUIDEP  *STO  'PA1'    'PA2'    'PA3'
                100.0    100.0    100.0
*GUIDEP  *STL  'PB1'    'PB2'    'PB3'
                100.0    100.0    100.0
*GUIDEP  *STG  'PC1'    'PC2'
                100.0    200.0
*GUIDEI  *STW  'IA1'    'IB1'    'IC1'
                100.0    200.0    100.0
```

Flag for Specifying Groups or Wells Under or not Under Group Apportionment (Optional) *GCPOFF, *GCIOFF, *GCPON, *GCION

PURPOSE:

*GCPOFF or *GCIOFF specifies the accompanying groups or wells are removed from participating in the apportionment of higher-level group production or injection targets.

*GCPON or *GCION specifies the accompanying groups or wells are involved in the apportionment of higher-level group production or injection targets.

FORMAT:

```
*GCPOFF   (group_list)
*GCPON    (well_list)
-or-
*GCIOFF   *GAS      (group_list)
*GCION    *WATER    (well_list)
```

DEFINITIONS:

*GAS, *WATER
: Indicates under or not under group apportionment for gas or water injection calculations (stream identifier *GASI or *WATI is also acceptable).

group_list
: A list of group names. No wildcarding is allowed for group names.

well_list
: A list of well names. A limited wildcard facility is available for lists of well names (but not for lists of group names). Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

DEFAULTS:

Optional keywords. Default is to participate in the group apportionment of higher level target.

CONDITIONS:

These keywords must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

*GCPOFF or *GCIOFF keyword isolates the groups or wells so that they can produce or inject according to their own most restrictive rate and pressure constraints, without being controlled by higher level constraints. However, their rates still contribute to the higher-level targets since the groups or wells are still members of the group structure. They are still subject to the actions taken due to higher-level group constraint violations.

Example:

```
*GCPOFF 'GR-PA'  'GR-PB'  'GR-PC'
*GCPOFF 'PA1'    'PB2'    'PC3'
*GCIOFF *GAS     'GR-PA'
*GCIOFF *WATER   'IB2'    'IC1'
```

Well/Group On-time Fraction (Optional) *ON-TIME

PURPOSE:

*ON-TIME specifies the fraction of time during which a well or group operates. It is useful for history-match or prediction runs in order to obtain the full-flow conditions which correspond to the instantaneous rates of the well or group but with cumulative flows reflecting the down time.

FORMAT:

```
*ON-TIME        'well_names' | 'group_names'
                OTF_input
```

DEFINITIONS:

well_names

Any number of quoted well names to specify the wells whose on-time fractions are defined in this command. These names must be on the same line as the *ON-TIME keyword. If more well names are to be specified than can fit on one line, then another *ON-TIME keyword must be entered. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. For backward compatibility, well_numbers are still allowed, but they cannot be mixed with the names in the same list.

group_names

A list of 'group_names' to which the on-time fractions are applied. No wildcard characters may be used in the group names. These names must be on the same line as the *ON-TIME keyword. If more group names are to be specified than can fit on one line, then another *ON-TIME keyword must be entered. 'group_names' cannot be mixed with 'well_names' in the same list.

OTF_input

The values for the on-time fractions to be assigned to the wells/groups specified. These values must appear on a (or several) new lines immediately after the line which begins with the *ON-TIME keyword. One fraction must be entered for each well/group in the list and the OTF_input are assigned to the wells/groups in the list in order. The n*value format may be used to enter multiple occurrences of the same fraction. Values must be between 0.001 and 1.0, inclusive. Values less than 0.001 are modified to 0.001 with notification.

DEFAULTS:

If no *ON-TIME keyword is encountered, on-time fractions of 1 are assigned to each well/group, and the simulation is not affected. Wells/groups never listed under *ON-TIME retain the on-time fraction of 1 throughout the simulation.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group. A well/group must be defined before it can be assigned options under *ON-TIME. The on-time fraction for a given well/group may be altered by an additional *ON-TIME line at a subsequent well change.

EXPLANATION:

Wells and groups are treated in an entirely analogue manner with wells at the lowest level. The user input on-time factor (OTF_input) for a particular well/group is the on-time fraction relative to the on-time of its parent group. For the field (top level) group, it is relative to the total simulated time. The actual (absolute) on-time factor (OTF_actual), computed internally, is the product of OTF_input's from itself and all the superior groups along the hierarchy branch (up to the field). If no group structure ever exists, the well on-time fractions are relative to the conceptual 'Default-Field', i.e. the total simulated time (and thus OTF_input = OTF_actual). In normal cases, the on-time factors are assigned on only one level of the well/group hierarchy.

The instantaneous rate (Q_instant) for a well refers to the rate when the subject well is running on its full-flow conditions (bhp and whp, et al.). The instantaneous rate for a group, by definition, is the summation of the on-time averaged rates from all its child groups or wells directly attached, which takes into account the lower-level down-time effect. The on-time averaged rate (Q_average) for a well/group is the relative mean rate over the period while its parent group is operating. The actual rate (Q_actual), resulting directly from the material balance, is the reported true volume produced from or injected into the reservoir per unit time. It can also be interpreted as the absolute on-time average, and is used to calculate all the cumulative flows.

At any well/group level, the following relationships hold:

```
Q_average = Q_instant * OTF_input      (relative)
Q_actual  = Q_instant * OTF_actual     (absolute)
```

For any well/group, the instantaneous rate and the on-time averaged rate are the same if OTF_input = 1, since this well/group has no down-time 'relative' to its parent group. This does not necessarily mean that the actual rate equals the instantaneous rate as well. Theoretically, these three rates might have different values. In case that all group on-time fractions are unity (well on-time fractions may not be unity), one expects

```
OTF_actual = OTF_input
Q_actual  = Q_instant * OTF_actual     (absolute)
```

for all levels. Moreover, Q_actual = Q_instant, for all groups (but not the wells). This is the settings in the earlier versions of CMG's Well Management before incorporating the group on-time fraction. It should be noted that these equivalences hold true only at the field (top) level once the group on-time fractions are set < 1.0.

Any flow constraints, targets and limits specified for a well (e.g. *OPERATE, *MONITOR) or group (e.g. *GCONP / *GCONI, *GCONM) will apply to its instantaneous rate. For wells, values of the bottom-hole pressure, drawdown, well layer pressure, and well head pressures which correspond to the specified (instantaneous) rate values are reported. When the well is operating on a bottom-hole pressure or well-head pressure constraint, the rate which results is the actual rate, which has been reduced by the actual on-time fraction. The group on-time factors are meant to model situations in which all wells in a group are brought down at the

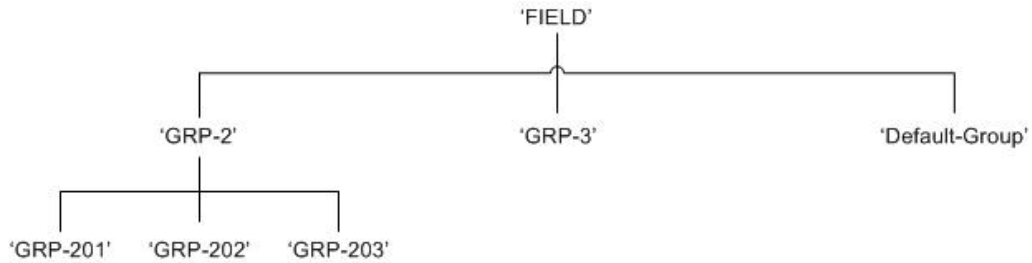
same time and restored to service at the same time; however, wells are still allowed to have individually a non-unit on-time factor when the group factors are in force.

Example:

Assign on-time fractions of 0.95 to wells 1, 3, 7, 8, 9, and 10.

```
*ON-TIME  1  3  7:10
          6*0.95
```

A more sophisticated example is to suppose the following group-well hierarchy:



The field is assigned an oil target (*GCONP/*GTARGET) of 500 which is to be apportioned using the Guide Rate method (*APPOR-METHOD/*GUIDE) with supplied guide rates:

```
*GUIDEP *STO  'GRP-2' 'GRP-3' 'GRP-201' 'GRP-202' 'GRP-203'
       3.      2.      1.        2.        3.
```

In a very general (although not normal) case, the group on-time fractions are input as

```
*ON-TIME 'FIELD' 'GRP-2' 'GRP-3' 'GRP-201' 'GRP-202' 'GRP-203'
     0.75    0.8     0.8     0.85      0.85      0.85
```

which specifies that: Field operates at 75% of the total simulated time; groups GRP-2 and GRP-3 (level 2) operate at 80% of the time while Field is on-time; groups GRP-201, 202, 203 (level 3) operate at 85% of the time while their parent group GRP-2 is operating. The on-time factors for all wells default to unity (OFT_input = 1), suggesting that they operate (100%) as long as their parent groups are operating.

Should all the well constraints have not been violated and the group target has been met, the actual on-time fractions and various flow rates at different levels computed by the simulator read:

	On-Time Factor (OTF)		**Rates (Q)**		
Group Hierarchy	**Input**	**Actual**	**Instant.**	**Average**	**Actual**
FIELD	0.75	0.75	[500.0]	375.0	375.0
GRP-2	0.80	0.60	375.0	300.0	225.0
GRP-201	0.85	0.51	73.5	62.5	37.5
GRP-202	0.85	0.51	147.0	125.0	75.0
GRP-203	0.85	0.51	220.5	187.5	112.5
GRP-3	0.80	0.60	250.0	200.0	150.0
Default-Group	1.00	0.75	/	/	/

[--]: targeted instantaneous value

All wells under group GRP-2 have the actual on-time fractions of 0.51, and those for wells under group GRP-3 are 0.6. The field instantaneous rate by which the target is set is the summation of the on-time averaged rates of its child groups GRP-2 and GRP-3, which can also be converted directly from the field actual rate (500.0 = 300.0 + 200.0 = 375.0/0.75). Similarly, summing the on-time averaged rates of groups GRP-201,202,203 yields the instantaneous rate of their parent group GRP-2, which can also be converted from GRP-2's actual rate by dividing it with GRP-2's actual on-time fraction (375.0 = 62.5 + 125.0 + 187.5 = 225.0/0.6).

As far as the behaviors of lower level groups (and wells) are concerned, the above dataset would run exactly the same if the user directly (and only) define OTF_input = 0.51 for group GRP-2 and OTF_input = 0.6 for group GRP-3. The actual rates and cumulatives for all the wells/groups remain unchanged. However, since now the Field has a unity on-time fraction, i.e. the instantaneous rate is indeed the actual rate for the field, one needs to adjust the field target to 375.0 to render a fair comparison.

	On-Time Factor (OTF)		**Rates (Q)**		
Group Hierarchy	**Input**	**Actual**	**Instant.**	**Average**	**Actual**
FIELD	1.00	1.00	[375.0]	375.0	375.0
GRP-2	0.51	0.51	441.0	225.0	225.0
GRP-201	1.00	0.51	73.5	73.5	37.5
GRP-202	1.00	0.51	147.0	147.0	75.0
GRP-203	1.00	0.51	220.5	220.5	112.5
GRP-3	0.60	0.60	250.0	150.0	150.0
Default-Group	1.00	1.00	/	/	/

[--]: targeted instantaneous value

Sets/Alters a Well's Phase Productivity (Optional) *SETPI

PURPOSE:

*SETPI sets/multiplies a well's specified phase productivity/injectivity to/by a user defined value.

FORMAT:

```
*SETPI    (*OIL)  ('well_names') or (well_numbers) values
          (*WAT)
          (*LIQ)
          (*GAS)
          (*INL)
          (*TOT)
          (*MULT)
          (*MULTO)
```

DEFINITIONS:

*OIL

This subkeyword identifies that the PI of oil will be set to the values specified. Units are (m3/day/kPa | Stb/day/psi)

*WAT

This subkeyword identifies that the PI of water will be set to the values specified. Units are (m3/day/kPa | Stb/day/psi)

*GAS

This subkeyword identifies that the PI of gas will be set to the values specified. Units are (m3/day/kPa | scf/day/psi)

*INL

This subkeyword identifies that the PI of the intermediate (INL) stream will be set to the values specified. Units are (m3/day/kPa | Stb/day/psi)

*LIQ

This subkeyword identifies that the PI of the sum of oil, intermediate (INL) stream and water will be set to the values specified. Units are (m3/day/kPa | Stb/day/psi)

*TOT

This subkeyword identifies that the PI of the sum of oil, water, intermediate (INL) stream and gas will be set to the values specified. Units are (m3/day/kPa | scf/day/psi). Note the units of *TOT is assumed to be scf/day/psi even if only liquid flow is encountered (not Stb/day/psi as is usual for liquids).

MULT

This subkeyword identifies that the current well indices and phase PI's will be multiplied by the values which follow this keyword. Existing well indices that may have been modified by previous uses of *SETPI will be multiplied.

*MULTO

This subkeyword identifies that unmodified/original well indices and phase PI's will be multiplied by the values that follow this keyword. Existing well indices that may have been modified/multiplied by previous uses of *SETPI will be ignored. The well indices that are multiplied by this keyword are directly obtained from actual model/reservoir properties.

A *MULTO factor of 1.00 is useful to return to the actual well productivity after several applications of the *SETPI keyword.

well_names

Any number of well names (in quotes) to specify the wells to which this alteration of the primary operating constraint applies. These names must be on the same line as the *SETPI keyword.

Note: wildcards may be used in the 'well_names' string as follows:

* replaces any number of characters at the end of a well name or can be used on its own to replace all wells (e.g. *SETPI *LIQ '*' or *SETPI *WAT 'wel*').

? replaces any single character anywhere in the well name (e.g. *SETPI *GAS '?ell1').

The two wild cards can be combined on any list and when wild cards are used the well list generated is printed out for the user to check.

well_numbers

Any number of integers, or a range of integers to specify the well numbers to which this alteration of primary operating constraint applies. These well numbers must be on the same line as the *SETPI keyword. If more than one line is required to specify all the wells then the *SETPI keyword must be repeated.

values

One number for each well identified by well_numbers or 'well_names' (or a single value for all wells) specifying the new value of the primary operating constraint. Values must appear on one or more new lines immediately following the *SETPI line.

If only one value of 'values' is input, all wells defined in the well list will be set to that single value.

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

*SETPI must be located in the Well and Recurrent Data keyword group, and may appear anywhere in this keyword group following the initial specification and perforation of the wells in the *SETPI list.

The use of *SETPI is only valid for producers and total mobility weighted injectors.

*SETPI calculates layer productivity ignoring the effects of both the *TURB and *QUAD keywords. That is: turbulence and quadratic pressure dependence is ignored in the productivity index calculation used to generate the wells modified well index.

*SETPI does not account for crossflow when working out productivity indices. If backflow does occur, the well's actual productivity will differ from the *SETPI value.

When PI is set for phases which currently do not exist at the well perforations, the following structure of defaults is used:

*GAS	reverts to *TOT
*WAT	reverts to *LIQ
*OIL	reverts to *LIQ
*LIQ	reverts to *TOT
*INL	reverts to *LIQ

A warning is issued when this occurs.

If the total mobility was found to be zero (an unlikely event), then the *SETPI value will be ignored for that well.

EXPLANATION:

A well's productivity index is defined as the ratio of a production rate divided by a drawdown.

The production rate may be gas production, liquid production, intermediate (INL) stream production, oil production, water production or total production, hence *SETPI can be used to set any one of these.

The essence of the *SETPI calculation is to allow the simulator to first generate its estimate of well productivity based on the well index calculated and the mobility of each phase. The wells basic productivity with respect to any of the phases is then available.

The new desired productivity (oil, water, gas, inl, liquid, or total) is used to scale the well index to produce the desired value.

The well's productivity will naturally change with time due to changes in the fluids near the wellbore. *SETPI can be entered whenever a well test was performed to account for known changes in well productivity.

The *SETPI calculation does not include solution gas when an oil productivity is specified, it is assumed the gas is separated from the oil before oil flow is measured.

Example 1:

```
*INUNIT  *FIELD
*WELL 1 'PRODUCER'
*WELL 2 'INJECTOR'
**             rad    geofac   wfrac   skin
*GEOMETRY *K  0.375   0.2488   1.0     -1.0
```

```
*PERF *GEO 1
**  if    jf    kf    ff   (setn)
    12    6    2:4   1.0     1
    13    6      5    .5     2
*PERF *KH  2
** if   jf    kf   kh      di    (setn)
    1    7   1:5   20   0.00002    2
    2    7   1:5   30   0.00004    1
*SETPI  *LIQ  1
5.0
*SETPI *GAS  'INJECTOR'
1.0d05
```

In the above example, the SETPI would force the liquid well productivity for WELL NO. 1 to be 5.0 bbl/day/psi. However, the original well index is retained and can be used if MULTO (multiply original) is used. For WELL NO. 2, the gas injectivity index will be set to 1.0e5 ft3/day/psi, without taking the (TURB) or (QUAD) into effect.

Example 2:

```
*INUNIT  *SI
*WELL 1 'PRODUCER'
*WELL 2 'INJECTOR'
**             rad    geofac   wfrac   skin
*GEOMETRY *K  0.375   0.2488    1.0    -1.0
*PERF *GEO 1
**  if   jf  kf    ff    (setn)
    12   6   2:4   1.0     1
*PERF *KH  2
**  if   jf  kf    kh      di     (setn)
     1    7  1:5   20   0.00002     2

** This will set the liquid well productivity for
** WELL No. 1 to 5.0 m3/day/kPa.
*SETPI  *LIQ  1
              5.0
** This will set the water well injectivity for WELL
** No. 2 to 8.0 m3/day/kPa.
*SETPI *WAT  2
            8.0
** This will set the water well injectivity for WELL
** No. 2 to 8.0 m3/day/kPa.
*SETPI *WAT  2
            8.0
DATE 1990 1 1

** This will set the total well injectivity for both
** wells to 7.0 m3/day/kPa.
*SETPI *TOT 1:2
           7.0
*DATE 1990 2 1

** This will reset the well index to its original value
*SETPI *MULTO 1:2
              1.0
```

```
*DATE 1990 3 1
** This will multiply the latest well index for both
**wells by 2.3
*SETPI *MULT '*'
            2.3

*DATE 1990 4 1
**This will multiply the latest well index for WELL No.
**1 by 1.5 and by 2.3 for WELL No. 2. For WELL No. 1 –
**the new well index is 3.45 ( 2.3*1.5) times the
**original , while for WELL No. 2 the current well
**index is 7.36 (2.3*3.2)  times the original.
*SETPI *MULT 1:2
            1.5 3.2
*INJECTOR 4
*MONITOR *MAX *WATER 100.0
:
*TIME 1200.
** At a later date, want to adjust the operating
** constraint values.
** well_numbers
*ALTER  1:2   3   4
** values
    2*1000.0  800.0  50.0
```

Individual Layer Controls (Optional) *LAYER-CTRL

PURPOSE:

Modify layer status or well indices for individual completions.

FORMAT:

*LAYER-CTRL *wn* action
{ *locat.* }
where
action = *OPEN | *CLOSE | *WIMULT *val* | *WIMULTO *val*

DEFINITIONS:

wn
A single well sequence number or well name enclosed in single quotes.

locat.
Layer location(s) identified by the User Block Address (UBA) that has been previously defined by the *PERF (or *PERFV) keyword for well *wn*.

*CLOSE
Action to close the currently active layer(s). Well *nw* will be shutin if there is no open layer left.

*OPEN
Action to open the currently active layer(s). Well *nw* will be opened if (and only if) it was previously shutin due to no open layers.

*WIMULT *val*
Action to multiply the current well indices by the non-negative value of *val*. See keyword *SETPI *MULT for details.

*WIMULTO *val*
Action to multiply the original well indices by the non-negative value of *val*. See keyword *SETPI *MULTO for details. A *WIMULTO factor of 1.0 is useful to return to the original well indices after several cumulative multiplications.

DEFAULTS:

Optional keywords. No defaults. If *LAYER-CTRL is specified without any layer UBA, the action will be applied to all the layers of this well.

CONDITIONS:

*LAYER-CTRL must be located in the Well and Recurrent Data keyword group, and may appear anywhere in this keyword group following the initial specification and perforation of the well.

EXPLANATION:

*LAYER-CTRL allows the user to change the layer status (*OPEN and *CLOSE) of individual layers without having to redefine the whole perforation of the well by the *PERF (or *PERFV) keyword. Similar to the *SETPI keyword, *LAYER-CTRL also allows the user to modify the well indices but for particular layers through the well index multipliers (*WIMULT and *WIMULTO). See *SETPI for detailed explanations. If the well has been recompleted by a new *PERF (or *PERFV) keyword, the layer multiplier specified by previous *LAYER-CTRL remains as long as the layer(s) is still active.

Example 1**:**

```
*WELL 1 'PRODUCER'
*WELL 2 'INJECTOR'

*PERF *GEO 1
   7 3 3 1.0
   7 3 2 1.0
   7 3 1 1.0

*PERF *GEO 2
   1 9 4 1.0
   1 9 3 1.0

TIME 365.0
*LAYER-CTRL 1 *WIMULT 2.0
   7 3 3

*LAYER-CTRL 2 *CLOSE
   1 9 4
```

Hydraulic Pressure Table (Conditional) *PTUBE1 / *ITUBE1

PURPOSE:

*PTUBE1 or *ITUBE1 introduce input of tubing-head (or outlet) pressure data tables for a production or injection well, or manifold group. The tabulated variable is the bottom-hole (or inlet) pressure, with tubing-head (or outlet) pressure and several independent variables such as surface rates or rate ratios. *PTUBE1 can also be used to input the well-head temperature table.

FORMAT:

```
*PTUBE1 / *ITUBE1          table_number
*DEPTH  refdepth

Flo
  flo(1) ... flor(nflo)

Gfr
  gfrl(1) ... gfrl(ngfr)

Wfr
  wfr(1) ... wfr(nwfr)

Add
  add(1) ... add(nadd)

*WHP
  whp(1) ... whp(nwhp)

*BHP | *WHT
** iflo        igfr        iwfr        iadd        bhp(1)...        bhp(nwhp)
   :            :           :           :           :                :
```

DEFINITIONS:

*DEPTH

This subkeyword introduces input of the reference depth (m | ft).

Flo

Independent Flow Rate variable. All rates are measured at surface conditions. For *PTUBE1, it is identified by one of the following subkeywords:

*OIL	Oil production rate (m3/day | Stb/day);
*LIQ	Liquid (oil+water) production rate (m3/day | Stb/day);
*GAS	Gas production rate (m3/day | SCF/day);
*WTG	Wet-gas production rate (m3/day | SCF/day);

*TML	Total molar production rate (gmol/day).

For *ITUBE1, it is identified by one of the following subkeywords:

*GAS	Gas injection rate (m3/days | SCF/day);
*WAT	Water injection rate (m3/day | Stb/day);
*TML	Total molar injection rate (gmol/day).

Gfr

Independent Gas Fraction variable (*PTUBE1 only), identified by one of the following subkeywords:

*GOR	Gas-oil ratio (m3/m3 | SCF/STB);
*GLR	Gas-liquid ratio (m3/m3 | SCF/STB);
*OGR	Oil-gas ratio (m3/m3 | STB/SCF);
*AMW	Average molecular weight (gram/gmol).

Wfr

Independent Water Fraction variable (*PTUBE1 only), identified by one of the following subkeywords:

*WOR	Water-oil ratio,
*WCUT	Water cut (water-liquid ratio);
*WGR	Water-gas ratio (m3/m3 | STB/SCF);
*WWR	Water-wet gas ratio;
*WMF	Water molar fraction.

Add

Independent Additional look-up variable, identified by one of the following subkeywords:

*LFG	Lift gas injection rate (m3/day | SCF/day);
*LFR	Injection gas-liquid ratio for gas lift (m3/m3 | SCF/STB);
*LFT	Total gas-liquid ratio for gas lift (m3/m3 | SCF/STB);
*DNO	Surface mass density of oil (kg/m3 | lb/ft3). It takes the volumetric-weighted average value for a manifold group;
*DNG	Surface mass density of gas (kg/m3 | lb/ft3). It takes the volumetric-weighted average value for a manifold group;
*ALQ	Artificial lift quantity. Since its interpolation values are input through keyword *WELLALQ for individual wells or *GROUPALQ for indicial groups, the variable itself becomes arbitrary and can be specified freely in the data, such as pump rating, compressor power, choke diameter for segmented wells. The unit of the variable is therefore not important so long as the assigned values and the tabular values are consistent;

*ENT Well-head enthalpy (J/gram | Btu/lb). This variable has an effect only if the thermal option is turned on (*THERMAL *ON) in the **Component Properties Section**.

All variables are valid for *PTUBE1, but only *ENT is valid for *ITUBE1.

*WHP

This keyword specifies the well-head (or manifold outlet) pressures (kPa | psi).

*BHP

This keyword introduces input of the bottom-hole (or manifold inlet) pressure table (kPa | psi).

*WHT

Mutually exclusive with *BHP, this keyword introduces input of the well-head temperature table (°C | °F) and is valid for *PTUBE1 only.

refdepth

Non-zero real number specifying the reference depth for the pressure table.

table_number

Hydraulic table number. Tables must be defined in order starting with table 1 and increasing sequentially.

flo(1) ... flo(nflo)
gfr(1) ... gfr(ngfr)
wfr(1) ... wfr(nwfr)
add(1) ... add(nadd)
whp(1) ... whp(nwhp)

Real numbers specifying the values of the first through the last entry of the corresponding variable. At least one (1) entry must be entered for each identified variable. Values must be monotonically increasing, respectively. Gfr, Wfr and Add (except for *ENT) cannot be entered for ITUBE1.

iflo, igfr, iwfr, iadd

Integers specifying the corresponding variable index. They must be in the fixed order for *PTUBE1 and only “iflo” (and “iadd” if *ENT presents) shall be entered for *ITUBE1 when reading the *BHP table.

bhp(1) ... bhp(nwhp)

Real numbers specifying the values of bottom-hole (or manifold inlet) pressure corresponding to the variable values specified by the integer indices iflo, etc. and well-head (or manifold outlet) pressure values (1, ..., nwhp). If one or several of the BHP values is not available (for example, if the program used to compute the BHP values did not converge for these conditions), the keyword *EXTP may be entered in place of the value. In this case the missing values in a row of the table are determined by linear interpolation or extrapolation from entered values in the same row. At least one BHP value (as opposed to the string *EXTP) must be entered in each row of the table.

If *WHT is specified instead of *BHP, the table values are for the well-head temperature.

DEFAULTS:

Conditional keyword for wells but optional for manifold groups. No defaults.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Required when *WHP is one of the well constraints or when the wellbore model is desired, and the TABLE is referred to. All the information must be entered before the subkeyword *BHP for reading the pressure table. All the variables, except for the *LFG, *LFR and *LFT and *ENT in the Add variable item, apply to the manifold groups. All the values in the table must be non-negative except for the *ENT entries.

EXPLANATION:

*PTUBE1 or *ITUBE1 sets up the independent variable and WHP values for the *BHP tables for wells (or manifold groups). The bottom-hole (or manifold inlet) pressure table is also entered here (see related keywords *IWELLBORE, *PWELLBORE, *GLCONTROL and *GPTABLE). When *GLIFT is in effect, the lift gas injection rate is added to the producer's formation gas rate before calculating the wellbore pressure drop and well liquid production rates, unless the *LFG, *LFR or *LFT is specified as the Add variable in *PTUBE1.

The depth required with *PTUBE1/*ITUBE1 is a well (or a manifold) depth for an individual well (or manifold, see keyword *GPHYDDEP). The depth is used to adjust the tabulated bottom-hole (or manifold inlet) pressures from the table depth to the actual well (or manifold outlet) depth using linear scaling. In this way several wells (or groups) with different depths may use the same hydraulics pressure table. Total well length should be entered for horizontal tables.

The bottom-hole (or manifold) pressures read under *PTUBE1/*ITUBE1 correspond to values of the defined variables (rates, surface ratios, water-cuts, etc.). The number of bottom-hole (or manifold inlet) pressures entered for each set of index integers must be equal to the number of well-head (or manifold outlet) pressures entered under *WHP.

For instance, if three well-head pressures were entered, then there must be four columns (iflo, igfr, iwfr, iadd) containing bottom-hole pressures for *PTUBE1 or one column (iflo) for *ITUBE1, or two columns (iflo, iadd) for ITUBE1 if *ENT presents. The integer indices identify the particular values of the defined variables among those entered, and must be in the fixed order. The first entered BHP value corresponds to the first WHP value, and so on.

The *PTUBE1 format can also be used to input the well-head temperature if the header of the table body is *WHT rather than *BHP. See keyword *TWELLBORE for details.

Example 1: Oil Production

```
*PTUBE1 1
*DEPTH 5000.0
*OIL   **flo(1)  flo(2)  flo(3)  flo(4)
         0.0     4000.0  8000.0  16000.0
*GOR   **gfr(1)  gfr(2)
         500.0   1000.0
*WCUT  **wfr(1)  wfr(2)
         0.00    0.50
*ALQ   **add(1)
         0.0
*WHP   **whp(1)  whp(2)  whp(3)
         200.0   900.0   1600.0
```

```
*BHP
**iflo igfr iwfr iadd  bhp(1)  bhp(2)  bhp(3)
   1    1    1    1    2176.1  2873.7  3573.7
   1    1    2    1    2646.7  3344.7  4044.7
   1    2    1    1    1997.7  2670.9  3370.9
   1    2    2    1    2447.7  3124.7  3824.7
   2    1    1    1    2135.5  2876.6  3576.6
   2    1    2    1    2618.0  3351.2  4051.2
   2    2    1    1    1682.7  2674.6  3374.6
   2    2    2    1    2189.0  3132.3  3832.3
   3    1    1    1    2133.6  2884.2  3584.2
   3    1    2    1    2630.9  3368.4  4068.4
   3    2    1    1    1463.1  2684.5  3384.5
   3    2    2    1    2022.0  3152.8  3852.8
   4    1    1    1    2160.1  2912.5  3612.5
   4    1    2    1    2696.4  3433.4  4133.4
   4    2    1    1    1425.7  2721.3  3421.3
   4    2    2    1    2080.0  3231.0  3931.0
```

Example 2: Water Injection

```
*ITUBE1 1
*DEPTH 1800.0
*WAT
** flo(1) flo(2) flo(3) flo(4) flo(5) flo(6)
   0.0    100.0  200.0  350.0  500.0  600.0
*WHP
**whp(1)   whp(2)   whp(3)   whp(4)   whp(5)
  101.325  10000.0  20000.0  30000.0  40000.0
*BHP
**iflo bhp(1)  bhp(2)  bhp(3)  bhp(4)  bhp(5)
   1   17820.0 27794.0 37872.0 47949.0 58026.0
   2   17816.0 27791.0 37868.0 47945.0 58022.0
   3   17813.0 27788.0 37865.0 47942.0 58019.0
   4   17807.0 27782.0 37859.0 47936.0 58013.0
   5   17802.0 27777.0 37854.0 47931.0 58008.0
   6   17798.0 27773.0 37850.0 47927.0 58004.0
```

Example 3: Gas Production in a Gas-Water run

```
*PTUBE1 1
*DEPTH 5000.0
*GAS **flo(1)  flo(2)   flo(3)   flo(4)
       0.0     40000.0  80000.0  160000.0
*GOR **gfr(1)
       0.0
*WGR **wfr(1)  wfr(2)
       0.00    0.01
*ALQ **add(1)
       0.0
*WHP **whp(1)  whp(2)  whp(3)
       200.0   900.0   1600.0
```

```
*BHP
**iflo igfr iwfr iadd  bhp(1)  bhp(2)  bhp(3)
   1    1    1    1    2176.1  2873.7  3573.7
   1    1    2    1    2646.7  3344.7  4044.7
   2    1    1    1    2135.5  2876.6  3576.6
   2    1    2    1    2618.0  3351.2  4051.2
   3    1    1    1    2133.6  2884.2  3584.2
   3    1    2    1    2630.9  3368.4  4068.4
   4    1    1    1    2160.1  2912.5  3612.5
   4    1    2    1    2696.4  3433.4  4133.4
```

Example 4: Oil Production with extrapolation

```
*PTUBE1  1
*DEPTH 5000.0
*OIL  **flo(1)  flo(2)  flo(3)  flo(4)
        0.0     4000.0  8000.0  16000.0
*GOR  **gfr(1)  gfr(2)
        500.0   1000.0
*WCUT **wfr(1)  wfr(2)
        0.00    0.50
*ALQ  **add(1)
        0.00
*WHP  **whp(1)  whp(2)  whp(3)
        200.0   900.0   1600.0
*BHP
**iflo igfr iwfr iadd  bhp(1)  bhp(2)  bhp(3)
   1    1    1    1    2176.1  2873.7  3573.7
   1    1    2    1    2646.7  3344.7  4044.7
   1    2    1    1    *EXTP   2670.9  3370.9
   1    2    2    1    2447.7  3124.7  3824.7
   2    1    1    1    2135.5  2876.6  3576.6
   2    1    2    1    *EXTP   3351.2  *EXTP
   2    2    1    1    1682.7  2674.6  3374.6
   2    2    2    1    2189.0  3132.3  3832.3
   3    1    1    1    2133.6  2884.2  *EXTP
   3    1    2    1    2630.9  3368.4  4068.4
   3    2    1    1    1463.1  2684.5  3384.5
   3    2    2    1    2022.0  *EXTP   3852.8
   4    1    1    1    2160.1  2912.5  3612.5
   4    1    2    1    2696.4  3433.4  4133.4
   4    2    1    1    1425.7  2721.3  3421.3
   4    2    2    1    2080.0  3231.0  3931.0
```

Row '2 1 2 1', which has two of three values extrapolated, will have constant BHP values of 3351.2.

Example 5: Production in a condensate field

```
*PTUBE1 5
*DEPTH 8000.0
*GAS **flo(1)  flo(2)  flo(3)  flo(4)
       0.0     1.0e+6  3.0e+6  7.0e+6
*OGR **gfr(1)  gfr(2)
       0.00001 0.00002
*WGR **wfr(1)  wfr(2)
       0.0     0.0001
```

```
*ALQ **add(1)
       0.0
*WHP **whp(1)  whp(2)  whp(3)
       500.0   1500.0  2500.0
*BHP
** iflo igfr iwfr iadd  bhp(1)  bhp(2)  bhp(3)
    1    1    1    1    2176.1  2873.7  3573.7
    1    1    2    1    2646.7  3344.7  4044.7
    1    2    1    1    1997.7  2670.9  3370.9
    1    2    2    1    2447.7  3124.7  3824.7
    2    1    1    1    2135.5  2876.6  3576.6
    2    1    2    1    2618.0  3351.2  4051.2
    2    2    1    1    1682.7  2674.6  3374.6
    2    2    2    1    2189.0  3132.3  3832.3
    3    1    1    1    2133.6  2884.2  3584.2
    3    1    2    1    2630.9  3368.4  4068.4
    3    2    1    1    1463.1  2684.5  3384.5
    3    2    2    1    2022.0  3152.8  3852.8
    4    1    1    1    2160.1  2912.5  3612.5
    4    1    2    1    2696.4  3433.4  4133.4
    4    2    1    1    1425.7  2721.3  3421.3
    4    2    2    1    2080.0  3231.0  3931.0
```

Example 6: Gas injection with thermal effect

```
*ITUBE1  1
*DEPTH  7500.00
 *GAS
  0.00000  6.31037E+07  1.26207E+08
 *ENT
  170626.    204751.    238876.    273001.
 *WHP
  744.729    1493.24    2241.76    2990.27    3738.78
 *BHP
   1   1  2674.90     3100.16     3530.37     3558.83     3738.78
   2   1  2732.49     3162.57     3573.41     3752.26     3882.27
   3   1  2557.68     3270.90     3628.63     3839.66     3992.08
   1   2  2149.02     2803.08     3367.64     3558.82     3738.78
   2   2  2221.46     2866.67     3406.75     3732.76     3882.27
   3   2  2287.37     2934.14     3381.75     3797.02     3992.09
   1   3  1816.58     2476.25     3096.37     3558.82     3738.78
   2   3  1872.59     2437.51     3153.79     3680.39     3882.27
   3   3  1969.51     2426.28     3056.54     3647.97     3992.12
   1   4  1377.08     2019.03     2727.00     3555.06     3738.78
   2   4  1472.38     2042.63     2707.98     3423.48     3882.27
   3   4  1632.97     2034.18     2574.03     3245.88     3987.89
```

The acceptable range of values for reference depth is:

	SI m	Field ft
Min	-1.0E+4	-32,808.0
Max	1.0E+4	32,808.0

Backward Compatibility Notes:

The previous tubing data keyword *PTUBE is obsolete. Its conversion to *PTUBE1 is given as follows:

```
*PTUBE *OIL                 *PTUBE1
 *QO          (iqol)         *OIL         (iflo)
 *GOR         (igor)         *GOR         (igfr)
 *WCUT        (iwcut)        *WCUT        (iwfr)
  ---                        *ALQ         (iadd)
 *BHPTO                      *BHP

*PTUBE *LIQ                 *PTUBE1
 *QO/*QLIQ    (iqol)         *LIQ         (iflo)
 *GOR         (igor)         *GLR         (igfr)
 *WCUT        (iwcut)        *WCUT        (iwfr)
  ---                        *ALQ         (iadd)
 *BHPTO                      *BHP

*PTUBE *WATER_GAS           *PTUBE1
 *QG          (iqg)          *GAS         (iflo)
  ---                        *OGR         (igfr)
 *WGR         (iwgr)         *WGR         (iwfr)
  ---                        *ALQ         (iadd)
 *BHPTG                      *BHP

*PTUBE *CONDENSATE          *PTUBE1
 *QG          (iqg)          *GAS         (iflo)
 *OGR         (iogr)         *OGR         (igfr)
 *WGR         (iwgr)         *WGR         (iwfr)
  ---                        *ALQ         (iadd)
 *BHPTC                      *BHP
```

For blanks in the *PTUBE format (---), add one entry of zero value for the corresponding variable in the *PTUBE1 format. Subkeywords *DEPTH and *WHP remain the same.

BUILDER will automatically convert *PTUBE to *PTUBE1. Well Management can still read the obsolete tubing table formats (*PTUBE, *VFPPROD and *VFPINJ), but can convert them into the new formats (*PTUBE1 and *ITUBE1) by command-line option “-newptube”. The new table(s) will be printed to the output file following the echo of the obsolete keywords.

Allow a Set of Keywords to be Processed When a Specified Condition (Trigger) is Satisfied (Optional) *TRIGGER

PURPOSE:

*TRIGGER allows the user to specify certain actions which are implemented when a specific condition(s) is/are satisfied during the simulation.

FORMAT:

*TRIGGER ‘trig_name’ trig_def1 *INCREMENT rinc1 *AVRGTIME avrgt1 *ANDnn /ORnn trig_def 2 *INCREMENT rinc2 *AVRGTIME avrgt2 *APPLY_TIMES napt *TEST_TIMES ntestt

*TEST_AFTER_TIMER rtimedr *TEST_AFTER_TIMEA rtimeda

{list of actions}

*END_TRIGGER

Where **trig_def1/trig_def2** is one of

(*ON_WELL 'well_names' well_condition operator condition_value)

-or-

(*ON_GROUP 'group_names' group_condition operator condition_value)

-or-

(*ON_LAYER 'well_name' layer_UBA layer_condition operator condition_value)

-or-

(*ON_SECTOR 'sector_name' sector_condition operator condition_value)

-or-

(*ON_FIELD 'FIELD' field condition operator condition_value)

-or-

(*ON_ELAPSED ‘TIME’ time condition operator condition_value)

-or-

(*ON_REPGRP ‘reporting group name’ rep_ group_condition operator condition_value)

-or-

(*ON_CTRLLUMP ‘control/reporting lump name’ condition operator condition_value)

-or-

(*ON_CYCGRP ‘cycling group name’ CYC-END operator condition_value)

And where **{list of actions}** represents valid well and recurrent data keyword lines**.**

DEFINITIONS:

trig_name

Enter a string (in single quotes) of less than 40 characters to uniquely identify the trigger. The name must immediately follow the *TRIGGER keyword. This token is required.

*ANDnn

Logical operator connecting two conditions. Both conditions must be evaluated as TRUE for logical statement to be TRUE as well. The integer following the *AND keyword assigns a priority ranking to the operator. For example AND1 implies that the logical statement consisting of two conditions and the connector AND will be evaluated first; AND3 implies that this operator and the associated conditions will be evaluated third etc. This token is optional.

*ORnn

Logical operator connecting two conditions. If either condition is evaluated as TRUE then the logical statement is TRUE as well. The integer following the *OR keyword assigns a priority ranking to the operator. For example OR1 implies that the logical statement consisting of two conditions and the connector OR will be evaluated first. OR3 implies that this operator and the associated conditions will be evaluated third etc. This token is optional.

*ON_WELL

This subkeyword indicates that the test condition is to be applied to a well or list of wells (list, if a wildcard is used in the well name or a list of wells is specified). One of *ON_WELL or *ON_GROUP or *ON_SECTOR or *ON_LAYER or *ON_FIELD or *ON_ELAPSED or *ON_REPGRP or *ON_CTRLLUMP or *ON_CYCGRP is a required token immediately following the trigger name string.

well_names

Any number of well names (in quotes) to specify the wells to which this trigger applies. The well(s) must be previously defined. The well names must all be specified on a single line.

Note: wildcards may be used in the 'well_names' string as follows:

* replaces any number of characters at the end of a well name or can be used on its own to replace all wells (e.g. *TRIGGER ‘trig1’ *ON_WELL '*' or *TRIGGER ‘trig1’ *ON_WELL 'wel*').

? Replaces any single character anywhere in the well name (e.g. *TRIGGER ‘trig1’ *ON_WELL '?ell1').

The two wild cards can be combined on any list and when wild cards are used the well list generated is printed out for the user to check.

If *ON_WELL is specified, then at least one well name must immediately follow the *ON_WELL keyword.

Well_condition:

Enter a single keyword identifying **one** of the following: a well stream rate or cumulative or composition or well bottomhole or tubing head pressure or backflow. The valid lists of conditions for a well are shown in Table 1 below: This is a required token and must immediately follow the well name or well list.

Table 1: Well quantities:

Subkeyword	Meaning
STO-RP	Stock Tank Oil - Rate of Production
STO-CP	Stock Tank Oil - Cumulative Production
STO-RI	Stock Tank Oil – Rate of Injection
STO-CI	Stock Tank Oil – Cumulative Injection
STW-RP	Stock Tank Water – Rate of Production
STW-CP	Stock Tank Water – Cumulative Production
STW-RI	Stock Tank Water – Rate of Injection
STW-CI	Stock Tank Water – Cumulative Injection
STG-RP	Stock Tank Gas – Rate of Production
STG-CP	Stock Tank Gas – Cumulative Production
STG-RI	Stock Tank Gas – Rate of Injection
STG-CI	Stock Tank Gas – Cumulative Injection
STL-RP	Stock Tank Liquid – Rate of Production
STL-CP	Stock Tank Liquid – Cumulative Production
BHF-RP	Bottomhole Fluid – Rate of Production
BHF-CP	Bottomhole Fluid – Cumulative Production
BHF-RI	Bottomhole Fluid – Rate of Injection
BHF-CI	Bottomhole Fluid – Cumulative Injection
STI-RP	Stock Tank Intermediate liquid – Rate of Production
STI-CP	Stock Tank Intermediate liquid – Cumulative Production
WTG-RP	Wet Gas – Rate of Production
WTG-CP	Wet Gas – Cumulative Production
BHP	Bottomhole Pressure
WHP	Well Head Pressure
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
MPWS MXX	Mole percent of component XX in produced hc well stream
MPVS MXX	Mole percent of component XX in produced separator vapor hc stream
MPLS MXX	Mole percent of component XX in produced separator liquid hc stream
BKFLOW	Back flow (true if any layer is back flowing)
GLIFTR	Gas lift rate
GLIFTC	Gas lift cumulative
GGLOR	Gas lift rate plus gas production rate over oil rate ratio
GGLIFTR	Gas lift rate plus gas production rate
GGLIFTC	Gas lift cumulative plus gas produced cumulative
GGLLIQR	Gas lift rate plus gas production rate over liquid rate ratio
GLOILR	Gas lift rate over oil rate ratio
GLLIQR	Gas lift rate over liquid rate ratio
WBHTEMP	Bottom Hole temperature
WTHTEMP	Tubing Head temperature, see keyword *TWELLBORE and *PTUBE1.

*ON_GROUP

This subkeyword indicates that the test condition is to be applied to a group or list of groups (if a wildcard is used in the group name or a list of groups is specified). The group hierarchy must be previously defined to enable group based triggers to be used. One of *ON_WELL or *ON_GROUP or *ON_SECTOR or *ON_LAYER or *ON_FIELD or *ON_ELAPSED or *ON_REPGRP is a required token immediately following the trigger name string.

group_names

Any number of group names (in quotes) to specify the groups to which this trigger applies. The groups(s) must be previously defined. There are no groups created by default. The names must all be specified on a single line.

Note: wildcards may be used in the 'group_names' string as follows:

* replaces any number of characters at the end of a group name or can be used on its own to replace all groups (e.g. *TRIGGER ‘trig1’ *ON_GROUP '*' or *TRIGGER ‘trig1’ *ON_GROUP 'grp*').

? Replaces any single character anywhere in the group name (e.g. *TRIGGER ‘trig1’ *ON_GROUP '?rp1').

The two wild cards can be combined on any list and when wild cards are used the group list generated is printed out for the user to check.

At least one group name must immediately follow the *ON_GROUP keyword. Required token.

Group_condition:

The valid list of quantities for groups and for the field is shown in Table 2 below:

Table 2: Group and Field quantities:

Subkeyword	Meaning
STO-RP	Stock Tank Oil – Rate of Production
STO-CP	Stock Tank Oil – Cumulative Production
STW-RP	Stock Tank Water – Rate of Production
STW-CP	Stock Tank Water – Cumulative Production
STW-RI	Stock Tank Water – Rate of Injection
STW-CI	Stock Tank Water – Cumulative Injection
STG-RP	Stock Tank Gas – Rate of Production
STG-CP	Stock Tank Gas – Cumulative Production
STG-RI	Stock Tank Gas – Rate of Injection
STG-CI	Stock Tank Gas – Cumulative Injection
STL-RP	Stock Tank Liquid – Rate of Production
STL-CP	Stock Tank Liquid – Cumulative Production
BHF-RP	Bottomhole Fluid – Rate of Production
BHF-CP	Bottomhole Fluid – Cumulative Production

BHF-RI	Bottomhole Fluid – Rate of Injection
BHF-CI	Bottomhole Fluid – Cumulative Injection
STI-RP	Stock Tank Intermediate liquid – Rate of Production
STI-CP	Stock Tank Intermediate liquid – Cumulative Production
WTG-RP	Wet Gas – Rate of Production
WTG-CP	Wet Gas – Cumulative Production
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
MPWS MXX	Mole percent of component XX in group production
MPVS MXX	Mole percent of component XX in produced separator vapor hc stream summed over all producers in the group
MPLS MXX	Mole percent of component XX in produced separator liquid hc stream summed over all producers in the group
GWGR	Gas-Wet Gas Ratio (production)
WWGR	Water-Wet Gas Ratio (production)
RECYSTG	Group Recycled Gas injection rate
RECYSTW	Group Recycled Water injection rate
VOIDRPG	Group voidage replacement ratio by gas injection.
VOIDRPW	Group voidage replacement ratio by water injection
VOIDRPT	Group voidage replacement ratio by all injection streams
GLIFTR	Gas lift rate
GLIFTC	Gas lift cumulative
GGLOR	Gas lift rate plus gas production rate over oil rate ratio
GGLIFTR	Gas lift rate plus gas production rate
GGLIFTC	Gas lift cumulative plus gas produced cumulative
GGLLIQR	Gas lift rate plus gas production rate over liquid rate ratio
GLOILR	Gas lift rate over oil rate ratio
GLLIQR	Gas lift rate over liquid rate ratio

*ON_LAYER

This subkeyword indicates that the test condition is to be applied to a well layer. The well must be fully defined previously for a trigger based on a layer condition to be specified.

well_name

A single string (in quotes) of less than 40 characters representing a well name must immediately follow the *ON_LAYER keyword. The name identifies the well to which the layer belongs.

Layer_UBA

To identify the layer, enter the layer user block address immediately following the well name. The user block address is specified in the following general format: i1 j1 k1 / i2 j2 k2 / … please review the manual pages on the *PERF keyword for more information.

Do NOT encapsulate the user block address in quotes.

Layer_condition:

The valid list of quantities for layers is shown in table 3 below:

Table 3 : Layer quantities:

Subkeyword	Meaning
STO-R	Stock Tank Oil – Rate (sign tells whether Prod or Inj)
STW-R	Stock Tank Water – Rate
STG-R	Stock Tank Gas – Rate
STI-R	Stock Tank Intermediate Liquid – Rate
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
BHP	Layer bottom-hole pressure
DWN	Layer drawdown – absolute value of block P – layer P

*ON_SECTOR

This subkeyword identicates that the test condition is to be applied to a sector. GEM creates a default sector named 'FIELD' which includes all grid blocks and all wells. IMEX creates a default sector named ('Entire Field'). Please note that there are 2 spaces between the word 'Entire' and 'Field'.

Any other sector name used with a trigger must be previously defined.

sector_name

A single string representing a sector name (in quotes) of less than 16 characters to specify the sector to which this trigger applies. The sector must be previously defined. Only the 'FIELD' for GEM is created by default. The name must be on the same line as the *TRIGGER keyword.

Sector_condition:

The valid list of quantities for sectors is shown in table 4 below:

Table 4 : Sector quantities:

Subkeyword	Meaning
STO-RP	Stock Tank Oil – Rate of Production
PAVE	Pore-volume weighted pressure
SOAVE	Pore-volume weighted oil saturation
SWAVE	Pore-volume weighted water saturation
SGAVE	Pore-volume weighted gas saturation
STOIP	Stock tank oil in place
STWIP	Stock tank water in place
STFGIP	Stock tank free gas in place
STGIP	Stock tank gas in place, including free and dissolved
TEMPMAX	Maximum in temperature
TEMPMIN	Minimum in temperature

PMAX	Maximum in pressure
PMIN	Minimum in pressure
SOMAX	Maximum in oil saturation
SOMIN	Minimum in oil saturation
SWMAX	Maximum in water saturation
SWMIN	Minimum in water saturation
SGMAX	Maximum in gas saturation
SGMIN	Minimum in gas saturation

*ON_FIELD

This subkeyword identicates that the test condition is based on a field level quantity. Note: The string 'FIELD' in single quotes must immediately follow the *ON-FIELD keyword. The ON_FIELD keyword on its own adequately identifies a trigger on the entire field. However to maintain consistency in format with other keywords such as ON_WELL and ON_GROUP, the field name string has been retained for ON_FIELD as well. The user should enter the string 'FIELD' for the field name token even if the actual field level group or sector name assigned is different from 'FIELD'. If a string other than 'FIELD' is entered following ON_FIELD, a warning message will be generated and the simulation will proceed normally.

Field_condition:

The valid list of quantities for field is identical to that of groups as shown in table 2:

Operator:

The operator for the triggering condition must be one of:

'<' Less than

'>' Greater than

One of < or > (not in quotes) is required immediately following the specification of the test quantity using keywords such as *STO-RP

condition_value:

The value of the trigger condition. Enter a value based on the trigger condition and unit system selected for the simulation.

*ON_ELAPSED

This subkeyword identicates that the test condition is based on a time elapsed since the beginning of the simulation OR time elapsed after the time the trigger is defined (that is, when the trigger keyword lines are parsed by the reader). The trigger definition time could simply be the time specified by the DATE or TIME card. For an inner trigger of a nested trigger, it is the time that the immediate outer trigger condition is satisfied. Consider the following examples:

Example 1:

```
*TIME 10.0
trigger 'trig1' on_elapsed  'time' treltd > 4.99 apply_times 1
*TARGET *BHP
'P1'
200
end_trigger
```

In this case the trigger will be read in after exactly 10 days have elapsed – trigger definition time is therefore 10 days. The time specified for the trigger condition is relative to the time the trigger is created – therefore at the bottom of the first timestep that results in simulation time greater than 10+4.99 = 14.99 days the trigger condition will be met.

Example 2:

```
*TIME 10.0
trigger 'trig1' on_elapsed  'time' treltd > 4.99 apply_times 1
*TARGET *BHP
'P1'
200
trigger 'trig2' on_elapsed  'time' treltd > 5.99
apply_times 1
*TARGET *BHP
'P1'
100
trigger 'trig3' on_elapsed  'time' treltd > 20.99
apply_times 1
*TARGET *BHP
'P1'
50
end_trigger
end_trigger
end_trigger
```

In this case the outermost trigger named 'TRIG1' is defined at 10 days based on the time card entry. The inner trigger 'TRIG2' will be defined or comes into existence when the outer trigger 'TRIG1' condition is satisfied (sometime after 14.99 days – because the timestep sizes are not known apriori it is not possible to state exactly when this time will be). The inner 'TRIG3' will be defined when the outer trigger 'TRIG2' condition is satisfied, this can occur after at least 10+4.99+5.99 = 20.98 days have elapsed.

Note: The string 'TIME' in single quotes must immediately follow the *ON-ELAPSED keyword. If a string other than 'TIME' is entered following ON_ELAPSED, a warning message will be generated and the simulation will proceed normally.

Time_condition:

Either the time elapsed from the start of the simulation can be entered or the time elapsed relative to the time when the trigger is first defined can be specified. To specify the absolute time use the sub keyword *TIMSIM followed by a value. To specify elapsed time relative to the time when the trigger is parsed or defined use the sub keyword *TRELTD followed by a

value. The time should be in days for FIELD/SI/MODSI units and in minutes for laboratory (LAB) units.

TIMSIM:

Indicates that the time value entered is time elapsed from the start of the simulation, also referred to as absolute time.

TRELTD:

Indicates that the time value entered is relative to the time the trigger is defined, also referred to as relative time.

*ON_REPGRP

This subkeyword indicates that the test condition is to be applied to a reporting group. The reporting group must be previously defined to enable reporting group based triggers to be used. This token must immediately follow the trigger name string.

reporting_group_name

A reporting group name (in quotes) to specify the reporting group to which this trigger applies. The reporting groups must be previously defined. There are no groups created by default.

The reporting group name must immediately follow the *ON_REPGRP keyword. Required token.

rep_group_condition:

The valid list of quantities for reporting groups is shown in Table 5 below:

Table 5: Reporting Group quantities:

Subkeyword	Meaning
STO-RP	Stock Tank Oil – Rate of Production
STO-CP	Stock Tank Oil – Cumulative Production
STW-RP	Stock Tank Water – Rate of Production
STW-CP	Stock Tank Water – Cumulative Production
STW-RI	Stock Tank Water – Rate of Injection
STW-CI	Stock Tank Water – Cumulative Injection
STG-RP	Stock Tank Gas – Rate of Production
STG-CP	Stock Tank Gas – Cumulative Production
STG-RI	Stock Tank Gas – Rate of Injection
STG-CI	Stock Tank Gas – Cumulative Injection
STL-RP	Stock Tank Liquid – Rate of Production
STL-CP	Stock Tank Liquid – Cumulative Production
BHF-RP	Bottomhole Fluid – Rate of Production
BHF-CP	Bottomhole Fluid – Cumulative Production
BHF-RI	Bottomhole Fluid – Rate of Injection
BHF-CI	Bottomhole Fluid – Cumulative Injection
STI-RP	Stock Tank Intermediate liquid – Rate of Production
STI-CP	Stock Tank Intermediate liquid – Cumulative Production
WTG-RP	Wet Gas – Rate of Production

WTG-CP	Wet Gas – Cumulative Production
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
MPWS MXX	Mole percent of component XX in reporting group production
GWGR	Gas-Wet Gas Ratio (production)
WWGR	Water-Wet Gas Ratio (production)

Operator:

The operator for the triggering condition must be one of:

‘<’ Less than

‘>’ Greater than

One of < or > (not in quotes) is required immediately following the specification of the test quantity using keywords such as *TIMSIM

condition_value:

The value of the trigger condition. Enter a value based on the time elapsed condition selected in units of days for field/SI units.

*APPLY_TIMES:

Subkeyword used to specify the maximum number of times that the actions specified with the trigger can be taken. An integer number must immediately follow this subkeyword. This subkeyword is optional.

napt:

Enter a single integer representing the maximum number of times that the specified list of actions can be executed. If no value is entered, then the default is 1. With the default of 1 the trigger condition is tested at the end of every timestep. As soon as the trigger condition is satisfied the list of actions is implemented and the trigger is removed from the list of active triggers. If more than 1 (say “n” times) is selected then the trigger remains active until the trigger condition is satisfied (“n”) times.

*INCREMENT:

Subkeyword used to specify the increment to the trigger value. A single real number must follow this subkeyword. An integer number must immediately follow this subkeyword. This subkeyword is optional.

Trigger Increment:

Enter a single real value representing an increment to be applied to the trigger value, each time the trigger condition is satisfied. The trigger increment can be a negative number. The trigger increment does not need to be entered. The trigger increment can only be entered if the preceding number for the repetition times is also entered. The trigger increment is used only if a value greater than 1 is specified for the number of times the trigger condition can be satisfied. Once a trigger condition is satisfied the trigger value is recalculated as value_new =

existing_value + increment. The new value is then used in testing the trigger condition for subsequent times until the trigger condition is once again satisfied.

*AVRGTIME

Subkeyword used to specify the time period over which a moving average of a quantity such as rate is to be calculated. A single real number must follow this subkeyword. This subkeyword is optional.

Average time

Enter a single real value representing the time period over which a moving average of a quantity such as rate is to be calculated. This value must be greater than zero. For SI/FIELD/MODSI enter a value in units of days and in LAB units enter a value in units of minutes. If time elapsed since the trigger is defined is less than average time then the trigger condition will not be satisfied since a moving average can only be calculated once time elapsed is greater than the average time specified.

*TEST_TIMES:

Subkeyword used to specify the maximum number of times that the trigger can be tested to ascertain if the trigger condition is satisfied. A single integer number must follow this subkeyword. This subkeyword is optional.

ntestt:

Enter a single integer representing the maximum number of times that the trigger can be tested to ascertain if the trigger condition is satisfied. If no value is entered, then the default is to test the trigger every timestep. If a value of 1 is entered then the trigger condition is tested only once at the end of the timestep during which the trigger is defined. The trigger is then removed from the active trigger list whether or not the trigger condition itself is satisfied. If more than 1 (say “n” times) is selected then the trigger condition is tested for “n” timesteps after the trigger is defined.

*TEST_AFTER_TIMER:

Subkeyword used to specify the time delay which must elapse before the trigger condition will be tested. This time delay is relative to the time that the trigger comes into existence or is defined. A single real number must follow this subkeyword. This subkeyword is optional.

rtimedr:

Enter a single real number representing the delay in time (in days) after the time that the trigger is defined or parsed that the user wishes the trigger condition to be tested. If no value is entered, then the default is to assume a time delay of zero.

*TEST_AFTER_TIMEA:

Subkeyword used to specify the time delay in days which must elapse before the trigger condition will begin to be tested. This time delay is relative to the start of the simulation or absolute time elapsed. A single real number must follow this subkeyword. This subkeyword is optional.

rtimeda:

Enter a single real number representing the delay in days after the start of the simulation that the user wishes to elapse before the trigger condition is tested going forward. If no value is entered, then the default is to assume a time delay of zero.

{action_list}

The list of actions in the form of valid well and recurrent data keyword lines may be specified following the *TRIGGER keyword and its subkeywords. The action list must start on a new line. The action list is optional. Following the action lines specify the keyword *END_TRIGGER on a new line to signal the end of the trigger definition. It is okay to not specify any action lines within a trigger.

*END_TRIGGER

This keyword marks the end of the list of actions or keyword lines associated with a given trigger. It must be on a new line. The *TRIGGER and *END_TRIGGER must be used as a pair. For each *TRIGGER keyword, an *END_TRIGGER keyword is required.

*STO-RP

Oil production rate at surface conditions. For SI units enter a value in m3/d and for field units in STB/D.

*STO-R

Oil phase rate at surface conditions for a layer. A positive number implies fluid is flowing in a normal direction. Specifically flow is expected to be from reservoir to wellbore for a layer belonging to a producer and from wellbore to reservoir if the layer belongs to an injection well. On the other hand if a negative number is entered, then the user is testing for a back flowing layer. The magnitude of the number indicated the severity of the back flow. For SI units enter a value in m3/d and for field units in STB/D.

Example 1:

```
*TRIGGER ‘trig1’ *ON_LAYER ‘well1’ 1 1 1 / 2 2 2 *STO-R > 300.0
```

If well1 is an injector, then the condition will be satisfied if flow for the layer is from the well to the reservoir (injection) and the flow rate is greater than 300. However if well1 is a producer, then the condition will be satisfied if flow is from reservoir to wellbore (production) and the oil flow rate is greater the 300.0.

Example 2:

*TRIGGER ‘trig2’ *ON_LAYER ‘well1’ 1 1 1 / 2 2 2 *STO-R > -300.0

The condition will be satisfied if the layer is back flowing, for an injector flow is from the reservoir to the wellbore and the magnitude of the back flow rate is greater than 300.0. For a producer, the condition will be satisfied if flow is from wellbore to reservoir and the magnitude of the back flow rate is greater than 300.0. Therefore if the back flow rate is 350, then the condition is satisfied, if the back flow rate is 250 the condition is not satisfied.

*STW-R

Water phase rate at surface conditions for a layer. A positive number entered for a layer belonging to an injection well implies flow is expected to be from wellbore to reservoir, that is the well layer is injecting into the reservoir. A positive number entered for a layer belonging to a production well implies flow is from reservoir to wellbore. A negative number implies a back flowing layer. For SI units enter a value in m3/d and for field units in STB/D.

*STG-R

Gas phase rate at surface conditions for a layer. A positive number entered for a layer belonging to an injection well implies flow is expected to be from wellbore to reservoir, that is the well layer is injecting into the reservoir. A positive number entered for a layer belonging to a production well implies flow is from reservoir to wellbore. A negative number implies back flowing layer. For SI units enter a value in m3/d and for field units in STB/D.

*STI-R

Intermediate stream rate at surface conditions for a layer. A positive number entered for a layer belonging to an injection well implies flow is expected to be from wellbore to reservoir, that is the well layer is injecting into the reservoir. A positive number entered for a layer belonging to a production well implies flow is from reservoir to wellbore. A negative number implies back flowing layer. For SI units enter a value in m3/d and for field units in STB/D.

*STO-CP

Oil cumulative production at surface conditions. For SI units enter a value in m3 and for field units in STB.

*STW-RP

Water production rate at surface conditions. For SI units enter a value in m3/d and for field units in STB/D.

*STW-CP

Water cumulative production at surface conditions. For SI units enter a value in m3 and for field units in STB.

*STW-RI

Water injection rate at surface conditions. For SI units enter a value in m3/d and for field units in STB/D.

*STW-CI

Water cumulative injection at surface conditions. For SI units enter a value in m3 and for field units in STB.

*STG-RP

Gas production rate at surface conditions. For SI units enter a value in m3/d and for field units in scf/D.

*STG-CP

Gas cumulative production at surface conditions. For SI units enter a value in m3 and for field units in scf.

*STG_RI

Gas injection rate at surface conditions. For SI units enter a value in m3/d and for field units in scf/D.

*STG-CI

Gas cumulative injection at surface conditions. For SI units enter a value in m3 and for field units in scf.

*STI-RP

Intermediate liquid stream production rate at surface conditions. For SI units enter a value in m3/d and for field units in STB/D.

*STI-CP

Intermediate liquid stream cumulative production at surface conditions. For SI units enter a value in m3 and for field units in STB.

*WTG_RP

Wet gas stream production rate at surface conditions. For SI units enter a value in m3/d and for field units in scf/D.

*WTG-CP

Wet gas stream cumulative production at surface conditions. For SI units enter a value in m3 and for field units in scf.

*STL_RP

Liquid (oil + water) stream production rate at surface conditions. For SI units enter a value in m3/d and for field units in STB/D.

*STL_CP

Liquid (oil + water) stream cumulative production at surface conditions. For SI units enter a value in m3 and for field units in STB.

*BHF_RP

The oil plus water plus gas phase production rate at bottomhole or reservoir conditions. For SI units enter a value in reservoir m3/d and for field units in reservoir barrels BBL/D.

*BHF_CP

The oil plus water plus gas phase production cumulative at bottomhole or reservoir conditions. For SI units enter a value in reservoir m3 and for field units in reservoir barrels BBL.

*BHF-RI

The oil plus water plus gas injection rate at bottomhole or reservoir conditions. For SI units enter a value in reservoir m3/d and for field units in reservoir barrels BBL/D.

*BHF-CI

The oil plus water plus gas phase injection cumulative at bottomhole or reservoir conditions. For SI units enter a value in reservoir m3 and for field units in reservoir barrels BBL.

*GOR

Gas oil ratio at surface conditions. For SI units enter a value as surface m3/d of gas production per m3/d of surface oil production and for field units in scf of gas per STB of oil.

*WCUT

Water cut at surface conditions. The water cut is the ratio of surface production of water divided by the total surface liquid or water + oil production. For SI units enter a value as surface water production rate (m3/d) divided by the sum of surface production rates of water and oil (m3/d) and field units in STB/D of water per STB/D of water + oil production.

*WGR

Ratio of water production rate at surface conditions divided by the surface gas production rate. For SI units enter a value as surface water production rate (m3/d) divided by the surface production rates of gas (m3/d) and field units in STB/D of water per scf/D of gas production.

*GLR

Ratio of gas production rate at surface conditions divided by the surface liquid (sum of water + oil phases) production rate. For SI units enter a value as gas production rate (m3/d) divided by the sum of surface production rates of water and oil (m3/d) and field units in scf/D of gas per STB/D of water + oil production.

*MPWS *Mxx

Mole percent of component “m” in the well stream. Mole percent is calculated as the molar rate of component “m” flowing into the well divided by the total molar hydrocarbon flow rate multiplied by 100. The total rate is the sum of all hydrocarbon components (does not include water). For groups the calculation accounts for all producers attached to the group. Enter the component number after the letter M. For example for component number 5 in the components list (as determined by the order that components are specified with *COMPNAME keyword) use *MPWS *M5 <value>.

*MPVS *Mxx

Mole percent of component “m” in the surface gas stream. Mole percent is calculated as the molar rate of component “m” in the separator gas stream divided by the total molar hydrocarbon flow rate of the separator gas stream multiplied by 100. The total rate is the sum of all hydrocarbon components (does not include water). For groups the calculation accounts for all producers attached to the group. For example for component number 5 in the components list (as determined by the order that components are specified with *COMPNAME keyword) use *MPVS *M5 <value>.

*MPLS *Mxx

Mole percent of component “m” in the separator “liquid” stream. If there is no intermediate liquid stream, then the liquid stream equals the surface oil stream, otherwise the “liquid” stream includes the oil and the intermediate liquid streams. Mole percent is calculated as the molar rate of component

"m" in the separator liquid stream divided by the total molar flow rate of the separator liquid stream multiplied by 100. The total rate is the sum of all hydrocarbon components (does not include water). For groups the calculation accounts for all producers attached to the group. Enter the component number after the letter M. For example for component number 5 in the components list (as determined by the order that components are specified with *COMPNAME keyword) use *MPLS *M5 <value>.

*BHP

Bottomhole pressure of the well. Enter a value in kPa for SI units and psi for field units.

*WHP

Tubing head pressure of the well. Enter a value in kPa for SI units and psi for field units.

*WBHTEMP

Bottomhole temperature of the well. For SI units enter a value in deg C and for field units in deg F.

*WTHTEMP

Tubing head temperature of the well. For SI units enter a value in deg C and for field units in deg F.

*GWGR

Ratio of gas production rate at surface conditions divided by the wet gas production rate at surface conditions. For SI units enter a value as surface gas production rate (m3/d) divided by the surface production rate of wet gas (m3/d) and field units in scf/D of gas per scf/D of wet gas production.

*WWGR

Ratio of water production rate at surface conditions divided by the wet gas production rate at surface conditions. For SI units enter a value as water production rate (m3/d) divided by the wet gas surface production rate (m3/d) and field units in STB/D of water per scf/D of wet gas production.

*BKFLOW

If any layer of the well is back flowing then this condition is satisfied. Note: To specify the case where action(s) are to be taken if the well IS backflowing use the *BKFLOW keyword followed by the greater than operator and a value of 0. Alternatively to specify the case where the action(s) are to be taken if the well IS NOT back flowing use the *BKFLOW keyword followed by the less than operator and a value of 0. If an increment is specified with the *INCREMENT keyword, then that value will be ignored.

*BKFLOW > 0 take actions if the well IS BACK FLOWING

*BKFLOW < 0 take actions if the well IS NOT BACK FLOWING

*GLR

Ratio of gas production rate at surface conditions divided by the surface liquid (sum of water + oil phases) production rate. For SI units enter a value

as gas production rate (m3/d) divided by the sum of surface production rates of water and oil (m3/d) and in field units in scf/d of gas per stb/d of water plus oil production.

*RECYSTG

Minimum group gas recycling rate. Group injection rate is calculated by summing the injection rates at surface conditions of all cycling gas injectors that belong to the group. For SI units enter a value in m3/d and in field units in scf/d.

*RECYSTW

Minimum group water recycling rate. Group injection rate is calculated by summing the injection rates at surface conditions of all cycling water injectors that belong to the group. For SI units enter a value in m3/d and in field units in stb/d.

*DWN

Difference in absolute value well layer pressure minus the pressure of well block where well layer is completed. Enter a value in kPa for SI units and psi for field units.

*PAVE

Sector average pressure calculated on the basis of total pore volume. For SI units enter a value in kPa and for field units in psi.

*SOAVE

Sector average oil saturation calculated on the basis of total pore volume.

*SWAVE

Sector average water saturation calculated on the basis of total pore volume.

*SGAVE

Sector average gas saturation calculated on the basis of total pore volume.

*STOIP

Sector oil in place at standard conditions. For SI units enter a value in m3 and in field units in STB.

*STWIP

Sector water in place at standard conditions. For SI units enter a value in m3 and in field units in STB.

*STFGIP

Sector gas in place based on reservoir gas phase alone at standard conditions. For SI units enter a value in m^3 and in field units in scf.

*STGIP

Sector gas in place based on both reservoir gas and reservoir oil phase at standard conditions. For SI units enter a value in m3 and in field units in scf.

*PMAX

Sector maximum in pressure . For SI units enter a value in kPa and for field units in psi.

*PMIN

Sector minimum in pressure. For SI units enter a value in kPa and for field units in psi.

*TEMPMAX

Sector maximum in temperature. For SI units enter a value in deg C and for field units in deg F.

*TEMPMIN

Sector minimum in temperature. For SI units enter a value in deg C and for field units in deg F.

*SOMAX

Sector maximum in oil saturation.

*SOMIN

Sector minimum in oil saturation.

*SWMAX

Sector maximum in water saturation.

*SWMIN

Sector minimum in water saturation.

*SGMAX

Sector maximum in gas saturation.

*SGMIN

Sector minimum in gas saturation.

*GLIFTR

Gas lift rate. For SI units enter a value in m3/day and for field units in scf/day.

*GLIFTC

Gas lift cumulative volume. For SI units enter a value in m^3 and for field units in scf.

*GGLOR

Gas lift rate plus gas production rate divided by the oil production rate. For SI units enter a value in m^3/m^3 and for field units in scf/stb.

*GGLIFTR

Gas lift rate plus gas production rate. For SI units enter a value in m3/day and for field units in scf/day.

*GGLIFTC

Gas lift cumulative plus gas produced cumulative. For SI units enter a value in m^3 and for field units in scf.

*GGLLIQR

Gas lift rate plus gas production rate divided by the liquid (oil+water) production rate. For SI units enter a value in m3/m3 and for field units in scf/stb.

*GLOILR

Gas lift rate divided by the oil production rate. For SI units enter a value in m^3/m^3 and for field units in scf/stb.

*GLLIQR

Gas lift rate divided by the liquid (oil + water) production rate. For SI units enter a value in m^3/m^3 and for field units in scf/stb.

*TIMSIM

Condition based on time elapsed since the beginning of the simulation in days.

*TRELTD

Condition based on time elapsed relative to the time when the trigger is defined, in days.

DEFAULTS:

*TRIGGER is an optional keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. Note that when surface network linked runs are performed, the action associated with a trigger condition that has been satisfied will not be performed until the next surface network timestep.

EXPLANATION:

This keyword allows for certain actions to be implemented when a specified condition or trigger is satisfied. The actions are specified in the form of a block of valid keywords encapsulated within the *TRIGGER and *END_TRIGGER keywords. With the exception of the following keywords: *DATE, *TIME, and any other valid keywords that can otherwise be used in the WELL and RECURRENT data section of a GEM input data file can be used with triggers.

A TRIGGER keyword is followed by one or more conditions. A pair of conditions is linked either by an *AND or an *OR logical operator. Each logical operator must have an integer number associated with it, to be specified by the user. This integer assigns a priority ranking to the operator. *AND/*OR logical connectors enable multiple conditions to be specified for a single trigger. An expression with the AND logical construct between two trigger conditions requires both conditions to be met or satisfied for the expression to be evaluated as true. An expression with the OR logical construct between two trigger conditions is evaluated as true if either conditions is met or satisfied. An integer following the AND/OR keyword specifies the priority ranking of the operator. The priority ranking determines the order (sequence) used to simplify a compound logical statement with multiple conditions successively until the entire expression is evaluated as either true or false. This is best illustrated with examples shown below. An operator with the integer value of 1 is parsed first, followed by the operator with the integer designation of 2, etc. The priority ranking only matters if there is more than 1 operator is specified and both types of operators are specified. If only the OR operator is specified then the entire expression is true if any one of the conditions or clauses is evaluated as true. If only the AND operator is specified then all conditions or clauses must be satisfied for the overall expression to be true. If the overall logical expression involving multiple conditions and the associated logical operators is evaluated as TRUE then the trigger condition(s) are met and the list of actions specified will be taken. In general therefore whether a compound logical

expression is evaluated as TRUE or FALSE at a given time depends on the following 3 factors: 1) Whether a given single condition is evaluated as true or false 2) The type of logical operator connecting two conditions and 3) the priority sequence attached to the operator. The priority assigned to a logical operator will be material if more than two conditions are specified. This is illustrated with a number of examples. The simplest case of a trigger statement involves a single condition. For example:

```
*TRIGGER ‘trig1’ *ON-GROUP ‘grp-1’ *STO-RP > 100
        *SHUTIN ‘well1’
*END_TRIGGER
```

In this case if the oil production rate of group named ‘grp-1’ is more 100 at the end of particular timestep then the logical expression is TRUE and the specified actions will be taken at the beginning of the next timestep; specifically well named ‘well1’ will be shut.

If exactly two conditions are involved then one logical operator is required. This can be either an *AND or an *OR. Since there is only one operator an integer for the priority sequence is not required. Examples are:

```
*TRIGGER ‘trig1’ *ON-GROUP ‘grp-1’ *STO-RP > 100 *AND *ON_WELL
‘well1’ STG-RP > 5.0E+6
        *SHUTIN ‘well1’
*END_TRIGGER
```

In this case both conditions must be TRUE for the overall logical expression to be TRUE as well. That is group production of ‘grp-1’ must be greater than 100 and the gas rate of ‘well1’ must be greater than 5.0e+6.

If an OR operator links two conditions than if either of the conditions is TRUE then the overall expression is also true. Using the above example

```
*TRIGGER ‘trig1’ *ON-GROUP ‘grp-1’ *STO-RP > 100 *OR *ON_WELL
‘well1’ STG-RP > 5.0E+6
        *SHUTIN ‘well1’
*END_TRIGGER
```

If either the ‘grp-1’ oil rate is greater than 100 or the gas rate of ‘well1’ is greater than 5.0E+6 the overall expression is evaluated as true and well ‘well1’ is shut in.

For cases where a trigger logical expression consists of three or more conditions, the statement is successively simplified by replacing a pair of conditions and associated operator with either a false or true result until the entire expression can be evaluated to either true or false. The order in which the logical operators are parsed (or evaluated) will influence how the overall logical expression is evaluated. Consider the following two cases:

Case 1:

```
*TRIGGER ‘trig1’ condition1 *AND1 condition2 OR2 condition3
      *SHUTIN ‘well1’
*END_TRIGGER
```

Case 2:

```
*TRIGGER ‘trig1’ condition1 *AND2 condition2 OR1 condition3
      *SHUTIN ‘well1’
*END_TRIGGER
```

Where condition1, condition2 and condition3 are trigger conditions such as *ON_GROUP ‘grp-1’ *STO-RP > 100. For case 1 the logical expression is TRUE if BOTH conditions 1

and 2 are true OR condition 3 is true. An alternative way of specifying the order would have been to use brackets to isolate the operator that must be evaluated first

```
*TRIGGER ‘trig1’ {condition1 *AND condition2} OR condition3
      *SHUTIN ‘well1’
*END_TRIGGER
```

Please note GEM DOES NOT support the use of brackets to specify logical operator order – only the integer number following *AND/*OR.

For case 2 the overall expression is TRUE if either condition2 OR condition3 is true AND condition 1 is true or equivalently

```
*TRIGGER ‘trig1’ condition1 *AND {condition2 OR condition3}
      *SHUTIN ‘well1’
*END_TRIGGER
```

As a further example, consider the following trigger:

```
*TRIGGER 'trig7' *ON_GROUP 'Grp-1' 'Field' sto-rp > 193.0 *OR
*ON_GROUP 'Field'  gor > 1640.0   *AND *ON_GROUP 'Grp-2' sto-rp >
164.0
```

Condition #1 is *ON_GROUP 'Grp-1' 'Field' sto-rp > 193.0

Condition #2 is *ON_GROUP 'Field' gor > 1640.0

Condition #3 is *ON_GROUP 'Grp-2' sto-rp > 164.0

And suppose at a particular time Field oil rate = 200, Field gor = 1500 and Grp-2 oil rate = 100. Consequently condition 1 will be evaluated as true, condition 2 will be false and condition 3 will be false. Now if the order of the logical connectors is OR first then AND, then the compound trigger will be NOT be satisfied. The expression *ON_GROUP 'Grp-1' 'Field' sto-rp > 193.0 *OR *ON_GROUP 'Field' gor > 1640.0 is true since the first condition is true. However on evaluating the subsequent conditions associated with the AND operator, the result is false since Grp-2 oil rate is less than 164. If the order is reversed and the AND operator is dealt with first, the result is false, however upon evaluating the resulting conditions with OR operator the result is true since the first condition, field oil rate is greater than 193.

The trigger condition of each active trigger is tested at the bottom (end) of the timestep (that is after the timestep is considered to be completed, the cumulatives have been updated and the simulation is ready to proceed to the next timestep). If the trigger condition is satisfied then the list of associated actions are processed at the top (beginning) of the next timestep. The order of the triggers in the trigger list depends entirely on the sequence on which the triggers are defined. The triggers are not sorted in any other way. If nesting is used (that is a trigger is defined within another trigger) then the inner trigger is defined or becomes active only if the outer trigger condition is satisfied – such nesting may impact on the order that triggers appear in the trigger list.

Once a trigger condition is met, the count of the number of times remaining that the trigger can be satisfied is reduced by 1. If for example the default value of 1 is used for the maximum number of times the trigger can be satisfied, then the count remaining is 0 and the trigger is removed from the list of active triggers. The trigger is also removed from the active trigger list once the trigger condition has been tested for the maximum number of times as specified with the *TEST_TIMES keyword.

If a well change time (as specified with *DATE or *TIME card) coincides with the time that the trigger actions are to be processed, then the trigger actions are implemented FIRST followed by the list of actions or keywords following the *DATE or *TIME card.

It is possible to nest triggers by including the definition of the ‘inner’ trigger within the keyword set of the ‘outer’ trigger (that is before specification of the *END_TRIGGER keyword of the ‘outer’ trigger). This offers great flexibility, however such nesting should be properly thought out to avoid unintended consequences. For example it is possible to define a trigger ‘recursively’ using nesting. That is the inner and outer triggers can be assigned the same name. In this case when the outer trigger condition is first satisfied, the trigger redefines itself based on the inner trigger definition. For example consider the following block of keywords in the well and recurrent data section of a GEM data file

```
*TIME 0.0
…
…
…
*WELL ‘well1’
    *OPERATE *STO  750.0
    *OPERATE *MIN *BHP 3000.0
*WELL ‘well2’
    *OPERATE *STO 250.0
    *OPERATE *MIN *BHP 3000.0
*WELL ‘well3’
    *OPERATE **STO 250.0
    *OPERATE *MIN *BHP 3000.0

*SHUTIN ‘well2’ ‘well3’
*TIME 50.0  **Time when Trigger is defined
*TRIGGER ‘trig1’ *ON_WELL ‘well1’ *STO-RP < 500.0 *APPLY_TIMES 3
  *INCREMENT -25.0        ** outer trigger
     *WTMULT  *STO ‘well2’
        1.25
     *TRIGGER ‘trig1’ *ON_WELL ‘well1’ *GOR > 4500.0
                             **inner trigger, same name
              *OPEN ‘well3’
      *END_TRIGGER
*END_TRIGGER
```

At time equal to 50 days, the outer trigger is defined. At the end of the next and subsequent timesteps the trigger condition is tested until it is satisfied or the simulation stops. Suppose at time equal to 65 days the oil production rate for well ‘well1’ drops to below 500 STB/D. The trigger condition is satisfied, well 2 is opened with an initial oil rate target of 250 *1.25 or 312.50. However because the inner trigger has the same name as the outer trigger, the trigger ‘trig1’ is also redefined at this time according to the inner definition. The maximum repeat count of 3 as well as the increment to the trigger condition of -25 STB/D specified with the outer trigger definition is redundant and can never be used. After 65 days the new trigger condition is based on GOR and if satisfied will open well 3 with an initial target of 250 STB/D.

The list of actions (keywords) appearing within a trigger definition are not validated in any way (that is checked for syntax or consistency with previous well and recurrent data etc.) until the trigger condition is satisfied. In *CHECKONLY mode however each line of the well and recurrent data including keywords (actions) within the trigger definition are processed.

With the implementation of trigger actions GEM restart runs are now handled differently. With previous versions of GEM, well management information required for a restart run was synthesized from both data saved to a restart file and from parsing the recurrent data to the last date/ time card prior to restart time. With the new version of GEM all information required for restart is written and read from the restart file. No recurrent data is read prior to restart time. Therefore whereas it was previously possible to have certain keywords (actions) take effect immediately upon restart by changing the recurrent data prior to the restart time, this is no longer possible.

The '@' symbol in quotes may be used as a place holder for a list of wells or groups that have satisfied the trigger condition. This list will be a subset of the list of wells or groups specified as part of the trigger definition. For example consider the following trigger:

```
*TIME 50.0
*TRIGGER 'trig1' *ON_WELL 'pr*' *GOR > 5000
       *SHUTIN '@'
*END TRIGGER
```

Suppose production wells named pr1, pr2, pr3, pr4 that match the wild card well list pr* exist prior to the time that trigger 'trig1' is defined, that is at 50 days. Then when 50 days of simulation time has elapsed the GOR of each well pr1, pr2, pr3 and pr4 respectively will be tested against the trigger value of 5000. Suppose that at some subsequent time, say 70 days, wells pr1 and pr3 have GOR > 5000. In that case the trigger condition is satisfied and wells pr1 and pr3 will be shut in as directed, that is the *SHUTIN '@' statement is equivalent to the user having specified *SHUTIN 'pr1' 'pr3'. **NOTE: When the '@' place holder is used ALL keyword(s) that PRECEDE the place holder must on the same line as the place holder. Therefore for example

```
*TARGET *STO '@'
1.5
 is valid, but
*TARGET *STO
'@'
1.5
or
*TARGET
*STO '@'
1.5
```

are not, since in the latter cases TARGET and STO and '@' are not on the same line. Because of the special significance of the '@' symbol, please do not use this symbol as part of a well, group, sector or trigger name.

It is possible to link a value specified within an action line (well/group operating/monitoring condition) with a value related to a trigger condition. As an example consider the following blocks of keywords:

```
*TRIGGER 'trig1' *ON_WELL 'pr1' *GOR > 5000
                *ALTER 'pr1'
                     '$/10'
*END TRIGGER
```

If at a particular time the GOR for well reaches a value greater than 5000, say 5525 then the trigger condition is satisfied, the expression '$/10' becomes 5525/10 or 552.5 and the primary constraint for well 'pr1' will be set equal to 552.5.

If the ‘#’ symbol is used instead of the ‘$’ symbol, then the value being tested AGAINST is substituted for ‘#’ symbol. So specifying the same trigger, that is,

```
*TRIGGER ‘trig1’ *ON_WELL ‘pr1’ *GOR > 5000
  *ALTER ‘pr1’
     ‘#/10’
*END TRIGGER
```

And assuming at a particular time the GOR of well ‘pr1’ reaches a value greater than 5000, say 5525, then the trigger condition is evaluated as true and the following action lines are parsed:

```
*ALTER ‘pr1’
  ‘500’

The value of 500 is calculated by substituting 5000 for ‘#’ symbol.
Only ONE of the operators *,/,+,- can be used within action line and
can  only be used ONCE. Therefore ‘$*15.5’ is OK, whereas ‘$*5.0 +
3.0’ is not allowed and neither is ‘$*5.0*3.0’
```

Alter Well Constraint Value via a Multiplier or an Increment (Optional) *WTMULT, *WTINCR

PURPOSE:

*WTMULT or *WTINCR allows modification of any previously specified well constraint value for well(s) listed by well_numbers or 'well_names' by applying a multiplying factor or an increment.

FORMAT:

*WTMULT or *WTINCR	*constraint_type*	(*well_numbers*) or ('*well_names*') or '@' place holder	*value_list*

DEFINITIONS:

constraint_type

A constraint type that is valid for *OPERATE for wells in the well list. See *OPERATE for the full list and explanations of constraint types.

well_numbers

Any number of integers, or a range of integers to specify the well numbers to which this target applies. These well numbers must be on the same line as the *WTMULT/*WTINCR keyword. If more than one line is required to specify all the wells, then the keyword must be repeated.

well_names

Any number of 'well_names' in quotes to specify the wells to which this target applies. These names must be on the same line as the *WTMULT/*WTINCR keyword. If more than one line is required for the well list, then the keyword must be repeated. Limited wild-carding is available for this list; please see the explanation on the manual page for the *SHUTIN keyword.

'@'

When the '@' symbol in quotes is specified in place of well numbers or well list, then it is considered to be a place holder for the list of wells that have satisfied the trigger condition. This list is necessarily a subset of the well list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. All keywords that precede the '@' place holder must appear on the same line as the '@' place holder. Therefore *WTMULT/*WTINCR and a keyword specifying a stream, such as STO must be on the same line as the '@' place holder. See the EXPLANATION section of *TRIGGER keyword for examples of the use of @ place holder.

value_list

One number for each well identified by *well_numbers* or '*well_names*' specifying a multiplying factor or an increment to be applied. The new value of the specified constraint = old value * multiplier for *WTMULT or = old value + increment for *WTINCR. *value_list* must appear on one or more new lines immediately following the keyword line. *value_list* must NOT appear on the same line as the well list. A single value can be applied to all listed wells. Values for the well cutback constraints refer to the base values.

DEFAULTS:

Optional keywords. No defaults.

CONDITIONS:

*WTMULT/*WTINCR must be located in the WELL AND RECURRENT DATA keyword group, and may appear anywhere in this keyword group following the initial *OPERATE declaration.

EXPLANATION:

This optional keyword is used to alter a constraint value for wells without having to redefine all of the additional operating constraints. It is an effective method of altering constraints when performing a history match.

If a wells constraint is subsequently changed (by the use of an *ALTER keyword for example), *WTMULT/*WTINCR is not reapplied automatically, and should be redefined if required.

*WTMULT/*WTINCR also opens a well if the well has been shut in by a previous action or if the well has been initially defined as a shut in well. When the keyword is encountered in a data set, the simulator checks if the altered constraint with the new value becomes the most restrictive well constraint. If so, the well is switched to this new constraint type. If not, the new value is entered for the constraint but the well is switched to (or continues to run on) the currently most restrictive constraint.

The constraint type must be valid for *OPERATE for wells in the well list.

The *WTMULT/*WTINCR keyword should NOT be used to define a new operating constraint. The multiplier/increment specified with this keyword will ONLY be applied if the constraint of the type specified with the keyword already exists for the well based on earlier input using the *OPERATE keyword.

Example 1:

```
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
:
*WTMULT *STO 1 ** This alters the target constraint of *STO
1.5            ** based on the *OPERATE keyword above, i.e.
               ** from 500 to 500 *1.5 = 750
*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
```

```
*WELL 3 'Producer 3'
*WELL 4 'Injector 1'
:
```

Example 2:

```
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
:
*WTINCR *STO 1 ** This alters the target constraint of *STO
150.            ** based on the *OPERATE keyword above, i.e.
                ** from 500 to 500 +150 = 650
```

The *WTMULT keyword may also look like this when several wells have been defined:

```
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 2
*OPERATE *MAX *STO 750.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 3
*OPERATE *MIN *BHP 2500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*INJECTOR 4
*OPERATE *MAX *STW 100.0
*MONITOR *MIN *STW 10.00 *SHUTIN
:
*TIME 1200.
** At a later date, want to adjust the target
** constraint values.

**           well_numbers
*WTMULT *STO 1:2
** multipliers
   2*1.25
**           well_number
*WTMULT *BHP 3
     ** multiplier ** BHP_new = 2500*0.25 = 500
    0.25
**           well_number
*WTMULT *STW 4
** values
     0.50
```

The *WTINCR keyword may also look like this when several wells have been defined:

```
*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
*WELL 3 'Producer 3'
*WELL 4 'Injector 1'
:
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 2
*OPERATE *MAX *STO 750.0
```

```
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 3
*OPERATE *MIN *BHP 2500.0

*MONITOR *MIN *STO 10.00 *SHUTIN

*INJECTOR 4
*OPERATE *MAX *STW 100.0
*MONITOR *MIN *STW 10.00 *SHUTIN
:
*TIME 1200.
** At a later date, want to adjust the target
** constraint values.

**           well_numbers
*WTINCR *STO 1:2
** increments
   2*125

**           well_number
*WTINCR *BHP 3
     ** increment ** BHP_new = 2500-2000 = 500
    -2000.0

**           well_number
*WTINCR *STW 4
** values
     -5.0
```

Group Production Constraints Multiplier (Optional) *GCONPMULT, *GCONPINCR

PURPOSE:

*GCONPMULT /*GCONPINCR is used to modify existing group production target controls with the use of multipliers or increments.

FORMAT:

*GCONPMULT	'group_names'
or	or
*GCONPINCR	'@'
constraint_type *value*	

DEFINITIONS:

'group_names'
: Are the groups to which the following constraint multiplier/increment applies.

constraint_type
: A production constraint type that is valid for *GCONP for groups in the group list. See *GCONP for the full list and explanations of constraint types.

'@'
: When the '@' symbol in quotes is specified in place of group names, then it is considered to be a place holder for the list of groups that have satisfied the trigger condition. This list is necessarily a subset of the group list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. The keyword *GCONPMULT/*GCONPINCR and the '@' place holder must appear on the same line. See the EXPLANATION section of *TRIGGER keyword for examples of the use of @ place holder.

value
: Multiplier/increment to be applied to the existing constraint value – see *GCONP for units of the increments. Constraint value for *PMAINT refers to the sector pressure target defined by *PMTARG.

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group together with the target constraint for the particular stream or control must be defined using the *GCONP 'grp_name' *GTARGET keyword before the target constraint multiplier/increment can be applied. Do not use the *GCONPMULT/*GCONPINCR keyword to define new group target constraints.

EXPLANATION:

*GCONPMULT/*GCONPINCR is used to modify group production constraints which have previously been defined using the *GCONP ‘grp_name’ *GTARGET keyword.

Example 1:

```
*GCONPMULT  ‘Field’  *STW  0.75
```

This directs the specified ‘Field’ group stock tank water production to be modified such that the new constraint value is 75% of the value specified with the last *GCONP ‘Field’ *GTARGET *STW card.

Example 2:

```
*GCONPMULT ‘Group-1’
*MNP 1.25
```

This sets a manifold pressure target of 125 % of the value specified with the last *GCONP ‘Group-1’ *GTARGET *MNP card. This group must have had its production assigned manifold treatment with the *MANIFOLD keyword. ‘Group-1’ must have wells directly attached to it.

Example 3:

```
*GCONPINCR  ‘Field’  *STW  -300.00
```

This directs the specified ‘Field’ group stock tank water production to be modified such that the new constraint value is equal to the value specified with the last *GCONP ‘Field’ *GTARGET *STW card less 300.

Group Injection Constraints Multipliers (Optional) *GCONIMULT, *GCONIINCR

PURPOSE:

*GCONIMULT/*GCONIINCR is used to specify multipliers/increments which modify existing group injection target controls.

FORMAT:

*GCONIMULT	'group_names'
or	or
*GCONIINCR	'@'
constraint_type	*value*

DEFINITIONS:

'group_names'
: Are the groups to which the following constraint multiplier/increment applies.

constraint_type
: A injection constraint type that is valid for *GCONI for groups in the group list. See *GCONI for the full list and explanations of constraint types.

'@'
: When the '@' symbol in quotes is specified in place of group names, then it is considered to be a place holder for the list of groups that have satisfied the trigger condition. This list is necessarily a subset of the group list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. The keyword *GCONIMULT/*GCONIINCR and the '@' place holder must appear on the same line. See the EXPLANATION section of *TRIGGER keyword for examples of the use of @ place holder.

value
: Constraint multiplier/increment value. Constraint value for *PMAINT refers to the sector pressure target defined by *PMTARG.

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group must be defined, by appearing in the list directly following *GROUP in a *GROUP line or after the *ATTACHTO keyword on a *GROUP line. The group target constraint for the particular constraint type multiplier/increment being specified with *GCONIMULT/ *GCONIINCR must be also be previously defined using the keyword *GCONI.

EXPLANATION:

*GCONIMULT/*GCONIINCR is used to specify multipliers/increments which can be used to modify existing injection group constraint targets. It can also be used to modify previously specified voidage replacement targets.

Examples:

```
*GCONIMULT ‘Group-1’ ‘Group-2’
*STG 3.0
```

This resets stock tank gas injection targets to 300 % of the previously specified values for Group-1 and Group-2.

```
*GCONIINCR ‘Group-1’ ‘Group-2’
*STG -25000.00
```

This resets stock tank gas injection target to the previously specified values for Group-1 and Group-2 less 25,000.00.

The stock tank gas injection target must be previously specified with a data line such as:

```
*GCONI ‘Group-1’ ‘Group-2’ *GTARGET *STG 5.555E+07
```

Example: This is an example using voidage replacement. The water voidage replacement fraction is reduced by half whereas the gas voidage replacement fraction is increased by a factor of 2.

```
*GCONIMULT 'Group1'
  *VREP *WATER 0.5

*GCONIMULT 'Group1'
  *VREP *GAS     2.0
```

The voidage replacement targets must be previously specified with data lines such as:

```
*GCONI ‘Group-1’ ‘Group-2’ *VREP *WATER 0.6

*GCONI ‘Group-1’ ‘Group-2’ *VREP *GAS 0.4
```

Allow a Cycling Group to be Defined (Optional)

***GCONCYCLE_START, *GCONCYCLE_END**

PURPOSE:

*GCONCYCLE_START allows the user to specify data for EOR/IOR processes which rely on alternating production and/or injection cycles. Generally multiple cycles are required and each cycle consists of a number of well defined parts. Examples include water alternating gas injection (WAG) or cyclic steam injection (SSI). A number of criteria are available to stipulate the duration of each cycle part.

FORMAT:

*GCONCYCLE_START ‘group_names’
*CYCSTREAMS ‘stream_name_list’ (total of nstreams) – optional
*NPARTS nparts – optional
*TARGETTYPES (# of streams must be equal to nstreams) – optional
‘stream_name1’ ‘list_of_target_types’ (# of values must be equal to nparts)
‘stream_name2’ ‘list_of_target_types’ (# of values must be equal to nparts)
*MAXRATES -- conditional
‘stream_name1’ maximum_rate, imposed as a group target (# of values must be equal to nparts)
‘stream_name2’ maximum_rate, imposed as a group target (# of values must be equal to nparts)
*VREPFR -- conditional
‘stream_name1’ voidage replacement fraction, imposed as a group target (# of values must be equal to nparts)
‘stream_name2’ voidage replacement fraction, imposed as a group target (# of values must be equal to nparts)
*RECYFR -- conditional
‘stream_name1’ recycling fraction, imposed as a group target (# of values must be equal to nparts)
‘stream_name2’ recycling fraction, imposed as a group target (# of values must be equal to nparts)
*MAXCUMS -- optional
‘stream_name1’ maximum cumulative injection/production (# of values must be equal to nparts)
‘stream_name2’ maximum cumulative injection/production (# of values must be equal to nparts)
*TOTHEAT – optional, can be specified for injection streams ONLY, for STARS
stream_name1 maximum cumulative enthalpy injection (# of values must be equal to nparts)
stream_name2 maximum cumulative enthalpy injection (# of values must be equal to nparts)
*MAXBHP – optional, can be specified for injection streams ONLY
stream_name1 maximum BHP (# of values must be equal to nparts)
stream_name2 maximum BHP (# of values must be equal to nparts)

*MAXTIMES ntime1 ntime2 (# of values must be equal to nparts) –optional
*DTWCYC dtime1 dtime2 (# of values must be equal to nparts) --optional
*NCYCLES ncycles –optional
*MINBHP bhp1 bhp2 (# of values must be equal to nparts) –optional
*MINQOIL minoil1 minoil2 (# of values must be equal to nparts) --optional
*DEPNDX depndx1 depndx2 ((# of values must be equal to nparts) --optional
*GCONCYCLE_END --required

DEFINITIONS:

*GCONCYCLE_START

This keyword indicates the start of keyword data input for group cycling control. The complementary keyword *GCONCYCLE_END signals the end of group cycling keyword data input.

*CYCSTREAMS

This keyword is used to specify the list of streams that participate in the cycling process. A stream name can be one of the following: Production stream, or 'PROD', or injection streams, gas 'GASI', water 'WATI' or for the IMEX simulator only solvent 'SOLI'. For each stream a target type must be entered for every cycle part using the keyword *TARGETTYPES as well as a maximum rate for every cycle part using the keyword *MAXRATES or a voidage replacement fraction (for injection streams) using the keyword *VREPFR or a cycling fraction using keyword *RECYFR. The target type and maximum rate (or voidage replacement fraction or recycling fraction) provide sufficient information to impose a group target rate for each cycle part for each stream. This keyword is optional, the defaults are 'WATI', and 'GASI', corresponding to a case where a WAG (water alternating gas injection) process is being modeled. For simulating any other process such as cycling steam etc, this keyword is necessary and is a prerequisite before any other keyword other than *NPARTS can be specified.

*NPARTS

This keyword is used to specify the number of cycle parts required to fully define one complete cycle. A particular cycle part will generally be based on a period where there is either production or injection of a particular stream or a period where there is neither production nor injection (soak). However the user can choose to specify simultaneous injection of one or more streams with production. This keyword is optional, the internal defaults correspond to the case where a WAG (water alternating gas injection) process is being modeled. Therefore the default is a value of 2. To simulate any other process such as cyclic steam stimulation, this keyword is required and is a prerequisite before any other keyword with the exception of *CYCSTREAMS can be specified.

*TARGETTYPES

This keyword is used to specify the particular phase associated with a given stream and whether the entered maximum rates (with keyword *MAXRATES) are to be interpreted as values at surface or reservoir conditions. The table below shows the available selections for the target types for each stream. This

keyword is optional with the internal defaults corresponding to the case where a WAG (or water alternating gas injection) process is being modeled. To simulate any other process, this keyword is required.

The internal defaults for a WAG process are equivalent to specifying the following:

*TARGETTYPES

‘WATI’ ‘STW’ ‘STW’

‘GASI’ ‘STG’ ‘STG’

STREAM	IMEX	GEM	STARS
Production (‘PROD’)	Stock tank oil (‘STO’)	Stock tank oil (‘STO’)	Stock tank oil (‘STO’)
	Stock tank gas (‘STG’)	Stock tank gas (‘STG’)	Stock tank gas (‘STG’)
	Stock tank water (‘STW’)	Stock tank water (‘STW’)	Stock tank water (‘STW’)
	Bottomhole fluid (‘BHF’) – includes oil, gas and water phase production at reservoir or bottomhole conditions as well as production of the 4th phase, if defined such as solvent or polymer	Bottomhole fluid (‘BHF’) – includes oil, gas and water phase production at reservoir or bottomhole conditions	Bottomhole fluid (‘BHF’) – includes oil, gas and water phase production at reservoir or bottomhole conditions
	Stock tank liquid (‘STL’) – includes production of oil and water phases at surface or stock tank conditions	Stock tank liquid (‘STL’) – includes production of oil and water phases at surface or stock tank conditions as well as the production of the intermediate liquid phase if defined	Stock tank liquid (‘STL’) – includes production of oil and water phases at surface or stock tank conditions
		Intermediate liquid phase (‘STI’) – the intermediate liquid stream has to be defined first as part of the surface separation or separator data	
		Wet gas (‘WTG’) -	
Water injection (‘WATI’)	Stock tank water (‘STW’) or bottomhole water (‘BHW’)	Stock tank water (‘STW’) or bottomhole water (‘BHW’)	Stock tank water (‘STW’) or bottomhole water (‘BHW’)
Gas injection (‘GASI’)	Stock tank gas (‘STG’) or bottomhole gas (‘BHG’)	Stock tank gas (‘STG’) or bottomhole gas (‘BHG’)	Stock tank gas (‘STG’) or bottomhole gas (‘BHG’)
Solvent injection (‘SOLI’)	Stock tank solvent (‘STS’) or bottomhole solvent (‘BHS’)		

*MAXRATES

This keyword is used to specify the maximum rate for each stream and for each cycle part. The values entered will be interpreted based on the target types specified using the keyword *TARGETTYPES. The specified maximum rates are then imposed as group targets. If a voidage replacement group target needs to be imposed, then enter a value less than or equal to zero for maximum rate for that stream and a value between 0.0 and 1.0 for the voidage replacement fraction for the stream using the keyword *VREPFR. If a recycling fraction group target needs to be imposed, then enter a value less than or equal to zero value for maximum rate for that stream and a value between 0.0 and 1.0 for the recycling fraction for the stream using the keyword *RECYFR. For example to specify a voidage replacement fraction for the water injection stream and a maximum rate group target for the gas injection stream:

```
*cycstreams 'WATI' ‘GASI’
*maxrates
'GASI' 0.   250.

*vrepfr
'WATI' 1.0 0.0
```

A zero or negative number specified with *MAXRATES indicates that the group target will be based on voidage replacement fraction or a recycling fraction. Accordingly a voidage replacement fraction needs to be specified using the keyword *VREPFR for that stream or a recycling fraction specified with *RECYFR. Similarly a zero or negative value specified with *VREPFR indicates that a value for maximum rate to be imposed as a group target is provided with the keyword *MAXRATES for that stream or by keyword *RECYFR for that stream. If maximum rate group targets are to be imposed for all streams then the *VREPFR or *RECYFR keyword is not required. Similarly if voidage replacement fraction is to be specified for all streams then the *MAXRATES keyword is not required. For the *MAXRATES or *VREPFR or *RECYFR keyword, data for a given stream must be specified for all cycle parts as shown in the above example.

Please note that group cycling control imposed targets will in general override any previously specified group targets through data. If, at any time group cycling control is turned off, any group targets that existed prior to imposition of group cycling control are not as rule re-applied. However the user can choose to re-specify any such targets using a special trigger at the termination of group cycling control.

One of the *MAXRATES and/or *VREPFR and/or *RECYFR is required. For example consider the following fragment of user data:

```
cycstreams 'PROD' 'WATI'
nparts 3
targettypes
'WATI' 'STW' 'STW' 'STW'
'PROD' 'STL' 'STL' 'STL'
maxrates
'WATI' 1000. 0. 0.
'PROD' 0.   0. 1000.0
```

In this case cycle part 1 involves water injection, followed by a soak period for cycle part 2 and then production for cycle part 3. For cycle part 1, the group cycling manager will impose a target of 0.0 for STL for the production stream and group injection target of 1000 units of STW for the water stream. For cycle part 2, group production target will be STL value of 0.0 and group injection target for the water stream will be 0.0 STW. For cycle part 3, the group production target will be 1000 units of STL and the group injection target will be 0.0 STW.

INPUT UNIT SYSTEM / TARGET TYPE	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
STO, STW, STL, STI, BHF, BHW	M3/day	Bbl/day	Cm^3/min
WTG, STG, BHG, STS, BHS	M3/day	Ft^3/day	Cm^3/min

*VREPFR

This keyword is used to specify the voidage replacement fraction in lieu of a maximum rate for an injection stream (WATI, GASI), for each cycle part or recycling fraction specified with keyword *RECYFR. The group control target for that stream will be imposed such that the specified fraction of total produced volume measured at reservoir condition is replaced by injection. For a given stream, specify either a maximum rate or a voidage replacement fraction or a recycling fraction. Please also refer to the documentation for the *MAXRATES keyword for further details. Please note that group cycling control imposed targets will in general override any previously specified group targets through data. If, at any time group cycling control is turned off, any group targets that existed prior to imposition of group cycling control are not as rule re-applied. However the user can choose to re-specify any such targets using a special trigger at the termination of group cycling control. This keyword is conditional. One of the keywords *MAXRATES, *VREPFR or *RECYFR must always be specified. For the *VREPFR keyword, data for a given stream must be specified for all cycle parts.

*RECYFR

This keyword is used to specify the recycling fraction in lieu of a maximum rate or voidage replacement fraction for an injection stream (WATI, GASI), for each cycle part. The group control target for that stream will be imposed such that the specified fraction of total produced volume measured at surface reservoir condition is recycled by injection. For a given stream, specify either a maximum rate or a voidage replacement fraction or a recycling fraction. Please also refer to the documentation for the *MAXRATES keyword for further details. Please note that group cycling control imposed targets will in general override any previously specified group targets through data. If, at any time group cycling control is turned off, any group targets that existed prior to imposition of group cycling control are not as rule re-applied. However the user can choose re-specify any such targets using a special trigger at the termination of group cycling control. This keyword is conditional. One of the keywords

MAXRATES, VREPFR or RECYFR must always be specified. For the RECYFR keyword, data for a given stream must be specified for all cycle parts.

*MAXCUMS

This keyword is used to specify the maximum cumulative production or injection for each stream and for each cycle part. The values entered will be interpreted based on the target types specified using the keyword *TARGETTYPES. The values specified are used to determine cycle part duration. At the end of every timestep the entered maximum cumulative will be compared with the actual volume of fluid injected for the particular stream. If the actual volume produced/injected is greater, then the current cycle is deemed to be completed and the specified group targets for the next cycle in sequence will be imposed starting with the next timestep. If timestep sizes are large it is possible that actual production/injection may exceed considerably the specified cumulative as cycling group control does not control timestep size to meet imposed cumulative exactly. This keyword is optional.

INPUT UNIT SYSTEM / TARGET TYPE	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
STO, STW, STL, STI, BHF, BHW	M^3	Bbl	Cm^3
WTG, STG, BHG, STS, BHS	M^3	Ft^3	Cm^3

*TOTHEAT

This keyword is used to specify the maximum cumulative heat injection for any gas or water injection streams for each cycle part. The values specified are used to determine cycle part duration. At the end of every timestep the entered maximum heat injection will be compared with the actual heat injected for the particular gas or water injection stream. If the actual heat injected is greater, then the current cycle is deemed to be completed and the specified group targets for the next cycle in sequence will be imposed starting with the next timestep. If timestep sizes are large it is possible that actual heat injection may exceed considerably the specified heat injection as cycling group control does not control timestep size to meet imposed targets exactly. This criterion for determining cycle duration can only be specified for gas or water injection streams and for the STARS simulator. This keyword is optional.

INPUT UNIT SYSTEM / TARGET TYPE	SI & MODI SI UNITS (STARS)	FIELD UNITS (STARS)	LAB UNITS (STARS)
STW, BHW, STG, BHG	Joule	Btu	Joule

*MAXBHP

This keyword is used to specify the maximum bottomhole pressure for any injection streams specified for each cycle part. The values specified are used to determine cycle part duration. At the end of every timestep the entered maximum bottomhole pressure will be compared with the actual bottomhole

pressure for all wells that are injecting fluid of the same type as the specified stream. For example if the stream is 'WATI' then the BHP of all water injection wells belonging to the group will be tested against specified value. If the BHP of any one well meets the criterion, then the current cycle part is deemed to be completed and the specified group targets for the next cycle in sequence will be imposed starting with the next timestep. This criterion for determining cycle duration can only be specified for injection streams. This keyword is optional.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
BHP	kPa (SI) kg/cm^2 (MODSI)	psi	kPa

*MAXTIMES

This keyword is used to specify the maximum cycle part duration in terms of time elapsed. The timestep size will also be modified to ensure that in the event that the cycle part duration is controlled by the specified times, then the specified times are honored exactly. This keyword is optional.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
MAX TIME	days	days	minutes

*MINTIMES

This keyword is used to specify the minimum cycle part duration. Therefore the cycle part cannot end even if some other criterion for determining cycle duration such as maximum cumulative is met until the minimum cycle duration time is also met. The timestep size is not modified when the cycle part duration is controlled by the minimum specified time. This keyword is optional. If the minimum cycle part time is not specified then by default a value of zero is assumed.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
MIN TIME	days	days	minutes

*DTWCYC

This keyword is used to specify the starting timestep size for each cycle part. If this keyword is missing the starting timestep size will be selected according to values entered with the *DTWELL and *DTMIN keywords. This keyword is optional.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
TIMESTEP SIZE	days	days	minutes

*NCYCLES

This keyword is used to specify the total number of cycles that must be completed to turn off group cycling control. This keyword is optional. If the number of cycles is not specified then group cycling control will continue to be imposed throughout the simulation unless turned off using *GCONCYCLE *OFF. The user also has the choice of specifying the starting cycle part using the keyword *CYCPRT_START and the ending cycle part with *CYCPRT_END. The actual number of total cycle parts completed will be at least equal to the product of nparts*ncycles. For example if nparts = 3, cycprt_start = 2 and cycprt_end = 1 and ncycles = 3, then the actual number of parts completed will be 9:

2, 3 – for cycle #1

1,2,3 – for cycle #2

1,2,3 – for cycle #3

1 – for cycle #4

Once all the specified cycle parts are completed or the user manually turns of group cycling control using *GCONCYCLE *OFF then any wells that are open as a result of group cycling control and any wells that were shut-in as a result of group cycling control will maintain their status, that is any wells open will remain OPEN and any wells shut will remain shut. In addition any group target or specified maximum constraints that became group targets will NOT be reinstated as group targets at the end of group cycling control. The user has been provided a means through the use of a special trigger to re-specify any group targets at the end of group cycling control. For the above example for cycle ending part number 1, as an illustration, involves production, then all producers will remain open at the conclusion of group cycling and any injection wells that were shut due to operation of group cycling control in earlier cycle parts 2 and 3 will remain shut.

*MINBHP

This keyword is used to specify the minimum bottomhole pressure as a criterion for switching cycle part for production wells. This criterion will only be applied if stream ‘PROD’ is specified and if the maximum rate for the ‘PROD’ stream is greater than zero for the cycle part. The values specified are used to determine cycle part duration. At the end of every timestep the entered minimum bottomhole pressure will be compared with the actual bottomhole pressure for all production wells belonging to the group. If the BHP of any one well meets the criterion, then the current cycle part is deemed to be completed and the specified group targets for the next cycle in sequence will be imposed starting with the next timestep. This criterion for determining cycle duration can only be specified for the production stream. This keyword is optional.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
MINIMUM BHP	kPa (SI) kg/cm^2 (MODSI)	psi	kPa

*DEPNDX

This keyword is used to specify the minimum depletion index as the criterion for switching cycle part for production wells. The depletion index is the ratio of total liquid production for the current cycle divided by total water or steam injected in a previous injection cycle part. This criterion will only be applied if stream 'PROD' is specified and if the maximum rate for the 'PROD' stream is greater than zero for the current cycle part. This criterion is only meaningful is a production cycle part follows an injection cycle part where water or steam is injected. The values specified are used to determine cycle part duration. This criterion for determining cycle duration can only be specified for the production stream. This keyword is optional.

*MINQOIL

This keyword is used to specify the minimum oil rate as a criterion for switching cycle part for production wells. This criterion will only be applied if stream 'PROD' is specified and if the maximum rate for the 'PROD' stream is greater than zero for the cycle part. The values specified are used to determine cycle part duration. At the end of every timestep the entered minimum oil rate for the group will be compared with the actual oil rate for the group. For the criterion to be satisfied the oil rate must first transit from a value higher than the specified minimum to a value less than the specified minimum. This criterion for determining cycle duration can only be specified for the production stream. This keyword is optional.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
MINIMUM OIL RATE measured at surface conditions	M3/day	Bbl/d	Cm^3/min

*MININJ

This keyword is used to specify the minimum injection rate as a criterion for switching cycle part for injection streams. This criterion will only be applied if an injection stream is specified and if the maximum rate for the injection stream is greater than zero for the cycle part. The values specified are used to determine cycle part duration. At the end of every timestep the entered minimum injection rate for the group will be compared with the actual oil rate for the group. For the criterion to be satisfied the injection rate must first transit from a value higher than the specified minimum to a value less than the specified minimum. This criterion for determining cycle duration can only be specified for injection streams. This keyword is optional.

INPUT UNIT SYSTEM	SI & MODI SI UNITS (IMEX, GEM, STARS)	FIELD UNITS (IMEX, GEM, STARS)	LAB UNITS (IMEX, STARS)
WATI (STW/BHW)	M3/day	Bbl/d	Cm^3/min
GASI (STG/BHG)	M3/day	Ft^3/day	Cm^3/min
SOLI (STS/BHS) IMEX only	M3/day	Ft^3/day	Cm3/day

DEFAULTS:

The defaults correspond to the case where a WAG (water alternating gas) process is being modeled. Accordingly the defaults are as follows:

```
*NPARTS 2
*CYCLESTREAMS 'WATI' 'GASI'
*TARGETTYPES
    'WATI'  'STW'  'STW'
    'GASI'   'STG'   'STG'
```

CONDITIONS:

A group must be defined before it can be additionally declared as a cycling group using the keyword *GCONCYCLE_START. Group hierarchy is specified using the keyword *GROUP. It is important to specify all potential wells that can participate in meeting group cycling targets as initially OPEN. The group cycling manager will not open a well if it is specified as SHUT through data. Likewise any wells that are shut in due to an action specified with a MONITOR will not be reopened to meet a group cycling target. Similarly any wells on the drill queue with the status of *AUTODRILL will not be opened by the group cycling manager. All wells of a given type associated with a specific stream that have the status of OPEN will automatically be included in meeting cyclic group targets for that stream. For example for stream 'WATI', water injection, all wells that inject water that are declared as OPEN will be included in meeting the target specified for the 'WATI' stream.

EXPLANATION:

Once all specified cycle parts are completed or alternatively a group previously under cycling control is taken off cycling control by specification of *GCONCYCLE *OFF, then all wells involved in group cyclic control will retain the status in effect at the end of the selected cycle part. In addition any group targets imposed by the group cycling manager will remain in effect as of the last cycle part. The user has the option of changing the status of any of the wells for a given group or modifying/appending to the existing group target constraints by specifying the actions through data using a trigger specifically provided to mark the event corresponding to the lifting of group cyclic control. The syntax of the trigger is:

```
*TRIGGER 'trigger_name' *ON_CYCGRP 'group_name' CYC-END > 0
   Actions
*END_TRIGGER
```

The trigger condition will be satisfied when a group previously under group cyclic control is removed from group cyclic control. This can happen in a number of ways:

i) The group was under cyclic control and the maximum number of cycle parts as stipulated by *CYCPRT_START, *CYCPRT_END and nparts*ncycles have been completed

ii) The group was under group cyclic control and *GCONCYCLE *OFF is specified

iii) The group is redefined with another *GCONCYCLE_START keyword

The user can choose to turn off or suspend group cycling control at any time and turn on group cyclic control back on at any time. If group cycling control is turned back on then group cyclic control will resume with the cycle part following the cycle part at which group cyclic control was turned off regardless of whether the criteria for completing or switching to the next cycle part was fulfilled or not at the time that group cyclic control was turned off. The cycle part number is saved or preserved when group cycling control is turned off. Consider the following example with ncycles = 4, nparts = 2

```
Time = 0.0
*GCONCYCLE 'wfgroup1' *ON
Time T1
GCONCYCLE 'wfgroup1' *OFF ** at this time cyclic group control is in the
middle of total cycle part 5 (cycle #3, part #1)
Time T2
*GCONCYCLE 'wfgroup1' *ON
```

From time T1 to time T2 group cyclic control is suspended. Group targets and well status in effect at the start of total cycle part #5 are imposed.

From time T2 group cyclic control resumes with total cycle part #6 (cycle #3, part #2) regardless of whether the criteria for switching from part 5 to part 6 were met or not.

The user can modify any of the criteria for determining cycle duration at any time using the keyword block GCONCYR_START and GCONCYCLE_END, provided that the cyclic group is defined first using the keyword *GCONCYCLE_START. However one cannot modify the number of cycle parts or streams with GCONCYCR_START. Any changes introduced with *GCONCYCR_START take effect immediately.

The user can specify the starting and/or ending cycle part with the keywords *CYCPRT_START and CYCPRT_END as well as the total number of cycles with *NCYCLES. The actual number of total cycle parts completed will be at least equal to the product of nparts*ncycles. The default for CYCPRT_START is 1 and CYCPRT_END is nparts or the last part. The default for ncycles is infinity. The cycles will start with the specified part and end after completing the cycle part specified with CYCPRT_END. The formula for the actual number of cycle parts completed is

```
Ncycles * nparts + (cycprt_end – cycprt_start + 1)
```

Example 1:

```
Ncycles =13; nparts = 2, cycprt_start = 2, cycprt_end = 1
Ncycles *nparts = 13*2 = 26
Start with cycle #1, part #2 (ipart = 2)
End with cycle #14, part #1 (ipart = 27)
Actual # of parts completed (27-2+1) = 26
```

Example 2:

```
Ncycles = 4, nparts = 3, cycprt_start = 3, cycprt_end = 2
Ncycles *nparts = 4*3 = 12
Start with cycle #1, part #3 (ipart = 3)
Complete 3 additional full cycles (3*3 = 9 cycle parts →
   ipart = 12)
Cycle #4, complete part 1&2 → ipart = 14
Actual number of cycle parts completed 14-3+1 = 12
```

Allow Cycling Group Data to be Revised (Optional)

***GCONCYCR_START, *GCONCYCR_END**

PURPOSE:

*GCONCYCR_START allows the user to revise data specified earlier with *GCONCYCLE_START. The revisions will be applied immediately.

FORMAT:

*GCONCYCR_START ‘group_name1’ ‘group_name2’
(Any of *TARGETTYPES, *MAXCUM, *MAXTIMES, *NCYCLES, MAXBHP, *MININJ, *MINBHP, *MINOIL, *DEPNDX, *TOTHEAT
*GCONCYCR_END

DEFAULTS:

The default is to continue applying values originally specified with *GCONCYCLE_START.

CONDITIONS:

This block of keywords can only be specified for a cycling group if that group is previously defined using the keyword *GCONCYCLE_START. Any or all of the group cycle part switching criteria specified earlier with *GCONCYCLE_START can be revised or new criteria added. However this keyword block cannot be used to change the number of cycle parts (keyword *NPARTS) or the number and type of streams (keyword CYCSTREAMS). Number of parts or streams can only be changed by redefining the group using *GCONCYCLE_START.

EXPLANATION:

See keyword *GCONCYCLE_START

Once a cycling group is defined using *GCONCYCLE_START this keyword block can be used to modify some of the data used to test for cycle part duration. It is also possible to add new criteria to determine cycle part end that were not previously specified with *GCONCYCLE_START. The data entered will take effect immediately. Consider the following example:

```
Time 0.0
GCONCYCLE_START 'cyc_group1'
cycstreams 'PROD' 'WATI'
nparts 3
ncycles 40
targettypes
'WATI' 'STW' 'STW' 'STW'
'PROD' 'STL' 'STL' 'STL'
maxrates
'WATI' 1000. 0. 0.
'PROD' 0.   0. 1000.0
maxcums
'PROD' 0. 0. 600000.0
tothet
'WATI' 5.0E+09 0. 0.
```

```
maxbhp
'WATI' 1100.0 0.0 0.0
maxtimes 10.0 3.0 348.0
dtwcyc 0.02 1.0E+5 1.0

GCONCYCLE_END

time 365.0 ** modify BHP criterion

GCONCYCR_START 'cyc_group1'
*maxbhp
'WATI' 400.0 0.0 0.0
GCONCYCLE_END
```

Allow Cycling Group Data to be Revised (Optional)

***GCONCYCLE**

PURPOSE:

*GCONCYCLE allows the user to turn on or off group cycling control specified earlier with *GCONCYCLE_START. The action indicated with GCONCYCLE takes effect immediately.

FORMAT:

*GCONCYCLE ‘group_name1’ ‘group_name2’ *ON|OFF

DEFAULTS:

Once data is specified is specified with *GCONCYCLE_START, the default is *GCONCYCLE *ON.

CONDITIONS:

This block of keywords can only be specified for a cycling group if that group is previously defined using the keyword *GCONCYCLE_START. Group cycling control can be specified at any time once the group is defined using *GCONCYCLE_START.

EXPLANATION:

See keyword *GCONCYCLE_START

Once a cycling group is defined using *GCONCYCLE_START this keyword can be used to turn off group cycling or to impose group cycling once more if group cycling was previously turned off. The desired state of group cycling control specified with this keyword will take effect immediately. If GCONCYCLE *OFF is specified for a group previously under group control, then group cycling control will be terminated immediately and the group target(s) and well state existing when GCONCYCLE *OFF is specified will continue to be in effect. If actions are specified as part of the special trigger provided to monitor the event of group cycling control being turned off, then that trigger condition will be satisfied and the stipulated actions will take effect at the top of the next timestep. If GCONCYCLE *ON is specified at a time when group control is turned off then group cycling control will resume with the next cycle part following the part when group cycling was turned off. The count of number of cycle parts completed is preserved between applications of GCONCYCLE *ON/*OFF. Consider the following example:

```
Time 0.0
GCONCYCLE_START 'cyc_group1'
cycstreams 'PROD' 'WATI'
nparts 3
ncycles 40
targettypes
'WATI' 'STW' 'STW' 'STW'
'PROD' 'STL' 'STL' 'STL'
maxcums
'PROD' 0. 0. 600000.0
tothet
'WATI' 5.0E+09 0. 0.
```

```
maxbhp
'WATI' 1100.0 0.0 0.0
maxtimes 10.0 3.0 348.0
dtwcyc 0.02 1.0E+5 1.0

GCONCYCLE_END
Time 100
*GCONCYCLE ‘cyc_group1’ *OFF ** was on part 3 when turned of
time 365.0 ** modify BHP criterion

GCONCYCLE 'cyc_group1' *ON ** will resume with cycle part 1
```

Select Cycling Group Cycle Part to Start and End Cycling Group Control (Optional) *CYCPRT_START, *CYCPRT_END

PURPOSE:

*CYCPRT_START allows the user to select the cycle starting part. CYCPRT_END allows the user to select the cycle ending part. Group cycling control must be specified earlier with *GCONCYCLE_START. The action indicated with CYCPRT_START and/or CYCPRT_END take effect immediately.

FORMAT:

*CYCPRT_START ‘group_name1’ ‘group_name2’ cycle_part_start
*CYCPRT_END ‘group_name1’ ‘group_name2’ cycle_part_end

DEFAULTS:

Once data is specified is specified with *GCONCYCLE_START, the default for group cycle starting part is 1 and for group cycle ending part is the value specified with *NPARTS, that is the last cycle part.

CONDITIONS:

These can only be specified for a cycling group if that group is previously defined using the keyword *GCONCYCLE_START.

EXPLANATION:

See keyword *GCONCYCLE_START

Once a cycling group is defined using *GCONCYCLE_START these keywords can be used to select the starting and ending cycle parts. The values specified with these keywords will take effect immediately. If cycling group control is in effect when *CYCPRT_START is specified then the value specified with *CYCPRT_START will be ignored. If *CYCPRT_START is specified following *GCONCYLE *ON then starting cycle part will be the value specified with *CYCPRT_START plus 1. For example

```
Time 0.0
GCONCYCLE_START 'cyc_group1'
cycstreams 'PROD' 'WATI'
nparts 3
ncycles 40
targettypes
'WATI' 'STW' 'STW' 'STW'
'PROD' 'STL' 'STL' 'STL'
maxrates
'WATI' 1000. 0. 0.
'PROD' 0.   0. 1000.0
maxcums
'PROD' 0. 0. 600000.0
tothet
'WATI' 5.0E+09 0. 0.
```

```
maxbhp
'WATI' 1100.0 0.0 0.0
maxtimes 10.0 3.0 348.0
dtwcyc 0.02 1.0E+5 1.0

GCONCYCLE_END
Time 100
*GCONCYCLE ‘cyc_group1’ *OFF ** was on part 3 when turned of
time 365.0 ** modify BHP criterion

*GCONCYCLE 'cyc_group1' *ON ** will resume with cycle part 1
*CYCPRT_START 'cyc_group1' 1 ** will now resume with cycle part 2
```

Control Layer Lumping Option (Optional) *LAYERCLUMP

PURPOSE:

*LAYERCLUMP provides a way for the user to define a control lump - a set of layers (perforations, completions) which possesses a name and a set of computed rate, cumulative, and stream ratio values. To each control lump there is also an associated setting value;

FORMAT:

```
*LAYERCLUMP          Control_lump_name
'well name 1'  layer_id_1
'well name 2'  layer_id_2
          .
          .
          .
'well name n'  layer_id_n
```

Control_lump_name

Name of the control lump; this name may contain up to 40 characters. The name of each control and report lump must be unique. Do not use the same name for a control lump if it is already been assigned to a report lump.

well name 1, well name n

Are valid well names; these names must have been entered using the *WELL keyword before their appearance under the LAYERCLUMP keyword. The same well name may be repeated in this list arbitrarily many times.

Layer_id_1, layer_id_n

Are sequences of triples of integers locating the layer -

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn krn} }

A range may be specified in any of the principal directions i,j, and/or k at any refinement level. A range allows the user to specify multiple layers of a well in a compact form. For example specifying a range such as 'WELL1' 8:12 5 7 is equivalent to listing the layers individually as

'WELL1' 8 5 7

'WELL1' 9 5 7

'WELL1' 10 5 7

'WELL1' 11 5 7

'WELL1' 12 5 7

It is necessary that all layer(s) whether singly entered or with a range belong to the associated well (in the above example 'WELL1') as per the earlier *PERF card for that well.

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It should appear after all well layers in the membership list have been specified with a WELL keyword.

EXPLANATION:

LAYERCLUMP allows the user to group of list of well perforation layers into a single unit or control lump.

For example given the layer list for well ‘Prod-1’ is as follows:

```
*WELL  'Prod-1'
*PERF  *GEOA  'Prod-1'
**$ UBA             ff  Status  Connection
    5 6 1 / 4 4 1    *MT  1.  *OPEN   *FLOW-TO  'SURFACE'  *REFLAYER  **layer  #1
    6 6 1 / 4 4 1    *MT  1.  *OPEN   *FLOW-TO  1                     **layer  #2
    7 6 1 / 4 4 1    *MT  1.  *OPEN   *FLOW-TO  2                     **layer  #3
    8 6 1 / 4 4 1    *MT  1.  *OPEN   *FLOW-TO  3                     **layer  #4
    9 6 1 / 4 4 1    *MT  1.  *OPEN   *FLOW-TO  4                     **layer #5
    10 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  5                     **layer  #6
    11 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  6                     **layer  #7
    12 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  7                     **layer #8
    13 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  8                     **layer  #9
    14 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  9                     **layer  #10
    15 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  10                    **layer # 11
    16 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  11                    **layer # 12
    17 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  12                    **layer # 13
    18 6 1 / 4 4 1   *MT  1.  *OPEN   *FLOW-TO  13                    **layer # 14
```

A control lump which has its members layers 4,5 and 14 of well ‘Prod-1’ can be defined as follows

```
*LAYERCLUMP 'clump1'
  'Prod-1' 8:9  6  1 / 4 4 1 *MT
  'Prod-1' 18   6  1 / 4 4 1 *MT
```

There is no fractional or multiple membership of a layer in a control lump. If LAYERCLUMP is entered more than once with the same 'control_lump_name', the later set of perforations is appended to the set established in the earlier encounters of 'control_lump_name'. The final lump set is the union of all layers entered with *LAYERCLUMP 'control_lump_name'. If a layer is specified more than once for a control lump no error is flagged but nothing is added to the lump's set of perforations - no duplicates of layers are allowed in the internal representation of the control lump.

A layer can belong to only one control lump at a time; it is an error to specify a layer for a new control lump without first removing it from the old lump using *RM-LAYERCLUMP. It is possible that a layer belongs to no control lump; this is represented internally by assigning a control lump number of zero.

To each control lump there is an associated flow restriction or control setting parameter. If the value is 1 there is no effect. If the value is 0 it is like shutting the layers without giving them shut status. The setting value is assigned to all layers in the control lump. The setting value is initialized to 1 for all control lumps. Setting values can be altered in data by using the CLUMPSETTING keyword. Setting values are useful in controlling flow for a set of perforations which share a certain attribute, for example completion depth.

There is also a set of computed rate, cumulative, and stream ratio values for each control lump. Triggers may be defined using these quantities specific to named control lumps; actions can be any valid well data lines, including lines which alter the control lump setting values. The syntax of the control lump based trigger statement is:

*TRIGGER ‘trigger_name’ *ON_CTRLLUMP ‘control_lump_name’ quantity > value

Table below shows the complete list of control lump quantities that may be referenced as a trigger condition. For example to add a trigger on control lump, ‘c_l_1’ on oil production rate greater than 5000.0 the syntax is:

*TRIGGER ‘trig1’ *ON_CTRLLUMP ‘c_l_1’ *STO-RP > 5000.0

{action lines}

*END_TRIGGER

To generate detailed lumped layer table output in the simulation output file add the following line in the simulator data file in the input output (I/O) section:

*OUTPRN *WELL *LAYER

Graphs of computed rate, cumulative and stream ratio values versus time can be created as special history variables. Declare control lump special history variables by adding an OUTSRF WELL line for each quantity in the I/O section of the data file. The syntax is:

*OUTSRF *WELL *CRLLRATE ‘lump_name’

*{SURFACE|PROD|INJ} {OIL|WATER|4TH|LIQUID|BHF}

Choose from one of the selections in the curly bracket. For example for the oil stream and surface volumetric rate at STC (i.e. oil injection rate – oil production rate) the syntax is as follows:

*OUTSRF *WELL *CRLLRATE ‘lump_1’ *SURFACE *OIL

For intermediate liquid phase, assuming this option is turned on the syntax for volumetric rate at STC assuming production stream is desired is

*OUTSRF *WELL *CRLLRATE ‘lump_1’ *PROD *4TH

For cumulative use

*OUTSRF *WELL *CRLLCUM ‘lump_1’ *SURFACE *OIL

For intermediate liquid phase, assuming this option is turned on the syntax for volumetric cumulative at STC assuming production stream is desired is

*OUTSRF *WELL *CRLLCUM ‘lump_1’ *PROD *4TH

For the gas phase volumetric rate at STC the syntax is

*OUTSRF *WELL *CRLGRATE ‘lump_1’ *{SURFACE|PROD|INJ}

For the gas phase volumetric cumulative at STC the syntax is

*OUTSRF *WELL *CRLGCUM ‘lump_1’ *{SURFACE|PROD|INJ}

For gas oil ratio at surface conditions the syntax is

*OUTSRF *WELL *CRLGOR ‘lump_1’

For water cut ratio in percent at surface conditions the syntax is

*OUTSRF *WELL *CRLWCUT ‘lump_1’

For water gas ratio at surface conditions the syntax is

*OUTSRF *WELL *CRLWGR ‘lump_1’

For gas liquid ratio at surface conditions the syntax is

*OUTSRF *WELL *CRLGLR ‘lump_1’

For oil gas ratio at surface conditions the syntax is

*OUTSRF *WELL *CRLOGR ‘lump_1’

To override dimensioning parameters internally determined by scanning the data file with user specified values, add the following line(s) to the recurrent section of the input data file

To change the dimensioning parameter for the total number of control lumps add

*WMDIM * MDCLMP integer1

To change the dimensioning parameter for the total number of layers, all control lumps add

*WMDIM * MDLYCLMP integer2

To change the dimensioning parameter for the total number of report lumps add

*WMDIM * MDRLMP integer3

To change the dimensioning parameter for the total number of layers, all report lumps add

*WMDIM * MDLYRLMP integer4

Table 1: Quantities available for inclusion as a trigger condition for control and report lumps

Keyword	Description
OILSRFR	Oil surface flow rate for lump; + for injection, - for production (sum over all layers in lump) Units: m3/day | stb/day
OILSRFC	Oil surface cumulative for lump; + for injection, - for production (sum over all layers in lump) Units: m^3 | stb
STO-RP	Oil surface production rate for lump (sums layers in producers only) Units: m3/day | stb/day
STO-CP	Oil surface production cumulative for lump; (sums layers in producers only) Units: m^3 | stb
STO-RI	Oil surface injection rate for lump (sums layers in injectors only) Units: m3/day | stb/day

STO-CI	Oil surface injection cumulative for lump; (sums layers in injectors only) Units: m^3 | stb
WATSRFR	Water surface flow rate for lump; + for injection, - for production (sum over all layers in lump) Units: m3/day | stb/day
WATSRFC	Water surface cumulative for lump; + for injection, - for production (sum over all layers in lump) Units: m^3 | stb
STW-RP	Water surface production rate for lump (sums layers in producers only) Units: m3/day | stb/day
STW-CP	Water surface production cumulative for lump; (sums layers in producers only) Units: m^3 | stb
STW-RI	Water surface injection rate for lump (sums layers in injectors only) Units: m3/day | stb/day
STW-CI	Water surface injection cumulative for lump; (sums layers in injectors only) Units: m^3 | stb
GASSRFR	Gas surface flow rate for lump; + for injection, - for production (sum over all layers in lump) Units: m3/day | stb/day
GASSRFC	Gas surface cumulative for lump; + for injection, - for production (sum over all layers in lump) Units: m^3 | stb
STG-RP	Gas surface production rate for lump (sums layers in producers only) Units: m3/day | stb/day
STG-CP	Gas surface production cumulative for lump; (sums layers in producers only) Units: m^3 | stb
STG-RI	Gas surface injection rate for lump (sums layers in injectors only) Units: m3/day | stb/day
STG-CI	Gas surface injection cumulative for lump; (sums layers in injectors only) Units: m^3 | stb
4THSRFR	Intermediate liquid flow rate for lump; + for injection, - for production (sum over all layers in lump) Units: m3/day | stb/day
4THSRFC	Intermediate liquid cumulative for lump; + for injection, - for production (sum over all layers in lump) Units: m3 | stb
4TH-RP	Intermediate liquid surface production rate for lump (sums layers in producers only) Units: m3/day | stb/day
4TH-CP	Intermediate liquid surface production cumulative for lump; (sums layers in producers only) Units: m3 | stb
LIQSRFR	Water + oil surface flow rate for lump; + for injection, - for production (sum over all layers in lump) Units: m3/day | stb/day

LIQSRFC	Water + oil surface cumulative for lump; + for injection, - for production (sum over all layers in lump) Units: m^3 | stb
STL-RP	Water + oil surface production rate for lump (sums layers in producers only) Units: m3/day | stb/day
STL-CP	Water + oil surface production cumulative for lump; (sums layers in producers only) Units: m^3 | stb
STL-RI	Water + oil surface injection rate for lump (sums layers in injectors only) Units: m3/day | stb/day
STL-CI	Water + oil surface injection cumulative for lump; (sums layers in injectors only) Units: m^3 | stb
BHFSRFR	Water + oil + gas + solvent reservoir flow rate for lump; + for injection, - for production (sum over all layers in lump) Units: m3/day | stb/day
BHFSRFC	Water + oil + gas + solvent reservoir cumulative for lump; + for injection, - for production (sum over all layers in lump) Units: m^3 | stb
BHF-RP	Water + oil + gas + solvent reservoir production rate for lump (sums layers in producers only) Units: m3/day | stb/day
BHF-CP	Water + oil + gas + solvent reservoir production cumulative for lump; (sums layers in producers only) Units: m^3 | stb
BHF-RI	Water + oil + gas + solvent reservoir injection rate for lump (sums layers in injectors only) Units: m3/day | stb/day
BHF-CI	Water + oil + gas + solvent reservoir injection cumulative for lump; (sums layers in injectors only) Units: m^3 | stb
GOR	Gas oil ratio at surface conditions for lump (sums layers in producers only) Units: m^3/ m^3 | scf/stb
WCUT	Water cut ratio at surface conditions for lump (sums layers in producers only) dimensionless
WGR	Water gas ratio at surface conditions for lump (sums layers in producers only) Units: m^3/ m^3 | stb/scf
GLR	Gas liquid ratio at surface conditions for lump (sums layers in producers only) Units: m^3/ m^3 | scf/stb
OGR	Oil gas ratio at surface conditions for lump (sums layers in producers only) Units: m^3/ m^3 | stb/scf

Report layer lumping option (Optional) *LAYERRLUMP

PURPOSE:

*LAYERRLUMP provides a way for the user to lump well layers in a group called a report lump. For report lumps, stream rates, cumulatives and production rate ratios are calculated and the user can design triggers based on these quantities. A given layer can be assigned a partial membership to report lump and the same layer can belong to more than one group.

FORMAT:

```
*LAYERRLUMP                    Report_lump_name
'well name 1'  layer_id_1      membership factor 1
'well name 2'  layer_id_2      membership factor 2
               .
               .
               .
'well name n'  layer_id_n      membership factor n
```

Report_lump_name

Name of the report lump; this name may contain up to 40 characters. The name of each control and report lump must be unique. Do not use the same name for a control lump if it is already been assigned to a report lump.

well name 1, well name n

Are valid well names; these names must have been entered using the *WELL keyword before their appearance under the LAYERCLUMP keyword. The same well name may be repeated in this list arbitrarily many times.

Layer_id_1, layer_id_n

Are sequences of triples of integers locating the layer

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn krn} }

A range may be specified in any of the principal directions i,j, and/or k at any refinement level. A range allows the user to specify multiple layers of a well in a compact form. For example specifying a range such as 'WELL1' 8:12 5 7 is equivalent to listing the layers individually as

'WELL1' 8 5 7

'WELL1' 9 5 7

'WELL1' 10 5 7

'WELL1' 11 5 7

'WELL1' 12 5 7

It is necessary that all layer(s) whether singly entered or with a range belong to the associated well (in the above example 'WELL1') as per the earlier *PERF card for that well.

Membership factor 1, membership factor n

Fraction of the total stream production/injection for a given layer that will contribute towards determining the total stream rates and cumulative for the report lump. Membership factor can range from 0 to 1. A given layer may belong to more than one report lump

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. It should appear after all well layers in the membership list have been specified with a WELL keyword.

EXPLANATION:

LAYERRLUMP allows the user to group of list of perforation layers into a single unit or report lump.

Fractional and/or multiple membership of a layer are allowed in a report lump. If LAYERRLUMP is entered more than once with the same 'report_lump_name', the later set of perforations is appended to the set established in the earlier encounters of 'report_lump_name'. The final lump set is the union of all layers entered with *LAYERRLUMP 'report_lump_name'.

There is also a set of computed rate, cumulative, and stream ratio values for each report lump. Triggers may be defined using these quantities specific to named report lumps; actions can be any valid well data lines.

The syntax of the report lump based trigger statement is:

*TRIGGER ‘trigger_name’ *ON_RPTLUMP ‘report_lump_name’ quantity > value

The stream rates, cumulatives and production rate ratios available for use as a trigger condition are identical to the list shown for control lumps in Table 1 (see keyword LAYERCLUMP)

The keyword syntax for writing report lump based rate, cumulative and production rate ratios to the simulation results file is exactly the same as for control lumps and is documented under the keyword LAYERCLUMP.

Control layer lumping option (Optional) *RM-LAYERCLUMP

PURPOSE:

*RM-LAYERCLUMP removes membership of a layer from the named control lump

FORMAT:

```
*RM-LAYERCLUMP                Control_lump_name
'well name 1'  layer_id_1
'well name 2'  layer_id_2
          .
          .
          .
'well name n'  layer_id_n
```

Control_lump_name
: Name of the control lump.

well name 1, well name n
: Are valid well names; The same well name may be repeated in this list arbitrarily many times.

Layer_id_1, layer_id_n
: Are sequences of triples of integers locating the layer -

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn krn} }

A range may be specified in any of the principal directions i,j, and/or k at any refinement level. A range allows the user to specify multiple layers of a well in a compact form. For example specifying a range such as 'WELL1' 8:12 5 7 is equivalent to listing the layers individually as

'WELL1' 8 5 7

'WELL1' 9 5 7

'WELL1' 10 5 7

'WELL1' 11 5 7

'WELL1' 12 5 7

It is necessary that all layer(s) whether singly entered or with a range belong to the associated well (in the above example 'WELL1') as per the earlier *PERF card for that well.

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. The named control lump must have previously been defined using the keyword LAYERCLUMP. The specific layer(s) as stipulated by well name and universal block address must previously been assigned membership to the control lump.

EXPLANATION:

RM-LAYERCLUMP allows the user to remove one or more layers previously assigned to a control lump.

Control layer lumping option (Optional) *RM-LAYERRLUMP

PURPOSE:

*RM-LAYERRLUMP removes membership of a layer from the named report lump

FORMAT:

```
*RM-LAYERRLUMP                Report_lump_name
'well name 1'  layer_id_1
'well name 2'  layer_id_2
          .
          .
          .
'well name n'  layer_id_n
```

Report_lump_name
: Name of the report lump.

well name 1, well name n
: Are valid well names; The same well name may be repeated in this list arbitrarily many times.

Layer_id_1, layer_id_n
: Are sequences of triples of integers locating the layer -

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn krn} }

A range may be specified in any of the principal directions i,j, and/or k at any refinement level. A range allows the user to specify multiple layers of a well in a compact form. For example specifying a range such as 'WELL1' 8:12 5 7 is equivalent to listing the layers individually as

'WELL1' 8 5 7

'WELL1' 9 5 7

'WELL1' 10 5 7

'WELL1' 11 5 7

'WELL1' 12 5 7

It is necessary that all layer(s) whether singly entered or with a range belong to the associated well (in the above example 'WELL1') as per the earlier *PERF card for that well.

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. The named report lump must have previously been defined using the keyword LAYERRLUMP. The specific layer as identified by well name and universal block address should have been previously assigned membership to the report lump.

EXPLANATION:

RM-LAYERRLUMP allows the user to remove one or more layers previously assigned to a report lump.

Control layer lumping option (Optional) *CLUMPSETTING

PURPOSE:

*CLUMPSETTING assigns a control setting or flow restriction parameter to the named control lumps. The flow restriction parameter is used as a multiplication factor in adjusting the value of the well index for all the layers belonging to a given control lump.

FORMAT:

```
*CLUMPSETTING
'control_lump_ name_ 1'  parameter_value_1
'control_lump_ name_2'  parameter_value_2
            .
            .
            .
'control_lump_name_n'  parameter_value n
```

Control_lump_name_1, control_lump_name_2
Name of the control lump.

Parameter_value_1, Parameter_value_2
A real number between 0 and 1.

DEFAULTS:

Optional keyword. The default is to apply a multiplication value of 1.0 to the well index.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. All the named control lumps referenced by this keyword must have previously been defined using the keyword LAYERCLUMP.

EXPLANATION:

CLUMPSETTING allows the user to enter a flow restriction parameter for each control lump. The parameter used to modify the well index for all layers belonging to the control lump. If a value of zero is specified for a given control lump, then all member layers are essentially shut in.

Selection of Instantaneous Vs. Actual Rates Lumps (Optional)

***LAYERLUMP_ACTR**

PURPOSE:

*LAYERLUMP_ACTR selects actual or on time averaged rates and ratios based on the rates in all calculations involving lumps. The alternative (default) is to use instantaneous rates in layer lump calculations.

FORMAT:

```
* LAYERLUMP_ACTR
```

DEFAULTS:

Optional keyword. The default is to use instantaneous rates in all layer lump calculations.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. This keyword must precede all *LAYERCLUMP and *LAYERRLUMP keywords.

EXPLANATION:

There is also a set of computed rate, cumulative, and stream ratio values for each control lump. The default for calculation of rates and quantities related to rates specifically stream ratios such as GOR; WOR is based on instantaneous rates. To select actual or on time average values instead in such calculations add the keyword LAYERLUMP_ACTR prior to specifying the first completion lump in the well and recurrent data section. LAYERLUMP_ACTR is a global keyword and will apply to all control and report lumps. Instantaneous rates will differ from actual rates when on time fractions are specified in the data set. If LAYERLUMP_ACTR is specified then all places where rates and ratios are reported such as the output file, the RESULTS Graph plots and layer lump based triggers will correspond to actual or on time values.

Declaring Cap Rock Leakage Wells (Optional) *CRL_WELLS

PURPOSE:

*CRL_WELLS indicates that the listed wells identified by *well_list* will be considered for cap rock leakage if the requisite criteria are met.

FORMAT:

*CRL_WELLS well_list

DEFINITIONS:

well_list

A list of 'well names' or well numbers to which this status applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility.

DEFAULTS:

Optional keyword. No default value. If a well is not declared a cap rock leakage well using *CRL_WELLS, no leakage will take place through it.

CONDITIONS:

This keyword must be located in the Well and Recurrent data section after wells have been defined with the keyword *WELL. For wells to have cap rock leakage occurring through them based on the criteria specified by *WEL_OPN_THR and related keywords in the Reservoir Description section, they must be declared cap rock leakage wells using *CRL_WELLS.

EXPLANATION:

See explanation in Reservoir Description section under Cap Rock Leakage through Wells.

Example:

```
*GRID *VARI 40 40 25
 .
 .
 ** Cap Rock Leakage Using Wells
 *CROCKTYPE 1
   *WEL_OPN_THR 700.
   *WEL_OPN_MLT 10.
   *WEL_OPN_MMX 5.
   *WEL_OPN_MMN 0.1
 *CROCKTYPE 2
   *WEL_OPN_THR 650.
   *WEL_OPN_MLT 10.
   *WEL_OPN_MMX 5.
   *WEL_OPN_MMN 0.1
*CTYPE *IJK
        1:10  1:10 1:25 1
       11:40 11:40 1:25 2
 .
 .
```

```
*RUN
 *DATE 2000 01 01
 *DTWELL 0.01
 *WELL  1 'CO2-Injector'
   *INJECTOR 'CO2-Injector'
   *INCOMP SOLVENT  1 0
   *OPERATE MAX STG  1.4E+06 CONT
   *OPERATE *MAX *BHP 1.E+05
   *GEOMETRY I 0.2 0.37 1. 0.
   *PERF GEO   'CO2-Injector'
         20 20 5 1. OPEN FLOW-FROM 'SURFACE'
 *WELL 2 'CAP1'
   *PRODUCER 2
   *OPERATE *MIN *BHP 8500.
   *PERF *GEO 2
    15 15 1  1.
 *WELL 3 'PROD'
   *PRODUCER 3
   *OPERATE *MAX *STG 1.E+05
   *OPERATE *MIN *BHP 1500.
   *PERF *GEO 3
    10 10 1  1.
 *WELL 4 'CAP2'
   *PRODUCER 4
   *OPERATE *MIN *BHP 8500.
   *PERF *GEO 4
    5 5 1  1.
 *CRL_WELLS ‘CAP*’ ** Wells 2 and 4 have leakage option
```

Salinity of Injected Water (Optional) *INJ-SALIN

PURPOSE:

*INJ-SALIN specifies the salinity of the injected water into the well in the range of (0 to 15) molal NaCl.

FORMAT:

```
*INJ-SALIN (*PPMWT|*PPMVOL|*MOLAL|*MOLAR|        (well)names
*WTFRAC|*MOLFRAC )
   injsalin_input
```

DEFINITIONS:

PPMWT
: Subkeyword of INJ-SALIN indicating units of salinity will be parts per million by weight (mg NaCl / kg brine).

PPMVOL
: Subkeyword of INJ-SALIN indicating units of salinity will be parts per million by volume (mg NaCl / L brine).

MOLAL
: Subkeyword of INJ-SALIN indicating units of salinity will be molality (gmol NaCl / kg H2O).

MOLAR
: Subkeyword of INJ-SALIN indicating units of salinity will be molarity (gmol NaCl / L brine).

WTFRAC
: Subkeyword of INJ-SALIN indicating units of salinity will be weight fraction (g NaCl / g brine).

MOLFRAC
: Subkeyword of INJ-SALIN indicating units of salinity will be mole fraction (gmol NaCl / gmol brine).

well_names
: Any number of quoted well names to specify the wells whose on-time fractions are defined in this command. These names must be on the same line as the *INJ-SALIN keyword. If more well names are to be specified than can fit on one line, then another *INJ-SALIN keyword must be entered. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. For backward compatibility, well_numbers are still allowed, but they cannot be mixed with the names in the same list.

injsalin_input

The value(s) for the equivalent NaCl concentration of the aqueous phase to be assigned to the wells specified. If no units subkeyword has been specified, the default unit for salinity is molality (gmol NaCl / kg H2O). These values must appear on a (or several) new lines immediately after the line which begins with the *INJ-SALIN keyword. One value must be entered for each well in the list and the Injtemp_input are assigned to the wells in the list in order. The n*value format may be used to enter multiple occurrences of the same fraction. Values greater than [15 molal NaCl] are not acceptable and will produce an error message.

DEFAULTS:

The default for salinity of injection water is zero.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group and may appear anywhere in this keyword group following the initial specification and perforation of wells in the *INJ-SALIN list. The injection salinity for a given well may be altered by an additional *INJ-SALIN line at a subsequent well change. Note that this input is acceptable only when the model has one and only one aqueous component with a component name ‘NaCl’.

EXPLANATION:

*INJ-SALIN is an alternate way of specifying injection salinity compared with the more general *INCOMP *AQUEOUS keyword and related inputs. It offers the choice of a number of salinity unit options and requires specification of only one aqueous component, ‘NaCl’ in the aqueous phase. Also see related keywords *NC-AQUEOUS, *COMPNAME-AQUEOUS, *SALIN-CALC and *SALINR.

Example:

```
1.  *INJ-SALIN  *PPMWT ‘W-1’  ‘W-2’  ‘W-3’
                        58000.  2*70000.

Assigns PPMWT of 58000 to well W-1, and PPMWT of 70000 each to wells W-2
and W-3.

2.  *INJ-SALIN     ‘W-1’  ‘W-2’  ‘W-3’
                    3*1.5

Assigns salinity of 1.5 molal NaCl to each of the wells W-1 through W-3.
```

Temperature of Injected Fluid (Conditional) *INJ-TEMP

PURPOSE:

*INJ-TEMP specifies the bottom-hole temperature of the injected fluid into the well [deg. C | deg. F]. The minimum allowed value is 0.85 deg. C or 33.53 deg. F.

FORMAT:

```
*INJ-TEMP      'well_names'
               Injtemp_input
```

DEFINITIONS:

well_names

Any number of quoted well names to specify the wells whose on-time fractions are defined in this command. These names must be on the same line as the *INJ-TEMP keyword. If more well names are to be specified than can fit on one line, then another *INJ-TEMP keyword must be entered. Limited wildcarding is available for the list of well names; please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. For backward compatibility, well_numbers are still allowed, but they cannot be mixed with the names in the same list.

Injtemp_input

The values for the bottom-hole injection temperature to be assigned to the wells specified. These values must appear on a (or several) new lines immediately after the line which begins with the *INJ-TEMP keyword. One value must be entered for each well in the list and the Injtemp_input are assigned to the wells in the list in order. The n*value format may be used to enter multiple occurrences of the same fraction. Values less than [0.85 deg. C | 33.53 deg. F] are modified to [0.85 deg. C | 33.53 deg. F] with notification.

DEFAULTS:

For thermal runs (*THERMAL *ON), if injection temperature for an injector is not specified via *INJ-TEMP or via the *IWELLBORE keyword, a fatal error message is issued. The value input via *INJ-TEMP keyword for a well is retained till it is overwritten by another *INJ-TEMP or *IWELLBORE keyword. For non-thermal runs specification of injection temperature is not required.

Dynamic Grid Amalgamation Control (Optional) *DYNAGRID, *DYNAGRID-TINT, *DYNAGRID-TSINT, *DYNAGRID-IGN-DIR, *DYNAGRID-WLN-A, *DYNAGRID-WLN-V

PURPOSE:

Create, or remove, amalgamated grid cells and refinements statically, or under simulator control.

FORMAT:

*DYNAGRID *AMALGAMATE (*control_list*) (*EVEN-IF-CMPLX | *EVEN-IF-WELL) *INTO *nir njr nkr*
{ *i1:i2 j1:j2 k1:k2* }
*DYNAGRID *DEREFINE (*control_list*)
{ *i1:i2 j1:j2 k1:k2* }
*DYNAGRID *DEAMALGAMATE
{ *i1:i2 j1:j2 k1:k2* }
*DYNAGRID *REREFINE
{ *i1:i2 j1:j2 k1:k2* }
Where
control_list = (*SATUR *dsn* | *GMOLAR *dzn* | *GMOFRC *dgn* | *OMOFRC *don*)
*DYNAGRID-TINT *tint*
*DYNAGRID-TSINT *tsint*
*DYNAGRID-IGN-DIR (*IDIR | *JDIR | *KDIR | *IJ | *IK | *JK)
*DYNAGRID-WLN-A *iwlna*
*DYNAGRID-WLN-V *iwlnv*

DEFINITIONS:

*AMALGAMATE (*control_list*) (*EVEN-IF-WELL | *EVEN-IF-CMPLX) *INTO *nir njr nkr*

Over the specified block region, amalgamate each regular group of *nir* × *njr* × *nkr* cells blocks together into a single block. Smaller groups near the edges of the region are amalgamated if a whole number of the requested amalgamations cannot be made to fit.

If one or more keywords from *control_list* appear, then amalgamation (or de-amalgamation) occurs dynamically, triggered by the degree of uniformity of properties specified by the *control_list* keywords being satisfied (or not) by the cells in and around each group.

If keyword *EVEN-IF-CMPLX appears then amalgamation is permitted in the specified cell groups even when complex connections are present. See **Complex Connections**, below.

If keyword *EVEN-IF-WELL appears then amalgamation is permitted in the specified cell groups even when active wells are present. See **Active Wells**, below.

*DEREFINE (*control_list*)

De-activate a refinement created by *REFINE, for the specified parent grid blocks. This keyword causes all the child grid blocks in each specified grid block to be replaced with the corresponding fine grid's single parent block, effectively reversing the refinement process.

If one or more keywords from *control_list* appear, then de-refinement (or re-refinement) occurs dynamically, triggered by the degree of uniformity of properties specified by the *control_list* keywords being satisfied (or not) by the cells in and around each group.

*DEAMALGAMATE

Return a previously amalgamated block range to its constituent blocks, effectively reversing the amalgamation process. Only an amalgamated region may be de-amalgamated. For the cancellation to occur, a range following *DEAMALGAMATE must match a range specified for an earlier *AMALGAMATE keyword.

*REREFINE

Re-activate a refinement that was created by *REFINE and then de-activated by *DEREFINE, for the specified fundamental grid blocks.

{ *i1:i2 j1:j2 k1:k2* }

A list of triples of block I-J-K ranges, one per line, specifying groups of fundamental blocks. There must be at least one I-J-K range. See the Keyword Data Entry System section (for instance, the discussion of *IJK) for how ranges work.

control_list

One or more of the following tolerance specifiers.

*SATUR *dsn*	Saturation (water, oil or gas phases)
*GMOFRC *dgn*	Gas phase fraction
*OMOFRC *don*	Oil phase fraction
*WMOFRC *dwn*	Water phase fraction
*GMOLAR *dzn* *	global mole fraction

*DYNAGRID-TINT *tint*

Specify a minimum interval of time *tint* (days) that should exist between dynamic amalgamation and de-refinement checks.

*DYNAGRID-TSINT *itsint*

Specify a minimum number of timesteps *tsint* that should exist between dynamic amalgamation and de-refinement checks.

*DYNAGRID-IGN-DIR (*IDIR | *JDIR | *KDIR | *IJ | *IK | *JK)

Specify a direction or plane that will be ignored when checking for amalgamations. *KDIR is often a useful choice.

*IDIR	Ignore I direction	(check in J-K plane only)
*JDIR	Ignore J direction	(check in I-K plane only)
*KDIR	Ignore K direction	(check in I-J plane only)

*IJ	Ignore I-J plane	(check in K direction only)
*IK	Ignore I-K plane	(check in J direction only)
*JK	Ignore J-K plane	(check in I direction only)

*DYNAGRID-WLN-A *iwlna*

Specify how far away from active wells in the area directions to protect from dynamic amalgamation and de-refinement checks. Integer *iwlna* refers to counting on the finest grid.

*DYNAGRID-WLN-V *iwlnv*

Specify how far away from active wells in the vertical direction to protect from dynamic amalgamation and de-refinement checks. Integer *iwlnv* refers to counting on the finest grid.

DEFAULTS:

All these keywords are optional.

If there are no keywords from *control_list* following *AMALGAMATE or *DEREFINE, the corresponding action is done immediately, that is, at the current (reading) time.

If the keyword *EVEN-IF-CMPLX is absent, then amalgamation is not done for any cell group that contains a complex connection.

If the keyword *EVEN-IF-WELL is absent, then amalgamation is not done for any cell group that contains an active well.

If *DYNAGRID-TSINT and *DYNAGRID-TINT are absent and grid checking is dynamic, then grid checking is done each timestep.

CONDITIONS:

DYNAGRID keywords should be included as the final keywords for a given DATE or TIME specification, following any well specification keywords for that time.

None of these keywords may be used with any of the following grid options in the Reservoir Description section: discretized wellbore (*WELLBORE), natural fracture (*DUALPOR, *DUALPERM, *SUBDOMAIN and *MINC), two-point upstream weighting (*TWOPTUP) and nine-point discretization (*NINEPOINT).

Keywords *DYNAGRID-TINT, *DYNAGRID-TSINT, *DYNAGRID-IGN-DIR, *DYNAGRID-WLN-A, and *DYNAGRID-WLN-V are applicable only for dynamic gridding.

If *DEREFINE is used, then keyword *REFINE in the Reservoir Description data section must have been used to form the refinement.

EXPLANATION:

The recurrent gridding features are designed to save computing time during a simulation by reducing the number of grid cells used in the model. The keywords described here can alter the grid in both a static or dynamic manner. If keywords from *control_list* do not appear, then the grid alterations are static, in that they are applied immediately at a time specified by a *DATE or *TIME keyword. If keywords from *control_list* do appear, the changes are dynamic and depend on the specified trigger criteria interacting with the changing conditions in the specified block ranges.

Amalgamation and De-refinement

The *AMALGAMATE keyword causes specified groups of cells, described by the triples of cell range descriptors listed after the keyword, to be lumped together into larger cells for further computation. When this lumping is done, new cell properties are assigned to the amalgamated cells in such a way as to preserve the total mass of the simulator's components and energy, and to provide appropriate averaged properties for the new cell. Pore volume weighted averaging is used for fluid related primary variables and block volume weighted averaging is used for thermal related primary variables. The extended variables are then computed accordingly. Inter-cell connections are done differently (see below). Amalgamation in and around wells and fault-type connections is not recommended (see below).

The *DEREFINE subkeyword is similar to the *AMALGAMATION subkeyword, although the ranges following this keyword are interpreted in a slightly different way. Each cell referred to in the list of ranges that is a parent cell (and hence was refined earlier using the *REFINE keyword) will be reformed back into its original state, just as if an amalgamation had been applied to the group consisting of all the child cells in the refinement. The parent cell thus becomes active, acting as if it was never refined, and the child cells are ignored in the on-going simulation.

The *DEAMALGAMATE, or *REREFINE, subkeywords cancel the actions of earlier encountered *AMALGAMATE, or *DEREFINE, subkeywords, respectively. Ranges following these keywords need only partially overlap earlier ranges to cause the cancellations to occur.

Note that if a group of cells on a grid were specified for *AMALGAMATE, and one or more of these cells had themselves been refined at earlier times, then amalgamation would still proceed. What would happen is that all refinements of any child cells in the region would be averaged up to their parents (using primarily volume weighted averaging as described earlier), after which the final group of cells would be averaged up as described above. Similarly, if the parent of a group of child cells was selected for '*DEREFINE', and some or all of those cells were themselves refined, then averaging would be done throughout all refinement levels until the parent described by the cell descriptor for the *DEREFINE keyword had been reformed.

Dynamic Trigger Criteria

The *control_list* specifies what level of activity in which properties will be tolerated before cells are amalgamated or de-amalgamated. At a time when the amalgamation state of a cell (or group of cells) is to be evaluated, the values of the properties are examined. For a range of cells described in the *AMALGAMATE keyword's list, if values of each property for the cells and its neighbours are the same within that property's specified threshold, that group will be marked for amalgamation. For example, if *TEMPER 1.0 is used and a cell group and neighbours has temperatures all within 0.5 of each other, then that cell group would be amalgamated. On the other hand, if a group of cells had passed these tests previously and were amalgamated under the *AMALGAMATE keyword, that amalgamated cell's value would be checked against its neighbours using the thresholds. If the difference of cell's value and its neighbours lie outside of the specified threshold(s), then the group will be marked for de-amalgamation. Similar calculations are done in the case of *DEREFINE. Thus, groups of cells will be amalgamated and de-amalgamated dynamically, depending on grid activity.

Dynamic grid keywords are first applied at the time specified by the *TIME or *DATE keyword at which they appear. However, dynamic amalgamation or de-refinement will be re-considered each timestep thereafter, except as controlled by the *DYNAGRID-TSINT and *DYNAGRID-TINT qualifiers.

Addressing Grid Blocks

All references made in the data set to cells, such as those made for grid property input, initialization, well perforations, etc., must always be made with respect to the finest grid, which is the grid resulting from the removal of all *DYNAGRID data. In fact, if all *DYNAGRID keywords were removed then the data set would (and must) still be a valid data set and would run a normal simulation. The simulator makes all the internal changes required to run a simulation with amalgamated cells and does not allow the user to address the amalgamated cells directly.

Interblock Connections

Flow connections between cells constructed by *AMALGAMATE or *DEREFINE are built up from the flow connections calculated on the finest grid, that is, from the usual connections that are constructed when no amalgamations or de-refinements of any type are present. When a connection between an amalgamated cell and some other cell (amalgamated or not) needs to be calculated, the connection is built by summing over the connection pairs between finest grid cells, where one member of the pair is in the amalgamated region and the other member is the other cell (or a member of its amalgamation, if there is one). The overlap areas (which incorporate transmissibility multipliers, if present) are summed and an overall permeability is formed by taking an area-weighted sum of the harmonically-averaged permeabilities that come from the fine-scale connections. An overall transmissibility is then computed by multiplying the overall permeability by the summed areas, and dividing by the cell centre-to-centre distance for the cells involved. Each cell's centre-to-face distance comes directly from the finest grid for an un-amalgamated cell, and otherwise is constructed when other property construction takes place for amalgamated cells.

Note that under the current flow connection regime described here, the only permeabilities that matter are those that were assigned to the fine grid cells on the boundaries of the amalgamated region (as these enter into the only fine grid transmissibilities that were used to build the new connection). This means that flow barriers (zero permeabilities and zero transmissibilities) strictly within amalgamated regions will not affect the overall flow. Thus, amalgamated regions should consist of relatively homogeneous cells. Obviously, all the cells in an amalgamated region should share the same rock types, etc., to avoid any deviations from the original settings.

Complex Connections

By default a cell group is not amalgamated if it contains complex fault-type connections, that is, certain faces of some cells have multiple, or no, connections to their neighbours. If used, the keyword *EVEN-IF-CMPLX must be used with care. As described above, the only permeabilities that matter for amalgamated regions are those that were assigned to the fine grid cells on the boundaries of the amalgamated region. Consequently, if these keywords are used then a flow barrier entirely inside an amalgamated block effectively disappears.

Active Wells

By default a cell group is not amalgamated if it contains an active well. If used, the keyword *EVEN-IF-WELL must be used with care. When a well is completed (perforated) in cells that are taken up in some kind of amalgamation, that well is re-perforated in the amalgamated cell and its productivity indices are recomputed based on the properties of the amalgamated cell. If the well originally perforated multiple cells that were taken up in a single amalgamated cell, then only one perforation will remain (for the amalgamated cell itself), and the remaining perforations will be marked as having been made in null cells. This makes it difficult, if not impossible, to match the results of original fine grids for many simulations.

Special Connections

The *SCONNECT keyword (in Reservoir Description section) is not supported in models using *DYNAGRID. For such models irregular interblock connections via. keyword *IRCONNECT are, however, allowed.

Custom Well Output (Optional) *WELPRN

PURPOSE:

*WELPRN is used to output supported custom well quantities to the GEM out file.

FORMAT:

*WELPRN (*well_list*) *supported quantity*

where *supported quantity* is any one of,
*WISTAT | *WI | *WIANDSTAT | *OFF

DEFINITIONS:

well_list
A list of well names or well numbers to which this definition applies. A limited wildcard facility is available for lists of well names. Please see the manual page for the *SHUTIN keyword for an explanation of the wildcard facility. If no *well_list* is entered, it applies globally to all wells including the future ones.

*WISTAT
Print a table of well index statistics, i.e, maximum, minimum, mean, and median value of completion layer well indices for each well in the list where significant well or perforation change has occurred.

*WI
Print a table of completion layer wise well index and associated key parameters for each well in the list where significant well or perforation change has occurred.

*WIANDSTAT
Print well index statistics table and the well index table for wells in the list where significant well or perforation change has occurred.

*OFF
Turn off the output of any of the *WISTAT, *WI, or *WISTAT specified at previous times, for the current and subsequent times.

DEFAULTS:

The default is *WELPRN *OFF, i.e., none of tables associated with *WISTAT, *WI, and *WIANDSTAT will be output.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword turns on the output of the statistics or well index for all the well completions for wells specified via thw *well_list*. In absence of the well_list on the *WELPRN input, printing is activated for all the current and future wells where significant well and perforation changes occur.

The output may be particularly useful to regulate or limit the WI of layers in all or in a selected number of wells. See keyword *WIRANGE for further details.

The keyword needs to be used with caution since it may result in a very large output file, especially for models with a large number of wells and/or well completions.

Example:

```
*WELL 1 'Producer1'
*WELL 2 'Injector1' *VERT 12 14
*WELL 3 'Prod. 54-233a'
*WELPRN 1 2 3 *WISTAT
*WELPRN 2 *WI
```

Terminate Simulation (Required) *STOP

PURPOSE:

*STOP causes simulation to terminate.

FORMAT:

*STOP

DEFAULTS:

Required keyword. No default.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group. *STOP should follow immediately a *DATE or *TIME data line; otherwise the restart file written after the last timestep may not be correctly interpreted in a restarting run.

EXPLANATION:

This keyword is generally the last keyword in the input data file, and is used in conjunction with a *DATE or *TIME keyword to indicate when to stop the simulation. However, it may appear at any point within the recurrent data section to terminate the simulation at that point without deleting subsequent data from the file. Data after the *STOP keyword are ignored.

For example, to complete the input data file in the normal manner at a specific date, use:

```
:
*DATE 1997 10 22
*STOP
```

To terminate simulation after 7 years, without deleting the remainder of the input data file, use:

```
:
*TIME 2555.0
*STOP
```

Note that although the *STOP line may be followed by (ignored) lines of data, *STOP must always be preceded immediately by a *DATE or *TIME line.

Appendix A

Theoretical Outline

This section describes the equations and variables used in GEM and the approach for solving these equations. The flow equations are discretized using the adaptive-implicit method (Collins, Nghiem and Li, 1986; Thomas and Thurnau, 1983) because i encompasses both the explicit-transmissibility method and the fully-implicit method as particular cases.

The equations, variables and solution method presented in the following are variations of the approach of Collins, Nghiem and Li (1986).

Flow Equations

The material-balance finite-difference equations for the components in the oil and gas phases, and for the water component are:

$$\psi_i \equiv \Delta T_o^m y_{io}^m \left(\Delta p^{n+1} - \gamma_o^m \Delta D\right) + \Delta T_g^m y_{ig}^m \left(\Delta p^{n+1} + \Delta P_{cog}^m - \gamma_g^m \Delta D\right)$$

$$+ q_i^m - \frac{V}{\Delta t}\left[N_i^{n+1} - N_i^n\right] = 0 \qquad i = 1,\ldots,n_c \qquad \text{(A.1.1)}$$

$$\psi_{n_c+1} \equiv T_w^m \left(\Delta p^{n+1} - \Delta P_{cwo}^m - \gamma_w^m \Delta D\right) + q_{n_c+1}^m - \frac{V}{\Delta t}\left[N_{n_c+1}^{n+1} - N_{n_c+1}^n\right] = 0 \qquad \text{(A.1.2)}$$

where Ni (i=1,...,nc) denote the moles of Component i per unit of gridblock volume, and where N_{n_c+1} denotes the moles of water per unit of gridblock volume. All other symbols are defined in the nomenclature. It is assumed that no mass transfer exists between the hydrocarbon and water phases. The superscripts n and n+1 denote respectively the old and current time level. The superscript m refers to n for explicit gridblocks and n+1 for fully-implicit gridblocks. In GEM, the term explicit refers to gridblocks with explicit transmissibilities where only pressure is treated implicitly.

The Ni's are related to porosities phase molar densities, saturations and compositions as follows:

$$N_i = \phi(\rho_o S_o y_{io} + \rho_g S_g y_{ig}) \qquad i = 1,\ldots,n_c \qquad \text{(A.1.3)}$$

$$N_{n_c+1} = \phi \rho_w S_w \qquad \text{(A.1.4)}$$

Phase-Equilibrium Equations

If the hydrocarbon system is in the two phase region at a given p, T and Ni (i=1,...,nc), the phase compositions and splits can be obtained by solving the thermodynamic-equilibrium equation

$$\mathbf{g}_i \equiv \ln f_{ig} - \ln f_{io} = 0 \qquad i=1,...,nc \qquad (A.1.5)$$

for Nig, the moles of Component i in the gas phase. The moles of Component i in the oil phase, Nio, can be obtained from

$$N_{io} = N_i - N_{ig} \quad i = 1,...n_c \qquad (A.1.6)$$

Saturation Equation

The saturations are related to Ni and ρ_m (m = o, g, w) through the following equation

$$S_w = N_{n_c+1} / (\phi \rho_w) \qquad (A.1.7)$$

$$S_o = (1 - S_w) \frac{N_o / \rho_o}{N_o / \rho_o + N_g / \rho_g} \qquad (A.1.8)$$

$$S_g = (1 - S_w) \frac{N_g / \rho_g}{N_o / \rho_o + N_g / \rho_g} = 1 - S_w - S_o \qquad (A.1.9)$$

Mole or Volume Consistency Equations

From the definition of Ni (i = 1,...,nc + 1), one can write

$$\psi_p \equiv \sum_{i=1}^{n_c+1} N_i^{n+1} - \phi^{n+1} \left(\rho_o S_o + \rho_g S_g + \rho_w S_w \right)^{n+1} = 0 \qquad (A.1.10)$$

Equation (1.1.10) forces the consistency between the Ni's and the densities, saturations, and porosities.

Equation (A.1.10) can also be rewritten as follows

$$\psi'_p \equiv V \frac{\sum_{i=1}^{n_c+1} N_i^{n+1}}{\left(\rho_o S_o + \rho_g S_g + \rho_w S_w \right)^{n+1}} - V \phi^{n+1} = 0 \qquad (A.1.11)$$

The first term in Equation (A.1.11) is the volume occupied by the fluids (oil, gas, water) and the second term is the pore volume. Thus Equation (A.1.11) forces the consistency between the fluid volume and the pore volume. A similar equation is referred to as volume balance equation by Acs et al (1985) and Watts (1986).

Decoupled Flash-Calculation Approach

Equations (A.1.11), (A.1.1), (A.1.2), and (A.1.5) form a system of nb (2nc + 2) non-linear equation which can, in principle, be solved for the variables

$$(p, N_1, ..., N_{n_c}, N_{n_c+1}, N_{1g}, ..., N_{n_c g})_k^{n+1} \quad \text{with} \quad k = 1, ..., n_b$$

where nb is the number of gridblocks. This approach will be referred to as the simultaneous-solution approach.

To alleviate the complexity associated with the simultaneous-solution of all governing equations, Collins, Nghiem and Li (1985) decoupled the phase-equilibrium equations and solved them in an inner loop. The technique used in GEM is a variation of their approach and will be referred to as the decoupled-flash-calculation approach. The equations and primary variables of gridblock k are

$$\mathbf{F}_k = \left(\psi'_p, \psi_1, \ldots, \psi_{n_c+1}\right)_k^T \quad k = 1, \ldots, n_b \qquad \text{(A.1.12)}$$

$$\mathbf{X}_k = \left(p^{n+1}, N_1^{n+1}, \ldots, N_{n_c+1}^{n+1}\right)_k^T \quad k = 1, \ldots, n_b \qquad \text{(A.1.13)}$$

After every Newtonian iteration of X, the phase-equilibrium equation (A.1.5) are solved to convergence.

The approach taken here separates the task of solving the consistency and flow equations from that of solving the phase-equilibrium equations. This is a desirable feature because flash calculations are localized in a module that can be developed independently. It also allows the use of different calculation techniques (e.g. Newton's method, quasi-Newton method, QNSS) and the easy implementation of stability-test procedures.

It is possible that this decoupled-flash-calculation approach may not be as efficient as the simultaneous-solution approach for systems with simple phase behavior, but it provides additional flexibility in implementing more robust flash algorithms for complex systems. It also allows the ability to repeat the flash calculations for gridblocks where convergence to unrealistic or metastable phases are obtained. The simultaneous-solution approach may require a repeat of the timestep. Note that the decoupled flash-calculation approach enforces the thermodynamic equilibrium between oil and gas at every iteration whereas in the simultaneous-solution approach, equilibrium is only satisfied after the convergence of the timestep.

Nomenclature

D	depth
f_{ij}	fugacity of component I in phase j
$\mathbf{F}$	function
g	phase-equilibrium function
n_b	number of gridblocks
n_c	number of components
N_i	moles of component i per unit block volume
p	pressure
P_{cog}	oil-gas capillary pressure
P_{cwo}	water-oil capillary pressure
q	injection/production rate
t	time
T_j	transmissibility of phase j

V	gridblock volume
y_{ij}	mole fraction of component I in phase j
γ	specific gravity or gravity term in flow equation
Δt	timestep
ρ_m	molar density of phase m
ϕ	porosity
ψ	function

Superscripts

(k)	iteration level
n	old time level
n+1	new time level

Subscripts

i	component
j	phase
o	oil
g	gas
w	water

References:

Acs, G., Doleschall, S., and Farkas, E., "General Purpose Compositional Model," Soc. Petrol. Eng. J., Vol. 25, August 1985, pp. 543-553.

Collins, D.A., Nghiem, L.X., and Li, Y.K., "An Efficient Approach to Adaptive-Implicit Compositional Simulation With an Equation of State," paper SPE 15133, presented at the 1986 California Regional Meeting, April 2-4, 1986, Oakland, California.

Nghiem, L.X., and Li, Y.-K., "Phase-Equilibrium Calculations for Reservoir Engineering and Compositional Simulation," paper to be presented on the Second International Forum on Reservoir Simulation, Alpbach, Austria, September 4-8, 1989.

Thomas, G.W., and Thurnau, D.H., "Reservoir Simulation Using an Adaptive-Implicit Method," Soc. Petrol. Eng. J., Vol. 23, October 1983, pp. 759-768.

Watts, J.W., "A Compositional Formulation of the Pressure and Saturation Equations," SPE Res. Eng., Vol. 1, May 1986, pp. 243-252.

Well Model

Injector

The injection well model correlates the reservoir flow rate of phase j (j = g, w) to the wellbore pressure, p_{bh} and the pressure at grid point via the relationship:

$$Q_j = \sum_l WI_l \, \lambda_{T,l} (p_{bh} - p_{o,i}) \qquad j = g, w \qquad \text{(A.2.1)}$$

$$WI = 2\pi \, ff \, kh \frac{wfrac}{\ln(r_e / r_w) + S} \qquad j = g, w \qquad \text{(A.2.2)}$$

with $p_{bh} > p_{o,i}$

where

Q_j = flow rate of phase j (j = g, w) at reservoir conditions (m^3/day | ft^3/day)

p_{bh} = bottomhole pressure (kPa | psia)

$p_{o,i}$ = pressure of i-th grid block containing the well (kPa | psia)

$WI_{j,l}$ = well injectivity index for phase j (j = g, w) to layer l

wfrac = well fraction, governed by areal geometry (see Figure A-1)

k = effective permeability in the plane perpendicular to the well direction

h = gridblock thickness in well direction (m | ft)

$\lambda_T = \sum_j \frac{k_{rj}}{\mu_j} j=o,g,w$ is the total mobility of the fluid in the well block. The relative permeabilities are calculated using the gridblock saturation

r_w = wellbore radius (m | ft)

$r_e = CC\sqrt{A_i / wfrac}$ effective radius (m | ft)

A_i = area of i-th gridblock perpendicular to well direction (m^2 | ft^2)

CC = geometric factor (see Figure A-1)

S = skin factor (dimensionless)

ff = fraction of completion of the well in the gridblock

$\sum_l$ = summation over all layers

The well model is a generalization of the well model proposed by Peaceman (1987, 1982) for square and non-square gridblocks. The mobility treatment follows the suggestion of Chappelear and Williamson (1979). In addition, the geometric factor allows the determination

of the equivalent radius from both the geometry of the gridblock and the location of the well in the gridblock.

Producer

The production well model is similar to the preceding case. The flow rate of phase j (j = o, g, w) at reservoir conditions is given by:

$$Q_j = \sum_l PI_{j,l} \left(\frac{k_r}{\mu} \right)_{j,l} \left(p_{o,i} - p_{bh} \right) \qquad j = o, g, w \qquad \text{(A.2.3)}$$

$$PI_j = 2ff\, kh \frac{wfrac}{\ln\left(r_e / r_w\right) + S} \qquad \text{(A.2.4)}$$

with $p_{bh} < p_{o,i}$

where

k_{rj}	=	relative permeability of phase j in the i-th gridblock (fraction), it is calculated using the gridblock saturations
μ_j	=	viscosity of phase j in the i-th gridblock, cp
PI_j	=	well productivity index for phase j (j = o,g,w) from layer 1

All the other parameters have the same definitions as in Equation (A.2.2).

In many field simulation problems, the well is completed through several gridblocks and either the total rate or the bottomhole pressure of that multilayer well is specified. In these cases the summation in l must be carried out over all completion blocks. Figure A-2 shows a multiblock well completion where for convenience both the gridblocks and well elements are numbered from 1 to n_e. The pressure in the i-th well element, $p_{bh,i}$, is related to the bottomhole pressure, $p_{bh,l}$ by the following equation:

$$p_{bh,i} = p_{bh,l} - \int_{h_i}^{h_l} \tilde{\rho}^T g\, dh \qquad \text{(A.2.5)}$$

where $\tilde{\rho}^T$ is the mass density of all phases (kg/m^3) within the wellbore, g the gravitational acceleration (m/s^2) and h the depth (m), measured positive downward.

For 1-D slim tube runs, the following are used:

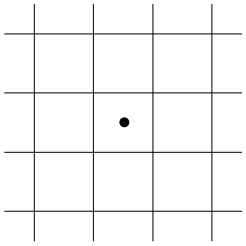
<u>Injector</u>

$$Q_j = CC\, k_x \left(\Delta y \Delta z\right) \lambda_T \left(p_{bh} - p_{o,i}\right) ff$$

<u>Producer</u>

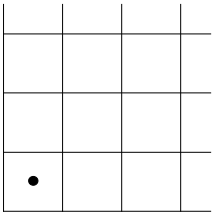
$$Q_j = CC\, k_y \left(\Delta y \Delta z\right) \left(\frac{k_r}{\mu} \right)_j \left(p_{o,i} - p_{bh}\right) ff$$

Figure A-1: Well Fraction and Geometrical Factor for Various Common Geometries



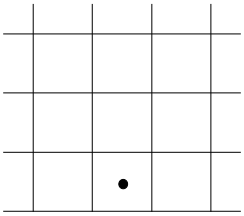
f = 1
CC = 0.37

(a) Centre of Square Grid Block



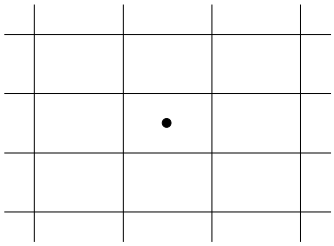
f = 1
CC = 0.34

(b) Centre of Corner Block



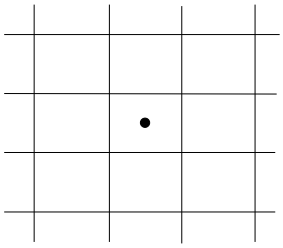
f = 1
CC = 0.35

(c) Centre of Edge Block



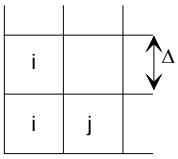
f = 1 **CC = 0.41**

(d) Centre Rectangular Grid Block Ratio $\Delta x/\Delta y=2$



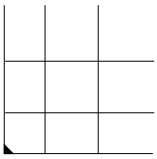
f = 1 **CC = 0.39**

(e) Centre Rectangular Grid Block Ratio $\Delta x/\Delta y=3/2$



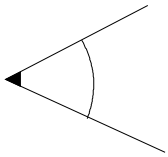
f = 1/4
CC = 0.64

(f) Well in Corner



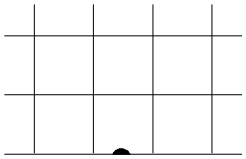
f = 1/4
CC = 0.56

(g) Edge Blocks Half Size



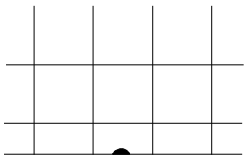
f = segment
CC = 0.5

(h) Radial Segment
2π radian f = 1
π radian f = 1
$1/2\pi$ radian f = 1/4



f = 1/2
CC = 0.54

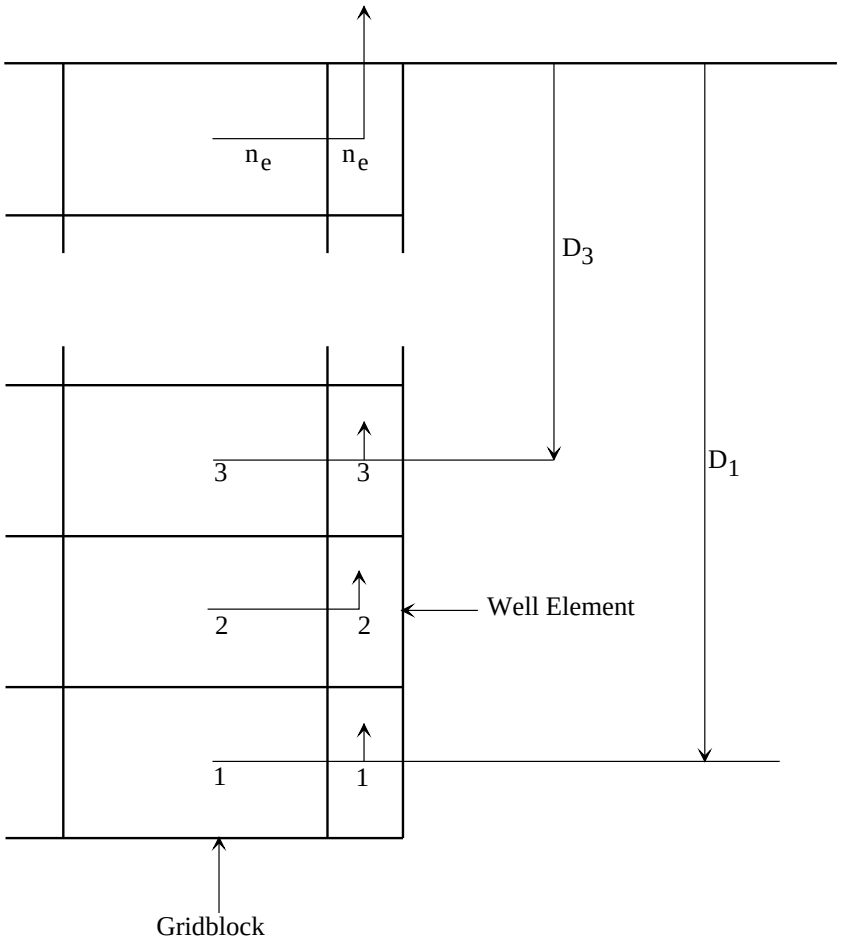
(i) Well on Edge



f = 1/2
CC = 0.47

(j) Edge Blocks Half Size

Figure A-2: Multiblock Well Completion



Wellbore Model

Pressure Loss Equation

The equation governing the change in pressure in the direction of flow used for the wellbore model is

$$\Delta p = \Delta p_H - \Delta p_{KE} - \Delta p_F \quad \text{(A.3.1)}$$

where

$$\Delta p_H = \rho\, g\, \Delta z \quad \text{(Hydrostatic head over the length } \Delta z\text{)}$$

$$\Delta p_F = \frac{2fv^2\rho}{D}\Delta z \quad \text{(Friction gain over a length Dz)}$$

$$\Delta p_{KE} = -\rho v^2 \ln\left(\frac{p_2}{p_1}\right) \quad \text{(Kinetic energy gain)}$$

Here

ρ = density of the flowing "in situ" mixture

g = acceleration due to gravity

f = fanning friction factor

v = average velocity of mixture

D = inside pipe diameter

$\Delta p = p_2 - p_1$ = pressure drop over the length Δz

The wellbore model is a modification of the method of Aziz et al (1972). The modifications involve the use of an EOS to calculate the phase behavior and fluid properties of the flowing fluid. Further details may be found in Agarwal and Li (1988).

References:

Aziz, K., Govier, G.W., and Fogarasi, M., "Pressure Drop in Wells Producing Oil and Gas," Journal of Canadian Petroleum Technology, July-Sept., 1972.

Agarwal, R.K., and Li, Y.-K., "Implementation of a Simplified Wellbore Model in a Compositional Simulator," CMG report 88.13.C, December, 1988.

Aquifer Model

There are two aquifer models available in GEM. These are the steady-state aquifer and the Carter-Tracy aquifer model (Dake, 1978).

Steady-State Aquifer

This model assumes that the pressure at the external boundary of the aquifer does not change. Then the rate of water influx is

$$Q_A = C_A\left(p_A^{(i)} - \overline{p}\right)$$

where

C_A	=	total compressibility of the aquifer = water compressibility + rock compressibility
$p_A^{(i)}$	=	pressure at the external boundary of the aquifer and is constant
$\overline{p}$	=	the average pressure at the aquifer-reservoir interface

Carter-Tracy Aquifer

In this approach the rate of water influx during the current timestep is assumed to be given by (Carter and Tracy, 1960)

$$q_w^{n+1} = a(t) + b(t)\ \{p_{ij}^{n+1} - p_{ij}^{n}\}$$

where p_{ij}^{n} is the pressure of block i, j at timestep n, and

$$a = \left[\frac{B_l\left(p_{ij}^{o} - p_{ij}^{n}\right) - W^n PI'^{n+1}}{PI^{n+1} - t_D^n PI'^{n+1}}\right]\left[\frac{t_D^{n+1} - t_D^n}{t^{n+1} - t^n}\right]$$

$$b = \left[\frac{-B_l}{PI^{n+1} - t_D^n PI'^{n+1}}\right]\left[\frac{t_D^{n+1} - t_D^n}{t^{n+1} - t^n}\right]$$

W = cumulative water influx

$B_l = 2\pi f\ \phi h\ \ \overline{c} r_o^2$

h = aquifer thickness

r_o = effective reservoir radius

$\overline{c}$ = total compressibility (water + rock)

f = $\theta/2\pi$ angle of influence

t_D = dimensionless time = $\dfrac{kt}{\phi\mu\overline{c}r_o^2}$

$PI(t_D)$ = dimensionless water influx function for constant terminal rate case

$PI'(t_D)$ = derivative of $PI(t_D)$ with respect to t_D

θ = encroachment angle

Further details of the Carter-Tracy unsteady-state aquifer model can be found in Carter and Tracy (1960). The built-in aquifer influx function PI is shown in Table A-1. This influx function is only valid for a radial aquifer.

References:

Carter, R.D., and Tracy, G.W., "An Improved Method for Calculating Water Influx," Petroleum Transactions, AIME, Vol. 219, 1960, pp. 415-417.

Dake, L.P., "Fundamentals of Reservoir Engineering," Chapter 9, Elsevier Scientific Publishing Co., 1978.

Table A-1: Built in Aquifer Influx Function

t	PI
0.00	0.000
0.01	0.112
0.05	0.229
0.10	0.315
0.15	0.376
0.20	0.424
0.30	0.503
0.50	0.616
0.70	0.702
1.00	0.802
1.50	0.927
2.00	1.020
3.00	1.169
5.00	1.362
7.00	1.500
10.00	1.651
15.00	1.829
20.00	1.960
30.00	2.147
40.00	2.282
50.00	2.388
60.00	2.476
70.00	2.550
80.00	2.615
90.00	2.672
100.0	2.723
200.0	3.064
300.0	3.263
400.0	3.406
500.0	3.516
600.0	3.608
700.0	3.684
800.0	3.750
900.0	3.809
1000.0	3.860

Solubility Model

Flow Equations for Simplified Henry's Law Flash for Species i

$$\psi_i \equiv \Delta\left[T_o^m y_{io}^m \left(\Delta p^{n+1} - \gamma_o^m \Delta D\right)\right] + \Delta\left[T_g^m y_{ig}^m \left(\Delta p^{n+1} + \Delta p_{cog}^m - \gamma_g^m \Delta D\right)\right]$$
$$+ \Delta\left[T_a^m y_{ia}^m \left(\Delta p^{n+1} - \Delta p_{cao}^m - \gamma_a^m \Delta D\right)\right] + q_i^{n+1}$$
$$- \frac{V}{\Delta t}\left[\phi^{n+1}\left(N_i^{n+1} + N_{a_i}^{n+1}\right) - \phi^n\left(N_i^n + N_{a_i}^n\right)\right] = 0 \qquad \text{(A.5.1)}$$

For Water

$$\psi_{n_c+1} = \Delta\left[T_a^m y_{wa}^m \left(\Delta p^{n+1} - \Delta p_{cao}^m \gamma_a^m \Delta D\right)\right] + q_{n_c+1}^{n+1}$$
$$- \frac{V}{\Delta t}\left[\phi^{n+1} N_{aw}^{n+1} - \phi^n N_{aw}^n\right] = 0 \qquad \text{(A.5.2)}$$

Volume Balance Equation

$$\psi_v = \frac{\sum_{i=1}^{n_c}\left(N_i^{n+1} + N_{ai}^{n+1}\right) + N_{aw}^{n+1}}{\rho_o S_o + \rho_g S_g + \rho_a S_a} - v\phi^{n+1} = 0 \qquad \text{(A.5.3)}$$

where

T_j = transmissibility of phase j, j = o, g, a

y_{ij} = mole fraction of component i in phase j

i = 1, ..., n_C, for oil and gas phases

i = 1, ..., n_C+1 for aqueous phase

γ = gravity potential of phase j

ΔD = depth in gravity direction

N_i = moles of species i in the oil and gas phases

N_{ai} = moles of species i in the aqueous phase

Relationship Between N_i and N_{ai}

1. Perform a two phase flash and calculate fo or fg or both
2. ai = fgi/Hi from Henry's law where Hi is the Henry's law constant (A.5.4)
3. $y_{aw} = 1 - \sum_{i=1}^{n_C} y_{ai}$ (A.5.5)
4. Naw is a primary variable and is thus a known quantity

5. $N_{ai} = y_{ai}\left(\frac{N_{aw}}{y_{aw}}\right)$ (A.5.6)

6. Total moles = $N_{Ti} = N_{ai} + N_i$, where N_i are the combined moles in the oil and gas phases

In step 2 above the Henry's law constant Hi is determined from the following equation

$$H_i = H_i^* \exp\left[\frac{v_i^\infty (p - p_{ref})}{RT}\right]$$

where H_i^* is the reference Henry's law constant at a pressure p_{ref}, v_i^∞ is the partial molar volume of component i at infinite dilution. R is the universal gas constant and T is the temperature.

Dual Porosity / Dual Permeability Model

The following describes the governing equations for the dual-porosity and dual-permeability approach to modelling naturally fractured reservoirs. The governing equations are an extension of the equations for single-porosity systems described in Nghiem and Li (1988). The representation of the matrix follows Kazemi et al (1976) where fractures are assumed orthogonal in the three directions and acts as boundaries to matrix elements.

Dual Porosity Formulation - Matrix Blocks

$$\psi_{im} = -\tau_{iomf} - \tau_{igmf} - \frac{V}{\Delta t}\left(N_i^{n+1} - N_i^n\right)_m = 0 \qquad i = 1,\ldots,n_c \qquad \text{(A.6.1)}$$

$$\psi_{n_c+1,m} = -\tau_{wmf} - \frac{V}{\Delta t}\left(N_{n_c+1}^{n+1} - N_{n_c+1}^n\right)_m = 0 \qquad \text{(A.6.2)}$$

$$\psi_{pm} = \sum_{i=1}^{n_c+1} N_{im}^{n+1} - \phi_m^{n+1}\left(\rho_o S_o + \rho_g S_g + \rho_w S_w\right)_m^{n+1} = 0 \qquad \text{(A.6.3)}$$

Dual Porosity Formulation - Fracture Blocks

$$\begin{aligned}\psi_{if} &= \Delta T_{of}^s y_{iof}^s \left(\Delta p^{n+1} - \gamma_o^s \Delta D\right)_f \\ &+ \Delta T_{gf}^s y_{igf}^s \left(\Delta p^{n+1} + \Delta p_{cog}^s - \gamma_g^s \Delta D\right)_f \\ &+ q_i^{n+1} + \tau_{iomf} + \tau_{igmf} - \frac{V}{\Delta t}\left(N_i^{n+1} - N_i^n\right)_f = 0 \qquad i = 1,\ldots,n_c \qquad \text{(A.6.4)}\end{aligned}$$

$$\begin{aligned}\psi_{n_c+1,f} &= \Delta T_{wf}^s \left(\Delta p^{n+1} - \Delta p_{cwo}^s - \gamma_w^s \Delta D\right)_f \\ &+ q_w^{n+1} + \tau_{wmf} - \frac{V}{\Delta t}\left(N_{n_c+1}^{n+1} - N_{n_c+1}^n\right)_f = 0 \qquad \text{(A.6.5)}\end{aligned}$$

$$\psi_{pf} = \sum_{i=1}^{n_c+1} N_{if}^{n+1} - \phi_f^{n+1}\left(\rho_o S_o + \rho_g S_g + \rho_w S_w\right)_f^{n+1} = 0 \qquad \text{(A.6.6)}$$

where N_i, i = 1, ..., n_c+1 are the moles of i per unit of gridblock volume. The subscript i with i = 1, ..., n_c corresponds to the hydrocarbon components and the subscript n_c+1 denotes the water component. τ_{iomf} and τ_{igmf} are the matrix-fracture transfer in the oil and gas phases for component i, and τ_{wmf} is the matrix-fracture transfer for water. The superscripts n and n+1 denote respectively the old and current time level and the superscript s refers to n for explicit blocks and to n+1 for implicit blocks. The subscript f and m correspond to the fracture and matrix respectively.

Dual-Permeability Formulation

This formulation is similar to the dual-porosity formulation, except that matrix blocks are connected to one another and thus provide alternate channels for fluid flow. The fracture

equations are the same as those in dual-porosity formulation. The matrix flow equations contain additional terms as follows:

$$\psi_{im} = \Delta T^s_{om} y^s_{iom} \left(\Delta p^{n+1} - \gamma^s_o \Delta D\right)_m$$

$$+ \Delta T^s_{gm} y^s_{igm} \left(\Delta p^{n+1} + \Delta p^s_{cog} - \gamma^s_g \Delta D\right)_m$$

$$- \tau_{iomf} - \tau_{igmf} - \frac{V}{\Delta t}\left(N^{n+1}_i - N^n_i\right)_m = 0 \qquad i = 1,\ldots,n_c \qquad \text{(A.6.7)}$$

$$\psi_{n_c+1,m} = \Delta T^s_{wm} \left(\Delta p^{n+1} - \Delta p^s_{cwo} - \gamma^s_w \Delta D\right)_m$$

$$- \tau_{wmf} - \frac{V}{\Delta t}\left(N^{n+1}_{n_c+1} - N^n_{n_c+1}\right)_m = 0 \qquad \text{(A.6.8)}$$

Matrix - Fracture Transfer, NTRAN = 0

Following Kazemi et al (1976).

$$\tau_{jmf} = \sigma V \frac{k_{rj}\rho_j}{\mu_j}\left(p_{jm} - p_{jf}\right) \qquad j = o, g, w \qquad \text{(A.6.9)}$$

where

$$p_g = p_o + p_{cog}$$

$$p_w = p_o - p_{cwo}$$

$$\sigma = 4\left(\frac{kx}{l_x^2} + \frac{ky}{l_y^2} + \frac{kz}{l_z^2}\right)$$

with V being the gridblock volume and l_x, l_y and l_z the fracture spacings in the x, y and z directions respectively. The matrix and fracture blocks are assumed to be at the same depth, no gravity term is included in Equations (A.6.9) explicitly. This equation is used with NTRAN = 0 in CARD RES4.

Generally, the matrix capillary pressure is much larger than fracture capillary pressures and Equations (A.6.9) cannot model correctly the gravity drainage processes.

Matrix - Fracture Transfer, NTRAN = 1

The gravity effects can be included by assuming complete gravity segregation of the oil, gas and water phases as follows:

$$\tau_{omf} = \sigma V \frac{k_{ro}\rho_o}{\mu_o}\left(p_{om} - p_{of}\right) \qquad \text{(A.6.10)}$$

$$\tau_{gmf} = \sigma V \frac{k_{rg}\rho_g}{\mu_g}\left[p_{gm} - p_{gf} + \left(\frac{S_g}{1 - S_{org} - S_{wr}} - \frac{1}{2}\right)_m \Delta\gamma_{ogm} h\right.$$

$$-\left(\frac{S_g}{1-S_{org}-S_{wr}}-\frac{1}{2}\right)_f \Delta\gamma_{ogf}\, h\Bigg] \qquad \text{(A.6.11)}$$

$$\tau_{wmf} = \sigma V \frac{k_{rw}\rho_w}{\mu_w}\Bigg[p_{wm} - p_{wf} - \left(\frac{1}{2}-\frac{S_w - S_{wr}}{1-S_{orw}-S_{wr}}\right)_m \Delta\gamma_{wom}\, h$$

$$+\left(\frac{1}{2}-\frac{S_w - S_{wr}}{1-S_{orw}-S_{wr}}\right)_f \Delta\gamma_{wof}\, h\Bigg] \qquad \text{(A.6.12)}$$

where

$\Delta\gamma_{og}$ = (oil mass density - gas mass density) g

$\Delta\gamma_{wo}$ = (water mass density - oil mass density) g

h = height of matrix element in the direction of gravity

and subscripts

f = fracture property

m = matrix property

Note that normally $\Delta\gamma_m = \Delta\gamma_f$ and the additional gravity terms just accounts for the differences in the height of the fluid columns. Evaluations of Equations (A.6.10) through (A.6.12) are simple and very little additional work is required. They will be used when NTRAN = 1 in CARD RES4.

Matrix - Fracture Transfer, NTRAN = 2

In deriving Equation (A.6.10 through A.6.12), the effects of capillary pressures and gravity are considered separately and later summed together. A more rigorous approach is to use the pseudo capillary pressures. Assuming instantaneous vertical gravity - capillary equilibrium, pseudo capillary pressures can be derived as follows:

$$\int_{\tilde{p}_{cog}-\gamma_{og}h/2}^{\tilde{p}_{cog}+\gamma_{og}h/2} S_g\, dp_{cog} - \gamma_{og}\, h S_g = 0 \qquad \text{(A.6.13)}$$

$$\int_{\tilde{p}_{cwo}-\gamma_{wo}h/2}^{\tilde{p}_{cwo}+\gamma_{wo}h/2} S_w\, dp_{cwo} - \gamma_{wo}\, h S_w = 0 \qquad \text{(A.6.14)}$$

where $\tilde{p}_{cog}$ and $\tilde{p}_{cwo}$ are the pseudo p_{cog} and pseudo p_{cwo} respectively which accounts for both the capillary and gravity effects.

$$\tau_{omf} = \sigma V \frac{k_{ro}\rho_o}{\mu_o}\left(p_{om} - p_{of}\right) \qquad \text{(A.6.15)}$$

$$\tau_{gmf} = \sigma V \frac{k_{rg}\rho_g}{\mu_g}\left(p_{om} + \tilde{p}_{cog,m} - p_{of} - \tilde{p}_{cog,f}\right) \qquad \text{(A.6.16)}$$

$$\tau_{wmf} = \sigma V \frac{k_{rw}\rho_w}{\mu_w}\left(p_{om} - \tilde{p}_{cwo,m} - p_{of} + \tilde{p}_{cwo,f}\right) \qquad \text{(A.6.17)}$$

The pseudo capillary pressures are functions of fluid saturations, densities as well as shape of rock capillary pressure curves. They change dynamically with the change in fluid saturations and densities. As these pseudo capillary pressures appear in the limit of integrals, they are evaluated iteratively and can be time-consuming. Generally, Equations (A.6.13) through (A.6.17) gives results similar to those in (A.6.10) through (A.6.12). In fact, the latter could be derived from the former by assuming negligible rock capillary pressures. Equations (A.6.13) through (A.6.17) will be used when NTRAN = 2 in CARD RES4.

Matrix - Fracture Transfer, NTRAN = 3

All earlier equations assumes that transfer of all phases occurs through the entire surfaces of the matrix element. This is not true when the matrix element is partially immersed in either gas or water. To account for this effect, Equations (A.6.15) through (A.6.17) were modified as follows:

$$\tau_{omf} = \sigma V \frac{k_{ro}\rho_o}{\mu_o}(p_{om} - p_{of}) \qquad \text{(A.6.18)}$$

$$\tau_{gmf} = \sigma V \frac{k_{rg}\rho_g}{\mu_g}\left\{(p_{om} - p_{of}) + \left[S_{gm} + \frac{\sigma_z}{\sigma}\left(\frac{1}{2} - S_{gm}\right)\right]\left(\tilde{p}_{cog,m} - \tilde{p}_{cog,f}\right)\right\} \qquad \text{(A.6.19)}$$

$$\tau_{wmf} = \sigma V \frac{k_{rw}\rho_w}{\mu_w}\left\{(p_{om} - p_{of}) - (p_{cwo,m} - p_{cwo,f})\right.$$

$$\left. - \left(\frac{1}{2}\frac{\sigma_z}{\sigma}\right)\left[\left(\tilde{p}_{cwo,m} - \tilde{p}_{cwo,f}\right) - (p_{cwo,m} - p_{cwo,f})\right]\right\} \qquad \text{(A.6.20)}$$

The form of these equations was found to give reasonable results. They will be used when NTRAN = 3 in CARD RES4. Similarly, the evaluations of the pseudo capillary pressures requires iterations and can be time-consuming.

Appendix B

Figure B-1: Numbering of the Grid System

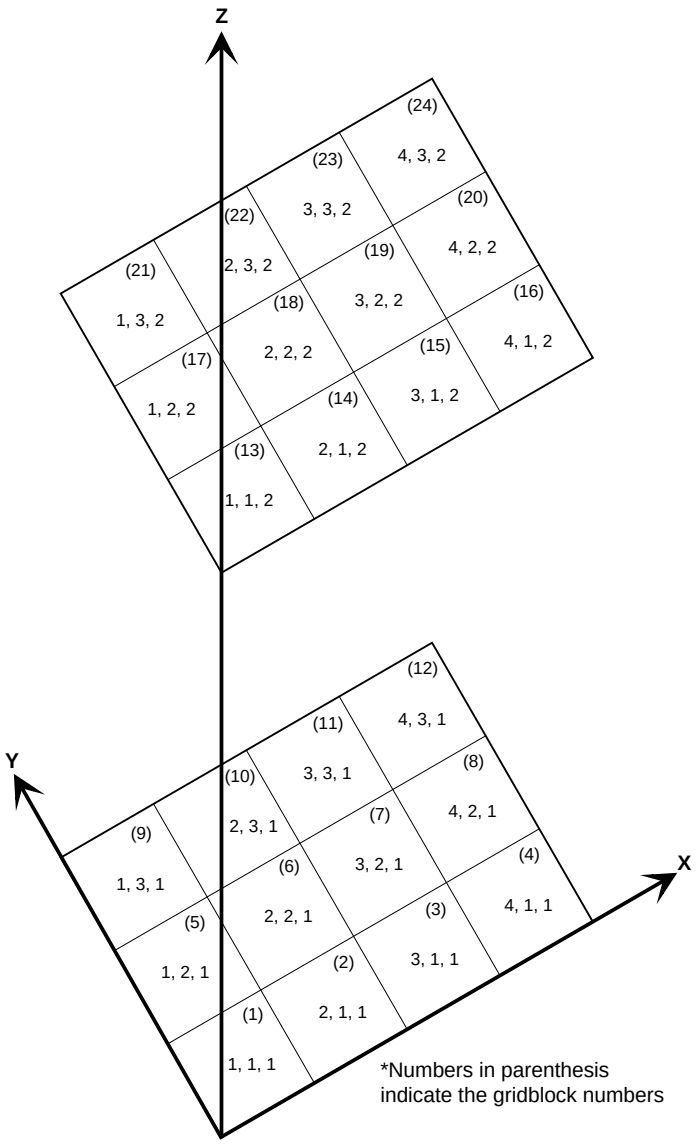


Figure B-2: Cylindrical Reservoir

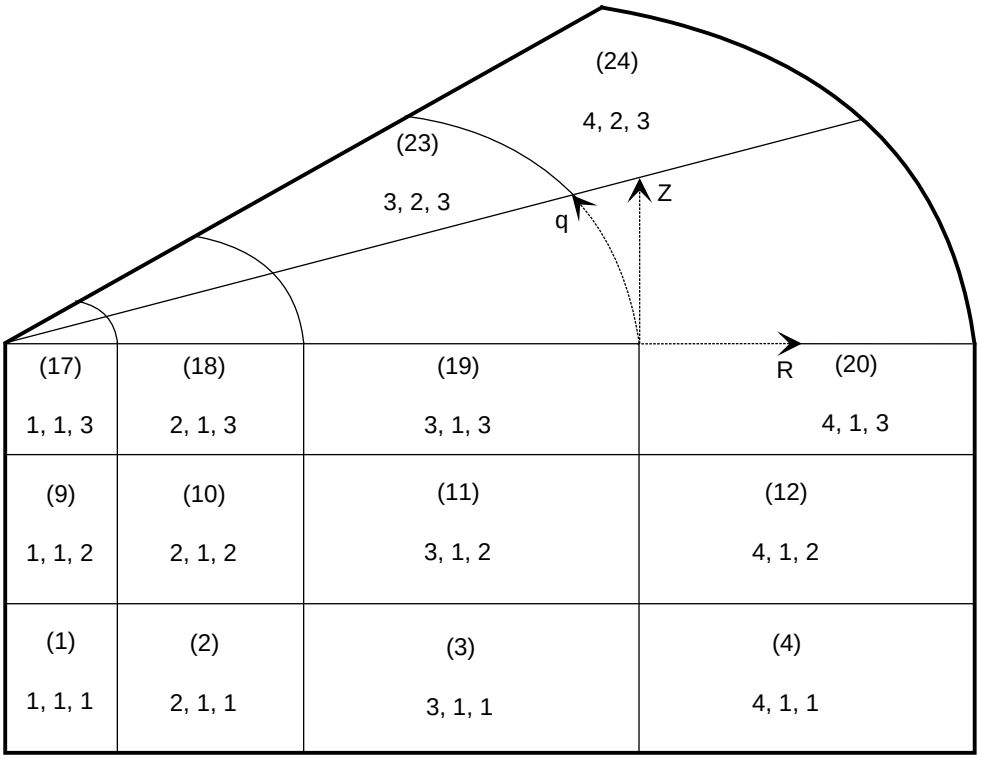
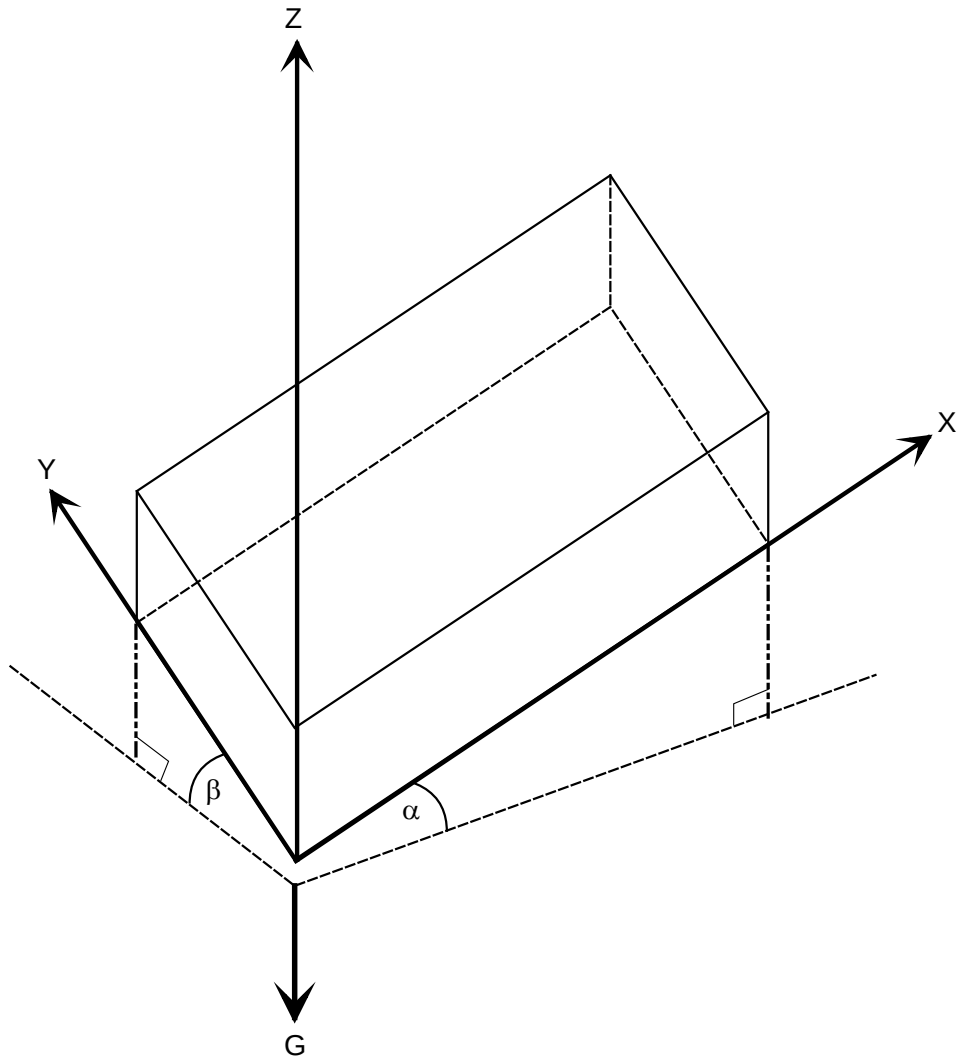


Figure B-3: Calculation of Gravity Components for a Tilted Grid System



$$\alpha = \text{dip in x} - \text{direction}$$

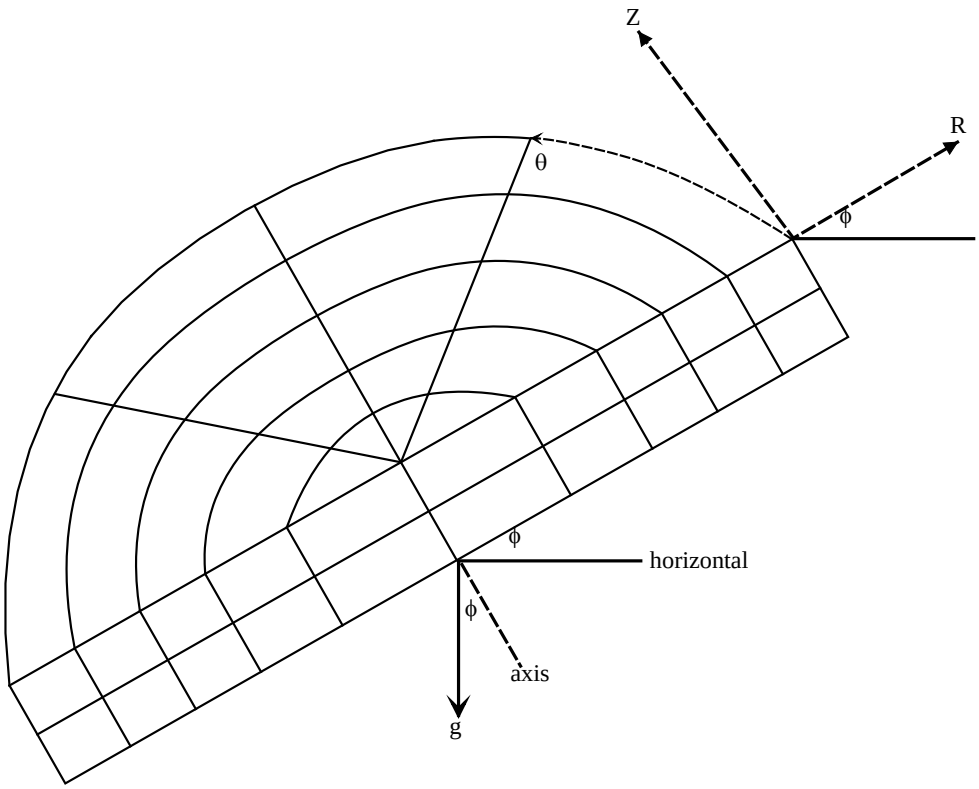
$$\beta = \text{dip in y} - \text{direction}$$

$$GRX = -\sin\alpha$$

$$GRY = -\sin\beta$$

$$GRZ = -\sqrt{1 - \sin^2\alpha - \sin^2\beta}$$

Figure B-4: Radial (Cylindrical) Coordinates



$$\theta = 0$$

$$GR = -\sin\phi$$

$$GZ = -\sqrt{1-\sin^2\phi}$$

Figure B-5a: 2-Dimensional Variable Thickness Grid System (8x1x3)

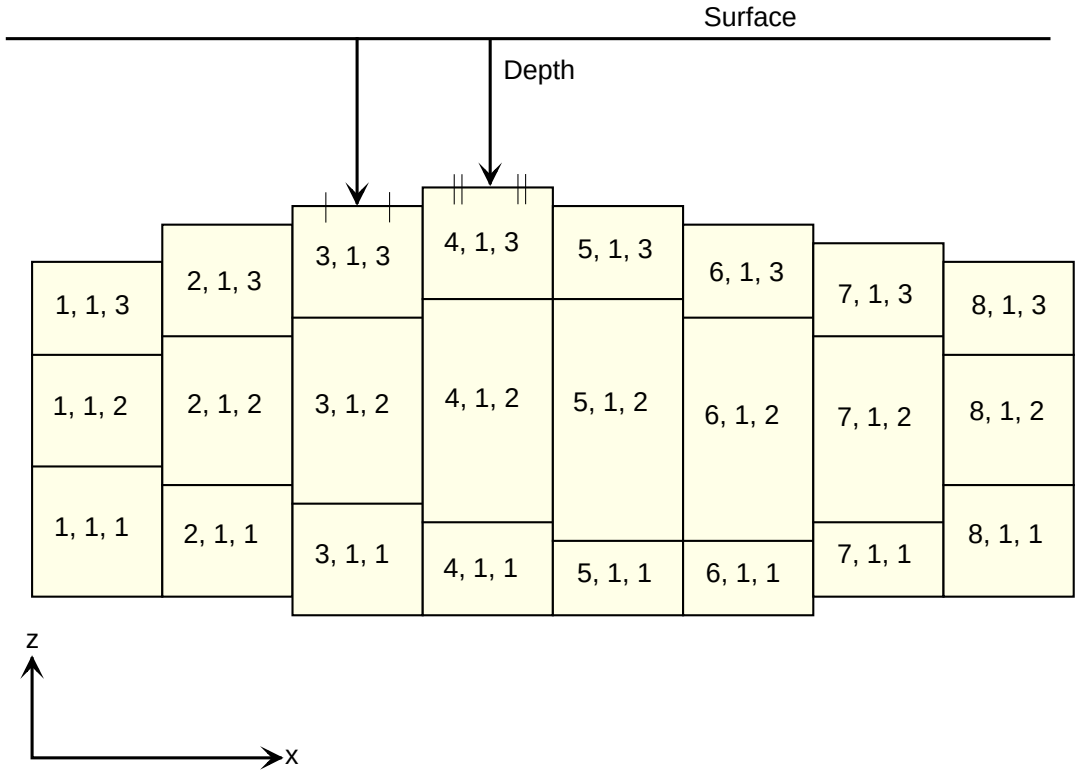


Figure B-5b: 3-Dimensional Variable Thickness Grid System (8x3x2)

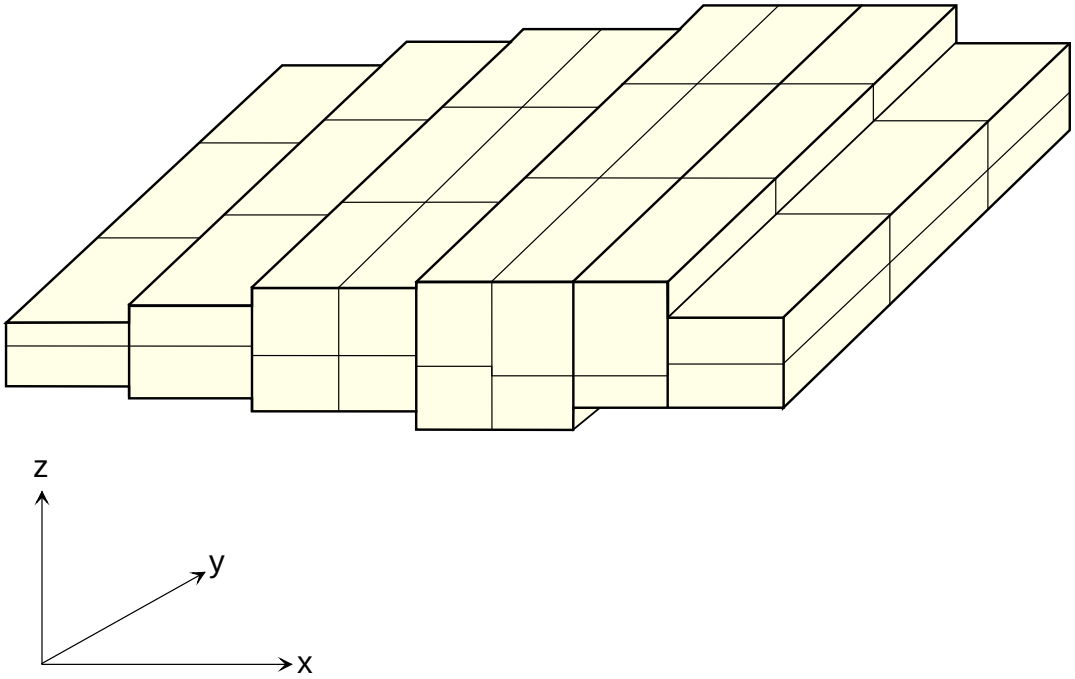


Figure B-6: Discretized Reservoir with Faults

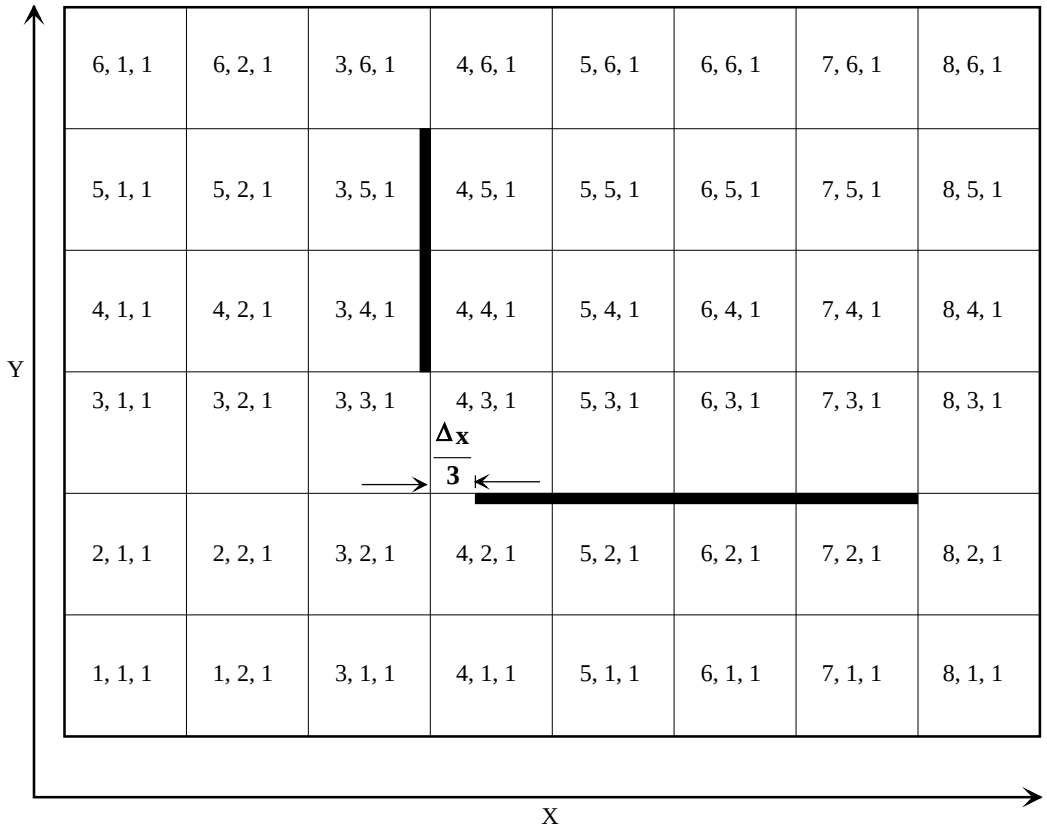
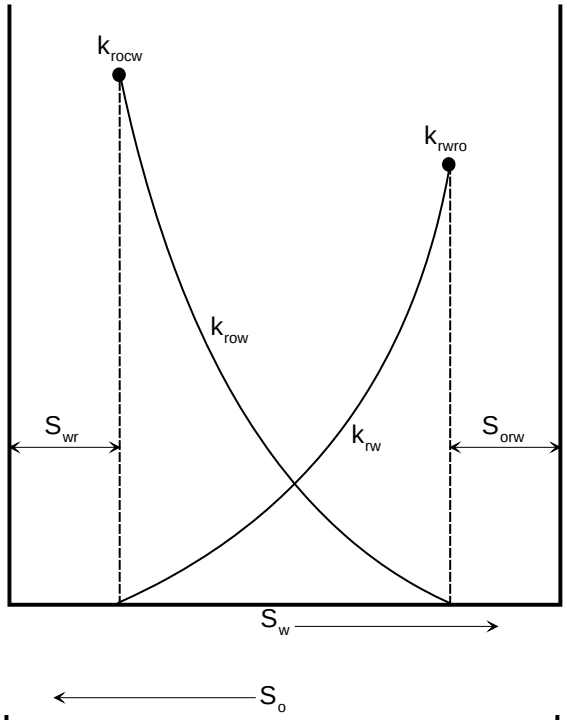


Figure B-7: Relative Permeability Curves

Water-Oil Relative Permeability Curves



Gas-Liquid Relative Permeability Curves at Connate Water Saturation

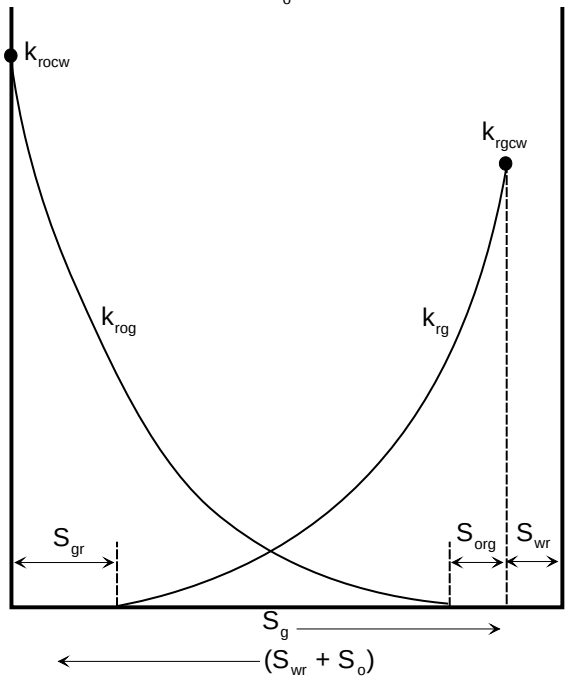
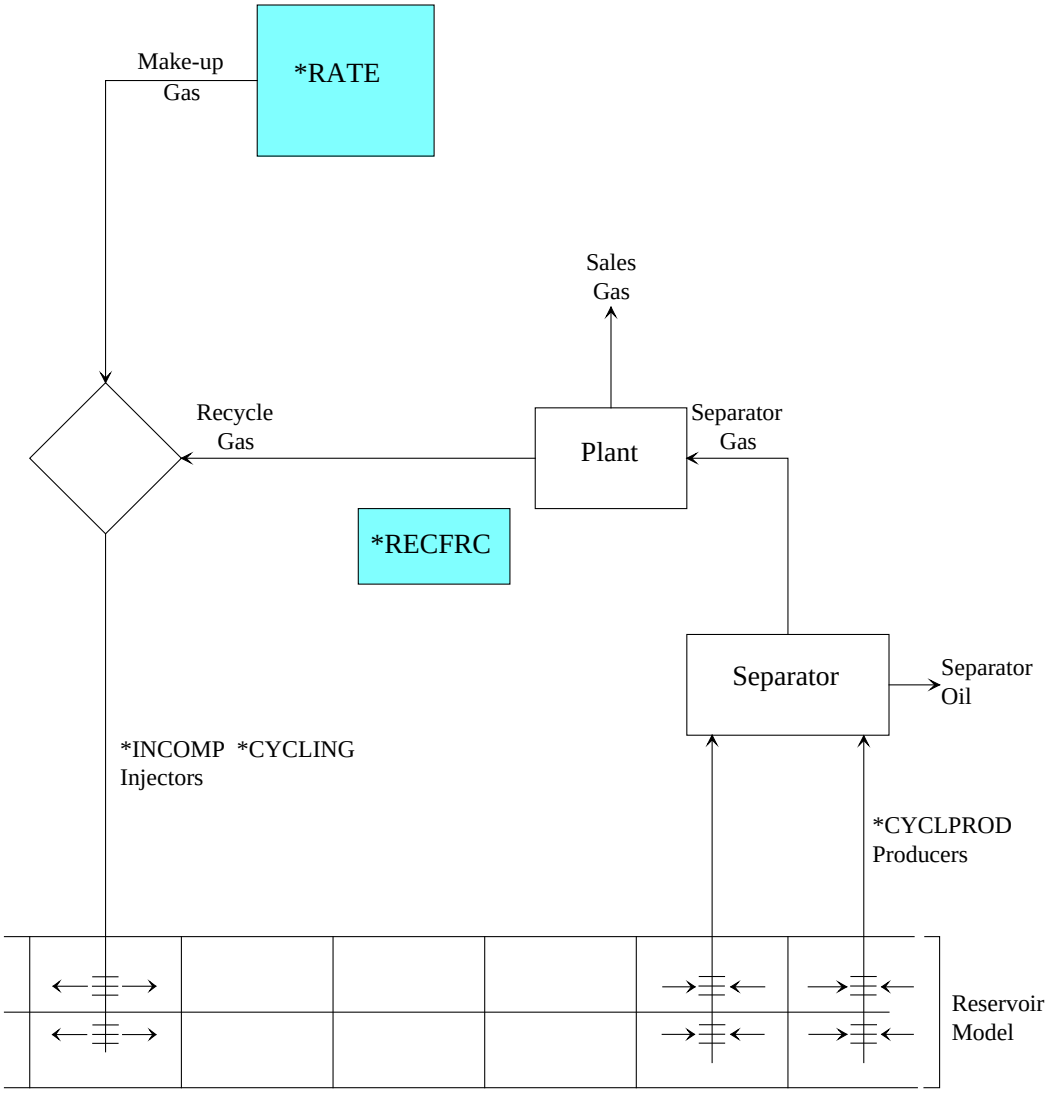


Figure B-8: Recycling Option in GEM



Appendix C

References

Aavatsmark, I., Barkve, T., and Mannseth, T., "Control-Volume Discretization Methods for 3D Quadrilateral Grids in Inhomogeneous, Anisotropic Reservoirs", SPE Paper 38000, presented at the 1997 SPE Reservoir Simulation Symposium, Dallas, 8-11, June, 1997.

Arri, L.E., Yee, D., Morgan, W.D. and Jeansonne, M.W.: "Modeling Coalbed Methane Production With Binary Gas Sorption," Paper SPE 24363, *Proceedings* SPE Rocky Regional Meeting, Casper, Wyoming, 18-21 May, 1992.

Aziz, K., and Settari, A., Petroleum Reservoir Simulation, Applied Science Publishers, London, 1979.

Baker, L. E., "Three-phase relative permeability correlations", SPE/DOE Paper 17369, presented at the 6-th Symposium on Enhanced Oil Recovery, Tulsa, Oklahoma, April 17-20, 1988.

Chappelear, J.E., and Williamson, A.S., "Representing Wells in Numerical Reservoir Simulation: Part 2 - Implementation," Soc. Petrol. Eng. J., Vol. 21, June 1981, pp. 339-344.

Evans, E.V. and Evans, R.D.: "Influence of an Immobile or Mobile Saturation on Non-Darcy Compressible Flow of Real Gases in Propped Fractures," *J. Pet. Tech.* (October 1988), pp.1343-1351.

Frederick Jr., D.C. and Graves, R.M.: "New Correlations To Predict Non-Darcy Flow Coefficients at Immobile and Mobile water Saturation," Paper SPE 28451, *Proceedings* 69th Annual Technical Conference and Exhibition, New Orleans, Louisiana, 25-28 September 1994.

Geertsma, J.: "Estimating the Coefficient of Inertial Resistance in Fluid Flow Through Porous Media," *Soc. Pet. Eng. J.* (October 1974), 445-450.

Gosset, Heyen, and Kalitventzeff, "An Efficient Algorithm to Solve Cubic Equations of State," Fluid Phase Equil., Vol. 25, 1986, pp. 51-64.

Hall, F.E., Zhou, C., Gasem, K.A.M., Robinson Jr., R.L. and Yee, D.: "Adsorption of Pure Methane, Nitrogen, and Carbon Dioxide and Their Binary Mixtures on Wet Fruitland Coal," Paper SPE 19194, *Proceedings* 1994 Eastern Regional Conference & Exhibition, Carleston, West Virginia, USA, 8-10 November 1994.

Harvey, A.H.: "Semiempirical Correlation for Henry's Constants over Large Temperature Ranges," AIChE J., Vol. 42, No. 5 (May 1996), pp. 1491-1494.

Katz, D.L., and Firoozabadi, A., "Predicting Phase Behavior of Condensate/Crude-Oil System Using Methane Interaction Coefficients," J. Petrol. Technol., Vol. 30, November 1978, pp. 1649-1655.

Kazemi, H., Vestal, C.F., and Deane Shank, G., "An Efficient Multicomponent Numerical Simulator," Soc. Petrol. Eng. J., Vol. 18, 1978, pp. 355-368.

Lake, L.W.: *Enhanced Oil Recovery* (1989), Prentice-Hall, Englewood Cliffs, New Jersey.

Lee, B.I., and Kesler, M.G., "A Generalized Thermodynamic Correlation Based on Three-Parameter Corresponding States," AIChE J., Vol. 21, May 1975, pp. 510-527.

Ma, H. and Ruth, D.: “Physical Explanations of Non-Darcy Effects for Fluid Flow in Porous Media,” *SPEFE* (March 1997), 13-18.

Nghiem, L.X., Fong, D.K., and Aziz, K., "Compositional Modeling With an Equation of State," Soc. Petrol. Eng. J., Vol. 21, December 1981, pp. 687-698.

Nghiem, L.X., and Heidemann, R.A., "General Acceleration Procedure for Multiphase Flash Calculations With Application to Oil-Gas-Water Systems," Proceedings 2nd European Symposium on Enhanced Oil Recovery, November, 1982, Editions Technip, Paris, France, pp. 303-316.

Nghiem, L.X., "A New Approach to Quasi-Newton Method With Application to Compositional Modelling," paper SPE 12242, presented at the Seventh SPE Symposium on Reservoir Simulation, San Francisco, California, November 16-18, 1983.

Nghiem, L.X., and Li, Y.-K., "Computation of Multiphase Equilibrium Phenomena With an Equation of State," Fluid Phase Equilibria, Vol. 17, 1984, pp. 77-95.

Nghiem, L.X., and Li, Y.-K., "Approximate Flash Calculations for Equation-of-State Compositional Models," paper SPE 13517, presented at the Eighth SPE Symposium on Reservoir Simulation, Dallas, Texas, February 10-13, 1985.

Peaceman, D.W., "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation," Soc. Petrol. Eng. J., Vol. 18, 1978, pp. 183-194.

Peaceman, D.W., "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability," Soc. Petrol. Eng. J., Vol. 23, December 1983, pp. 531-543.

Pedrosa, O. A., and Aziz, K., "Use of hybrid grids in reservoir simulation," SPERE, November, 1986, pp. 611-621.

Peneloux, A., Rauzy, E., and Freze, R., "A Consistent Correction for Redlich-Kwong-Soave Volumes," Fluid Phase Equilibria, Vol. 8, 1982, pp. 7-23.

Peng, D.Y., and Robinson, D.B., "A New Two-Constant Equation of State," Ind. Eng. Chem. Eng. Fundam., Vol. 15, 1976, pp. 59-64.

Perry, R.H., Chilton, C.H., and Kirkpatrick, S.D., editors, Chemical Engineer's Handbook, 4th Edition, McGraw-Hill, New York, 1977, pp. 3-201.

Reid, R.C., Frautsnitz, J.M., and Sherwood, T.K., The Properties of Gases and Liquids, 3rd Edition, McGraw-Hill, New York, 1977.

Riazi, M.R., and Daubert, T.E., "Simplify Property Predictions," Hydrocarbon Processing, March 1980, pp. 115-116.

Rubin, B. and Blunt, M., "Higher-Order Implicit Flux Limiting Scheme for Black Oil Simulation", SPE Paper 21222, presented at the 11th SPE Symposium on Reservoir Simulation held in Anaheim, Ca., Feb. 17-20, 1991.

Sandler, S.I., Chemical and Engineering Thermodynamics, John Wiley & Sons, New York, 1977.

Soave, G., "Equilibrium Constants from a Modified Redlich-Kwong Equation of State," Chem. Eng. Sci., Vol. 27, 1972, pp. 1197-1203.

Sigmund, P.M.: “Prediction of Molecular Diffusion at Reservoir Conditions. Part II – Estimating the Effects of Molecular Diffusion and Convective Mixing in Multicomponent Systems,” *J. Can. Petrol. Technol.* (July-September 1976) 53-62.

Twu, C.H., "An Internally Consistent Correlation for Predicting the Critical Properties and Molecular Weight of Petroleum and Coal-Tar Liquids," Fluid Phase Equil., Vol. 16, 1984, pp. 137-150

Warren, J. E. and Root, P. J., "The Behaviour of Naturally Fractured Reservoirs," SPEJ, Sept. 1963, pp. 245-255 (Trans. AIME, 234).

Wilke, C.R. and Chang, P.: “Correlation of Diffusion Coefficients in Dilute Solution,” *AIChE J* (June 1955), 264-270.

Appendix D

Coupling GEM with Surface Network Models

GEM (2005.15 and later) can be linked to Petroleum Experts Ltd. (Petex)'s GAP or SPT Group's FORGAS surface network model. The methodology adopted in coupling the two surface network models is quite similar. While coupling to GAP is made through the RESOLVE framework, coupling to FORGAS is handled by FORGAS calculation engine itself. CMG provides a GemGapLink dll to facilitate linking between GEM and RESOLVE/GAP. This dll is not required for coupling between GEM and FORGAS.

GEM and the surface network model have their own input and output files. GEM and the network model can have wells that are shared or linked between the two. At the same time each model can have its own well(s) which are not shared with the other model.

Before connections are made, the individual cases have to be modeled satisfactorily in their separate application. Thus a GEM data set which is the case to be loaded into the network model must be able to run without the network model. All necessary sources and sinks (wells) should be defined appropriately.

When a simulation is run, the controlling interface (RESOLVE or FORGAS) collects and passes data between all the connected applications. The data passed in these cases is an IPR table (Inflow Performance Relationship - a table of bottom-hole flowing pressures against phase rates, and compositions). In the case of FORGAS certain additional information like mobility weighted well block pressures, separator pressures and temperatures are also passed from the simulator.

The communication between the simulator and surface network model is made through signal and data files that are in ASCII format. The simulator and the network model have their own signal and data files which are automatically created and continuously updated by the coupled system. These files normally reside in the directory where other input and output files are located and should not be deleted by the user while the coupled system is running.

General Considerations and Guidelines

Connections between the reservoir and the surface network are made at well flowing bottom-hole conditions. Thus the network model must handle flow in the wellbore.

Some or all GEM wells can be linked to the network model through the controlling interface (RESOLVE or FORGAS). GEM generates IPR tables for those wells that are linked to the network model and in turn, gets instructions from the controlling interface (RESOLVE or FORGAS) on how to control those wells.

In general, GEM wells linked to the network model have their operating constraints and schedules described in the network model. On the other hand, wells within the GEM model that are not linked to network model have their operations controlled by GEM.

After release 2006.12, GEM can also be made to control certain aspects of a linked well's operating condition.

Specifically for linked wells:

1. Any well monitor defined via the *MONITOR keyword is allowed.
2. Keywords that control well and layer operations within a *TRIGGER are allowed.

Details are given in the 'Recurrent data' section of 'Preparation of GEM data set' section below.

Group production and injection controls (*GCONP,*GCONI, or *GCONM) are not permitted for groups containing a mix of linked and unlinked wells. However, *GCONM could be exercised on groups consisting of all the wells that are linked to the surface network.

GEM is connected to the network model on a per well basis, using IPR curves.

For RESOLVE, VFP tables must be defined in network using a program like PROSPER or any other utility. In order to use the linked model, the network model method for determining optimum rates must be set in RESOLVE to VFP/IPR curve intersection.

GEM has the capability to pass information related to multiple EOS sets. The EOS set for a well is the same as one assigned to the grid-block associated with the well's reference layer. The information contains EOS-type (PR/SRK), component names, critical temperatures, critical pressures, critical volumes, accentric factors, molecular weights, parachors, omega-As, omega-Bs, specific gravities, true boiling points, volume shifts, and binary interaction coefficients, and specific gravity of water.

Both the surface network models (GAP and FORGAS) have certain limitations in handling the EOS parameters. For instance, GAP and FORGAS do not use the volume shifts, and FORGAS, at this time, does not handle the binary interaction coefficients. The user is advised to read the manual pages for the respective network model to understand these limitations.

There should be no GEM well control keywords for network model linked wells besides the initial well definition (see Preparation of GEM Data Set).

Normally the network model (through its interface, RESOLVE or FORGAS) controls the timestep the coupled model takes; however, GEM can be made to provide input to RESOLVE/FORGAS about timestep selection. RESOLVE/FORGAS will then select the minimum of the network model and GEM timesteps.

If the time to the next network model specified network solve date is too large, it is possible that a GEM well's bottom-hole pressure may vary considerably while timestepping to this date. Because GEM no longer controls its own well constraints for network model linked wells, a failsafe was implemented to limit a well's minimum pressure to 101.325 kPa (14.7 psi). The use of this feature can be avoided by reducing the network model maximum timestep size and/or by allowing the network model to use its own adaptive timestep selector.

The GEM start date defined in the recurrent data set should be equal to or later than the network model forecast start date. If the reservoir started production before the desired forecast start date, run GEM alone to create a restart record on the desired forecast start date.

Use that restart record for the runs using the combined models. If the GEM prediction start date is later than the network model forecast start date, wells in the linked model will be masked out until the GEM forecast start date is reached.

A linked model may be started from a restart created by a linked or unlinked model. For instance, a surface network linked model can run a prediction starting from the end of a reservoir history match simulation.

Preparation of the GEM Data Set

There are no keywords required for GEM to couple with the network model. But as listed below, some restrictions and changes apply.

- The EOS set of the reservoir and surface network should be identical. In other words, at reservoir temperature the surface network EOS should match the EOS in GEM.
- The forecast start date of GEM can not be earlier than the start date of the network model time schedule. The GEM start date can be a GEM restart start date.

Recurrent Data

The following well data restrictions only apply to GEM-Network model linked wells.

Well Definition

All GEM-Network model linked wells should be defined once at the start time of recurrent data. No further well keywords are necessary in the GEM recurrent data after the start date. Wells which cycle between injectors and producers should be defined as two wells in both the network and GEM models.

Well SHUT/OPEN Status

GEM passes open/shut-in status for linked wells to the surface network at each surface network date. After receiving this well status information from GEM, the surface network honors those wells GEM which shut in and may add other shut-in wells based on its schedule or optimization procedure.

The surface network then runs a new network timestep and passes the updated well status back to GEM. GEM retains the received well status for the GEM timesteps required to reach the new network date.

GEM can attempt to open a well (using a monitor or a well control in a trigger), but if the surface network schedule or its optimization overrides this, the well remains shut-in until the next network date, where it will be checked again by the surface network, and if the network schedule or optimization allows the well to open, it will open. This checking occurs at every surface network date.

So either GEM or the surface network can shut-in a well. However, both GEM and the surface network must agree that a well should be open before it actually opens.

Well Constraints

For producers, a non-zero surface rate constraint should be given. For example,

```
*OPERATE *MAX *STO  500
```

indicates the target stream of the producer is oil and the specified rate is ignored. Similarly, non-zero STG, STW, or STL can be used to indicate a gas, a water, or a liquid producer (if applicable) respectively.

If more than one of the STO, STG, STW, and STL constraints are defined, the first or primary phase will be used to determine the GEM constraint from the three phase rates received from surface network. If the first or primary constraint happens to be other than a rate constraint, GEM will use its internal logic to determine the phase for which the well constraint would apply. Within the recurrent data, the keyword *OPERATE with a non-zero rate can be used as the primary constraint to change the target stream phase from one phase to another. The *OPERATE keyword for linked wells for phase change should normally be placed within a trigger to avoid rate mismatch between GEM and the surface network.

For injectors, the injection stream is given by the keyword *INCOMP.

Bottom-hole pressure constraints, if applicable, should be defined in the surface network. A BHP constraint given in GEM is ignored for rate calculation purposes. However, for injectors, if the keyword:

```
OPERATE *MAX *BHP  1000
```

is given, the value of 1000 will be used only to limit the maximum pressure of the IPR table.

The violation action *SHUTIN within the *OPERATE keyword should be avoided. For example,

```
*OPERATE *MIN *BHP  1000  *SHUTIN
```

for a producer may act at any GEM timestep and cause rate mismatch between GEM and the surface network. Such conditional operations may be performed instead by the monitor or trigger keywords.

Injectors must be of the default mobility weighted type.

Monitors and Triggers

GEM and the surface network ONLY exchange information at each network timestep. When linked to RESOLVE, it is critical that the well and well layer status is not changed between surface network dates. FORGAS automatically uses ALL of the GEM dates and thus the date restrictions described in this section apply only when GEM is coupled to RESOLVE.

a) Monitor

For linked wells, monitors are checked at surface network dates ONLY. For unlinked wells, monitors are checked at each GEM timestep.

For well groups defined in GEM, if a group contains all the wells as linked wells, the applicable group monitor option is limited to the group monitor (GCONM) and the group monitor will only be checked at surface network dates. On the other hand, if a group does not include linked wells, group monitors can be applied as usual. An error status occurs when a group contains a mix of both linked and unlinked wells.

b) Trigger

Keywords that directly change the well status or that modify productivity for linked wells (i.e. not in triggers) should be avoided. For example, if well 2 is a linked well and the date “2000 1 1” is not a surface-network date. The inputs,

```
*DATE  2000 01 01
*SHUTIN  2
```

will shut in well 2 between two surface network dates, but cause a rate mismatch between GEM and the surface network. This inconsistency should be avoided by using triggers as discussed below.

Triggers are the means by which the user can ensure that well controls are only applied when GEM and GAP are synchronized. The user must place all well control keywords within triggers because in linked runs only trigger keywords checked at surface network dates.

In the example below we redefine the layers for well no. 2, then multiply its original productivity index by five at the first surface network DATE after “2000 1 1”

```
*DATE  2000  1  1
*TRIGGER    'trg_well2'    *ON_ELAPSED    'TIME'    treltd  >  0.0
    *GEOMETRY    *K    0.07    0.37    1.0    0.0
    *PERFV    *GEO    2
    ** KF    FF
    2:3    1.0
*SETPI *MULTO 2
    5.0
*END_TRIGGER
```

The use of this “elapsed time” trigger ensures that the keywords within the trigger will be read at the first possible RESOLVE date following “2000 1 1”

The use of the “*ON_ELAPSED ‘TIME” treltd > 0.0” trigger would normally force the trigger to fire immediately, but as this is a linked model, the trigger is checked at the next surface network date.

Keywords *OPEN and *SHUTIN for wells should be included in triggers. For example,

```
*DATE  2000  1  1
*TRIGGER    'shut 2'    *ON_ELAPSED    'TIME'    treltd  >  0.0
    *SHUTIN    'PRODUCER-2'
*END_TRIGGER
```

The use of this trigger above will shut-in 'PRODUCER-2’ at the first RESOLVE date following “2000 1 1”.

Of course triggers can still be used to control when a well is shut in based on a real criteria, such as in the example below, where a well is shut in when its well bottom-hole pressure is less than 1000 psia.

```
*TRIGGER    'trg_bhp2'    *ON_WELL    'PRODUCER-2'    BHP  <  1000.0
    *SHUTIN    'PRODUCER-2'
*END_TRIGGER
```

The trigger above will be executed at the surface network date when the bottom-hole pressure of PRODUCER-2 is lower than 1000 psia.

GEM does not exchange well layer status information with the surface network. However, similar to well status, changes for well layer status should be specified only on surface network dates.

It is important to note that for a GEM run coupled with surface network, trigger action will occur only at a surface network date irrespective of whether the well in question is linked or unlinked.

ON-TIME Fraction

For linked wells, at each RESOLVE date, the GAP down-time percentage is passed to GEM as an on-time fraction. GEM uses the received on-time fraction as if the same value was input through the GEM recurrent data. Essentially the definition/handling of on-time fraction for coupled wells is moved to the GAP network.

For unlinked wells, on-time fraction should be given in GEM recurrent data as usual.

GEM passes back on-time fraction for all wells to RESOLVE, so that it is accessible via a RESOLVE script and readable in the RESOLVE reporting window.

DATE and TIME

*DATE and *TIME keywords can normally be used within the GEM data set. These can be used to obtain GEM output when the *WPRN/*WSRF output options are used.

The forecast start date of GEM cannot be earlier than the forecast start date of the surface network date schedule. Use a restart record to adjust the GEM forecast start date to correspond to that of the surface network model.

When coupled with RESOLVE, the stop date in the GEM data set should be later than or equal to the stop date specified in the network schedule. FORGAS will honor the GEM stop date; the earliest stop date (between that specified in FORGAS and that specified in GEM) will be used.

Multiple Reservoirs

- The surface network model can connect to multiple GEM reservoir models. For RESOLVE these GEM models may have different start dates. However, one of the models should have a start time equal to the start date of the network model. FORGAS requires all GEM files to have same start date.
- In RESOLVE, when multiple instances of GEM are set up on the network model interface, each instance should be given a different name. For example, the default instance of GEM is GEM; the second instance may be named GEM2. It is only necessary that the two names are different; each instance may point to the same executable.
- GEM and the network programs communicate through ASCII files. Four ASCII files per GEM instance are created and updated during the linked run. The files are named using the root GEM input data file name and the extension related to their use within the linked model (LSResolve/LSForgas, LDResolve/LDForgas, LSGem, and LDGem). These files must not be altered by the user during a linked model run.

Coupling with GAP

RESOLVE is a tool written by Petex to allow connectivity between arbitrary petroleum engineering packages. It is a high-level framework that allows engineering data to be passed between applications. When a case of a client application such as GEM is loaded into RESOLVE, RESOLVE will create a slave instance of that application. RESOLVE will then query the case for its sources and sinks – these are the inputs and outputs from the system. In the case of GEM, for example, wells can be sources or sinks depending on whether they are production or injection wells.

Once the case is loaded into RESOLVE, equivalent sources and sinks can be connected together graphically. For example, wells in a GEM reservoir can be connected to their equivalent wells in GAP (Petex's surface network analysis package).

When a simulation is run RESOLVE will perform the following steps:

1. Reload and initialize all cases
2. Broadcast to all clients a request to calculate inflows at all sources (GEM)
3. Pass the inflow tables to the connected sinks
4. Broadcast a solve command to all connected modules
5. Pass the optimized results back to the objects that generated the inflow relation (GEM)
6. Broadcast to all clients a request to perform a timestep of the requested length (GEM)
7. Return to step 2

Since Gem requires and generates EOS compositional data, it is necessary that there is a 'continuity' in the compositional data throughout the GEM-GAP coupled system. It implies that composition passed to RESOLVE by an instance of GEM must map exactly to the composition required by the connected network program say, an instance of GAP. Although component names can differ, there must nevertheless be a one-to-one correspondence between each component passed between each application. This is achieved by establishing component connections through a composition table provided by the RESOLVE interface.

Using Resolve Scripts

- With the 2006.12 and later versions of GEM, a portion of the GEM simulation results are available for scripting. This allows RESOLVE to access variables other than IPR tables and can be used by GAP to control the network. A Resolve script can be written to do calculations using results from both GEM and GAP.
- All of the wells, groups and sectors defined in GEM pass results back to RESOLVE where the well and group rates are average values between two synchronized dates. For accessible GEM results and their scripting format, please see the 'GEM Variables Accessible from Resolve Script' towards the end of this Appendix.
- Accessible GEM results are also reported in the Resolve Reporting window.
- A script is normally designed to execute pre- or post- solution of a specified instance (for example, before the network is solved or after the network is solved).

Voidage Replacement

- Resolve is equipped with a wizard that generates a script to do voidage replacement in a reservoir model. GEM provides the necessary simulation results to let the script work out the amount to inject from the GAP surface network. Wells can be grouped as a voidage replacement unit so as to support pattern replacement. Groups in a voidage replacement script do not have to be defined in GEM (as well groups) or in GAP. Please refer the Resolve manual for instructions in how to use the wizard.
- Generated script can be modified for various purposes and scenarios. For example, it may become necessary to carry out voidage replacement with a reservoir model which does not have all the wells linked in GAP. In this case, the automatically generated script will only include linked wells. However, the script can be modified (hand edited) to access the results for all wells in the reservoir model.
- Care should be taken when naming instances. As mentioned before, the scripting is designed to execute pre- or post-solution of a specified named instance. For example, if two GAP instances are given the names 'GAP' and 'GAP-1', a script for instance 'GAP-1' will also be executed for 'GAP' as the name 'GAP' is contained within 'GAP-1'. This can be avoided by giving the names 'GAP-1' and 'GAP-2' to the two GAP instances.

Resolve Reporting

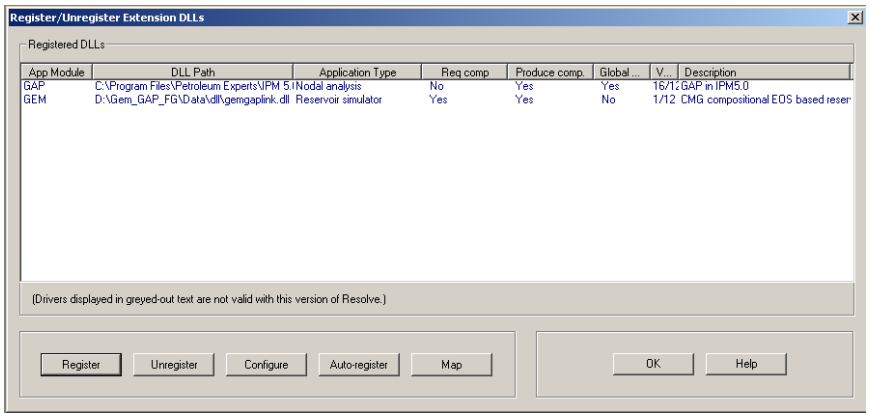
The Resolve reporting window contains GEM result data at each synchronized date. Different from GAP reporting, the GEM reporting value at a corresponding RESOLVE date represents the result for the passed timestep, that is, from the previous synchronized date to current synchronized date.

The user may notice that under the simulator instance name, for example "GEM", there are well aggregate values. The same reporting entries can also be found under GEM group names. The top group name, "FIELD" for example, includes all GEM wells. The difference between well values reported under "FIELD" and "GEM" is that the instance name "GEM" includes the aggregate for only linked wells while "FIELD" includes all GEM wells.

Getting Started

a) Register the GemGapLink.dll as a driver for RESOLVE

   Start RESOLVE and then choose the 'Register Driver' from 'Drivers'. In the window similar to following, click the 'Register' button to assign the location of GemGapLink.dll. This step only needs to be done initially or when you need to update drivers as RESOLVE remembers the location of the registered driver.



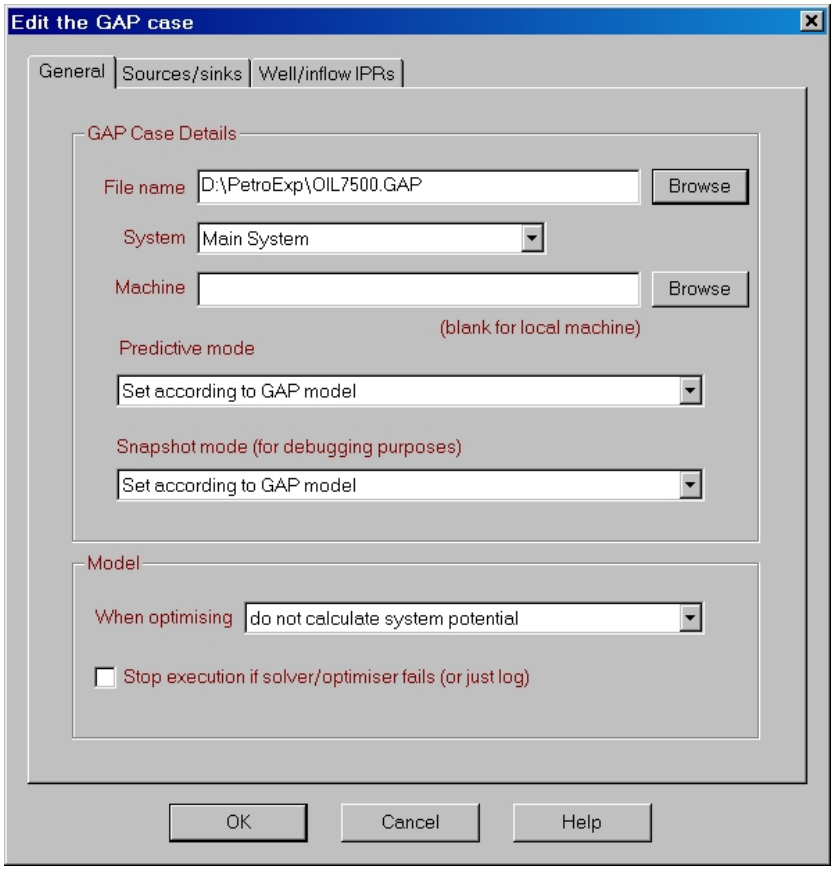
Click ‘OK’ to finish.

b) Open a new RESOLVE Graphic View

Click the ‘File’ => ‘New’ to start a new graphic view in RESOLVE.

c) Add the GAP model into RESOLVE

From ‘Edit System’ => ‘Add Client Program’, choose ‘GAP’. Then add the GAP icon onto the RESOLVE view. Double click the GAP icon. In the edit window, give the location of *.GAP file.



d) Add the GEM model into RESOLVE

Similarly, to add a GEM model, from ‘Edit System’ => ‘Add Client Program’, choose ‘GEM’ and add it onto the RESOLVE view. Edit the GEM Case setting by double clicking the GEM icon. Give the location of both the GEM executable and the GEM data set. The “Timestep selection for network solution” option provides the option for GEM feedback in determining the next timestep size. If feedback is enabled, the timestep size used by RESOLVE will not be greater than the timestep selected by GEM for the next step.

Advanced simulator options allow the user to create a simulation log file instead of seeing the GEM run progress in a DOS window. It also allows user to run the simulator in multi-processor mode. If required, additional command line switches could be used to input various command line arguments available for the simulator (See Command-Line Arguments under Input/Output Control section in GEM User Guide).

e) GEM Case Settings (see graphic below)

Local or remote job submission is possible using the “Computer” combo box. By default, “<Local Computer>” is selected for running a simulation on a local machine. The user can also select a network computer as the target machine. Using the “Computer” combo box, the user can type the computer name, choose a listed computer or select “<More …>” to browse the network. The targeted remote computer should be able to receive a SSH command to invoke the simulator. Please see “Remote Simulation Job Submission via SSH” in this appendix for explanations.

In all cases, the user needs to give the location of both the GEM executable and the GEM data set. Please note the given paths should be locatable by the target machine.

When the target machine is a remote Windows computer, paths using the Universal Naming Convention (UNC path, e.g. \\SIM-SERVER1\dataset) are expected. The user can use the browser button to find them.

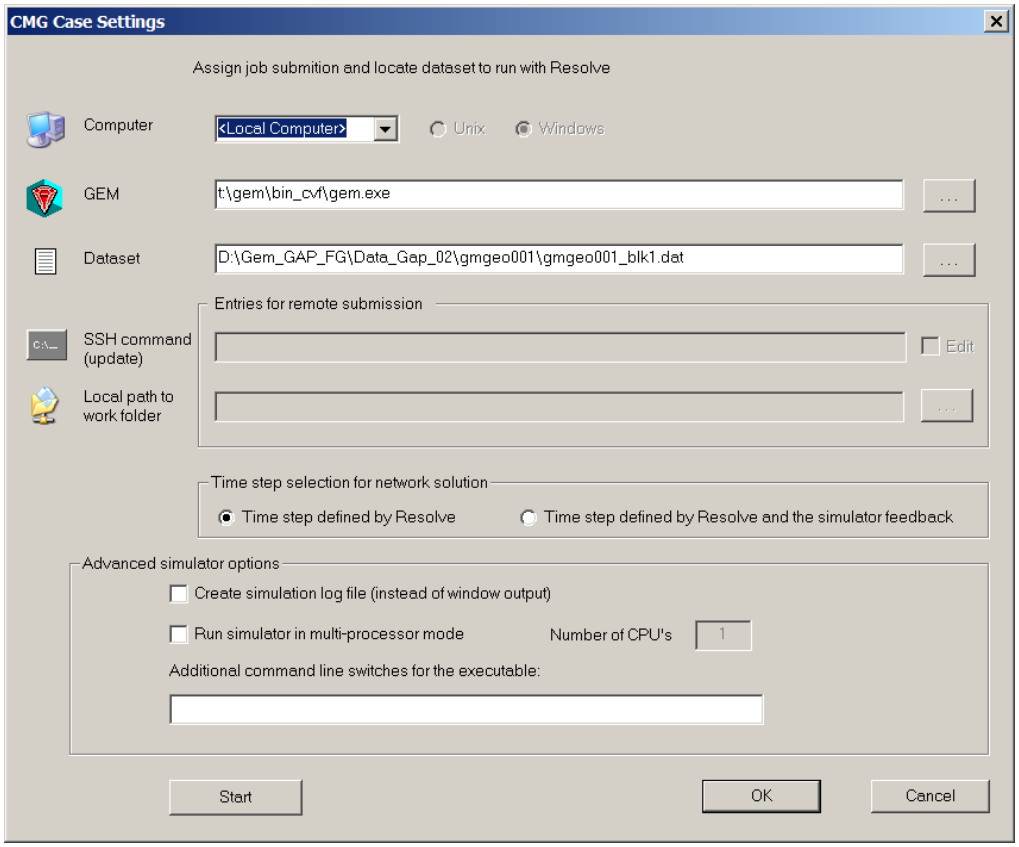
The “Entries for remote job submission” text boxes (SSH command update and Local path to work folder) are only active when a remote computer is selected.

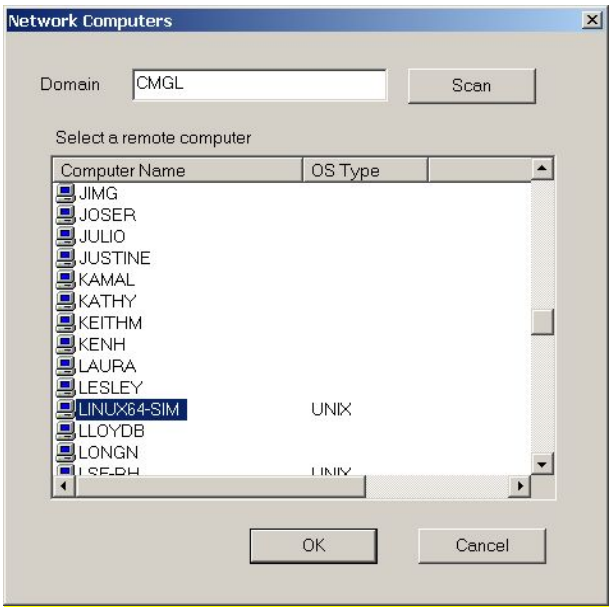
The SSH command button is used to display/update the ssh command. The displayed SSH command is editable. For instance, if the user has a different login name on the server computer than that on the local computer, the command should be edited such that *ssh –l username_on_client* is altered to *ssh –l username_on_server*.

“Local path to work folder” is the local path that points to the folder/directory where the GEM dataset exists. Please see “Location of the work folder” in “Remote Simulation Job Submission via SSH” below for a more detailed description of the entry.

The “Timestep selection for network solution” radio buttons provides the option for GEM feedback in determining the next timestep size. If feedback is enabled, the timestep size used by RESOLVE will not be greater than the timestep selected by GEM for the next step.

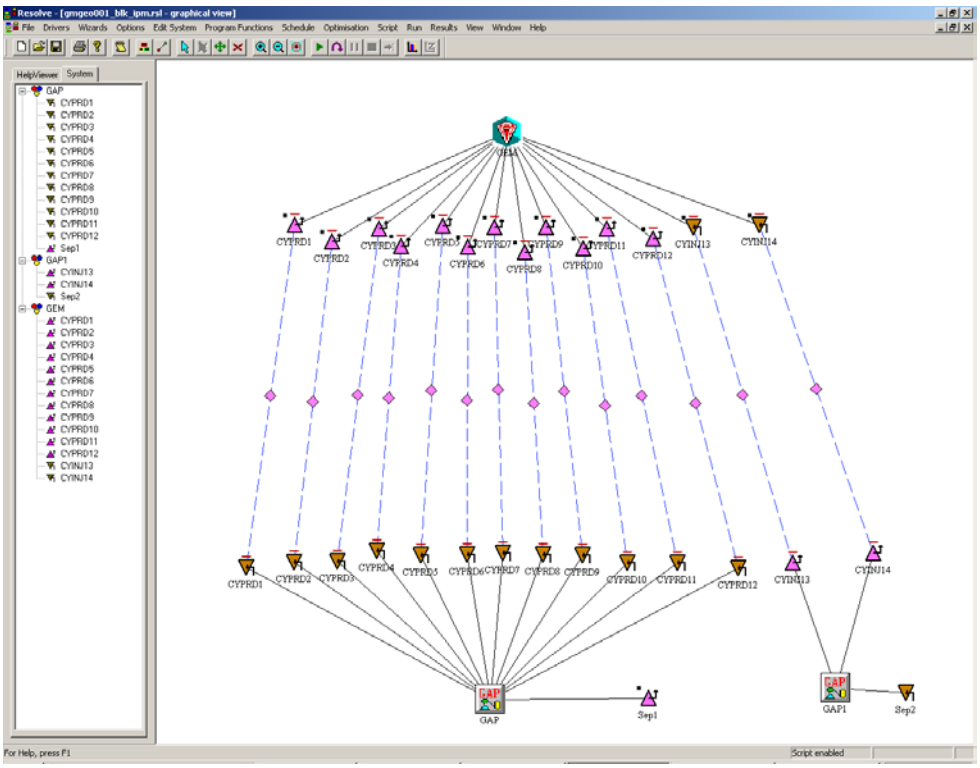
From release GEM-2007.11, a new “Start” button is added to perform the functions of the original “OK” button. “Start” starts the model initialization. After the simulator finishes its initialization, the “Start” button converts to “Stop”, so that user has a chance to stop the simulation at its initialized stage. The current “OK” button is used to only save the current case settings -- it does not run the simulator. However, in case the name of the datafile is changed, the “OK” button becomes unavailable (greys out). The user needs to click the “Start” button to start the simulation and update the well information.





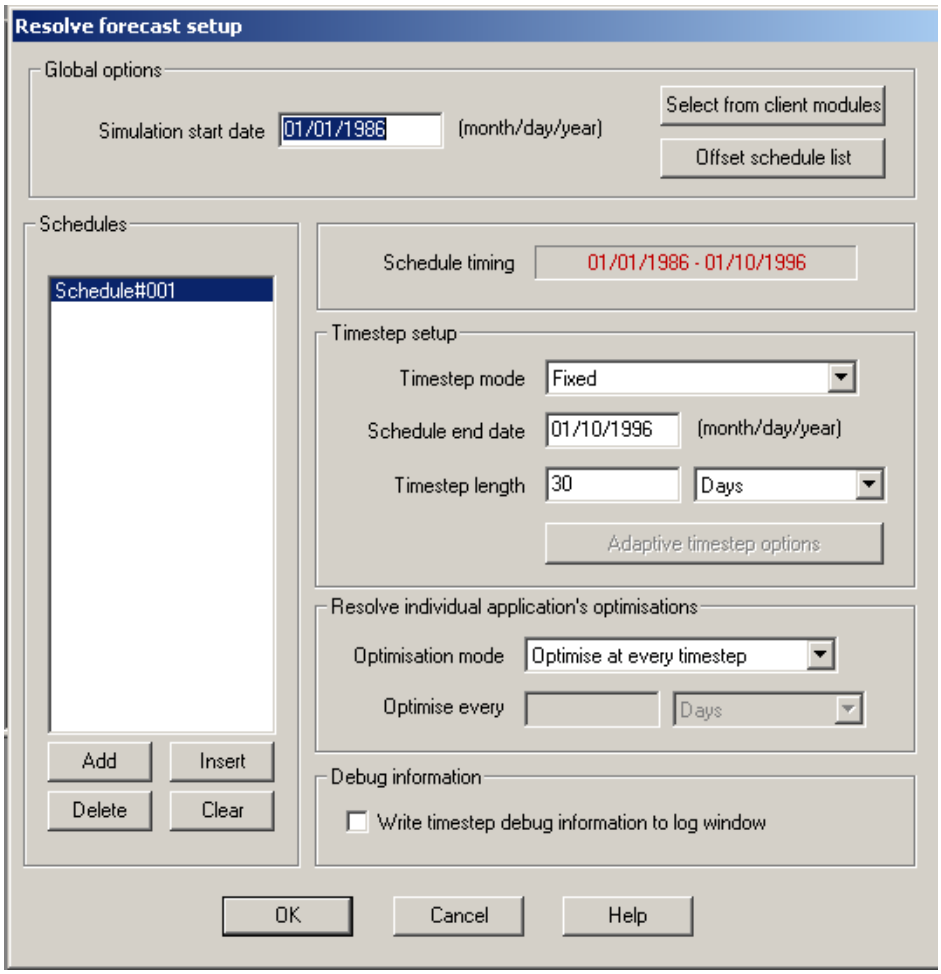
f) Link the wells between GEM and GAP

Use the link function to connect the corresponding wells between GEM and GAP.



g) Set RESOLVE's Schedule

RESOLVE's schedule must be set before starting a run. The timestep mode can be Fixed or Adaptive. Note that if the fixed time-step is chosen and the option allowing GEM time-step selection feedback is on, GEM will still be able to reduce the time-step to values less than the fixed timestep size, if its internal timestep selector determines this is necessary.

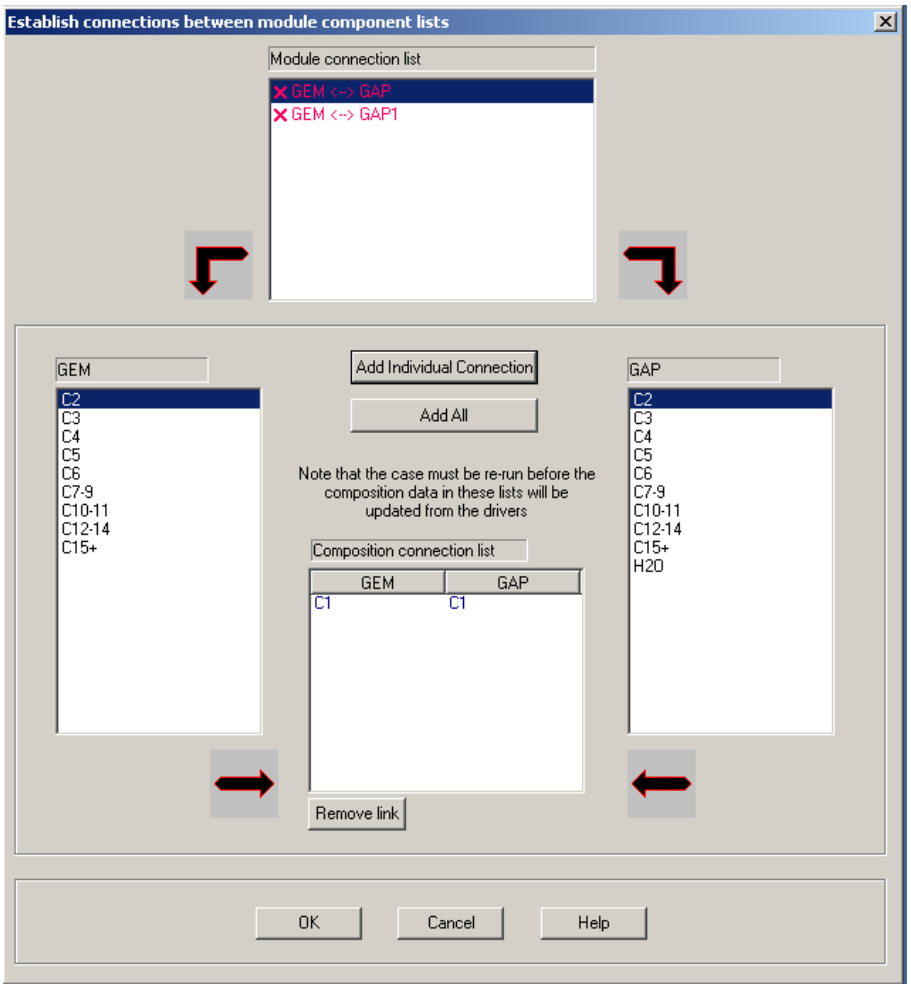


h) Save the RESOLVE file and Run

At this step, the user can start the linked run by clicking the start button on RESOLVE's tool bar. The user can also save the current settings to a RESOLVE file (file extension is .rsl). The .rsl file provides a shortcut to start a case without duplication of the steps from b through f.

After release GEM-2007.11, the saved .rsl file includes GEM well information, so that the link settings can be established by only loading the .rsl file. This is convenient if the user wants to create a new case by loading and then modifying an existing case, without running the later.

i) When a compositional case is run for the first time, RESOLVE queries different instances of GEM for their base composition. It then produces the following screen:



j) From the module connection list select the applications (for instance, GEM-GAP, or GEM-GAP1) that are connected together.

k) Next, from component lists select the component names that are to be linked together. Click ‘Add Individual Component Connection’ or alternatively click ‘Add All’ to make the component links. To remove a component link click on ‘Remove Link’, if required.

l) Click ‘OK’ and RESOLVE will save the component mappings in its (.rsl) file. The screen will not automatically be displayed in subsequent runs unless there is a change or discrepancy. However, the mappings can be edited any time by clicking on ‘Run=>Edit Composition Table’ from the main menu of RESOLVE.

m) Note that a ‘g’ alongside the GEM timestep number on the GEM log file indicates completion of a GAP timestep.

Remote Simulation Job Submission via SSH

The GEM-GAP link DLL (GEM-2007.11 and later) and GEM-FORGAS link is designed to optionally submit simulation jobs to network computers via Secure SHELL (SSH). Here the network refers to the intranet/LAN. The required environment requirements for remote submission are:

a) The targeted remote computer should be a SSH server (running SSH service) and the local computer should have a SSH client program installed. The SSH client issues a ssh command to execute the simulation on a SSH server.

b) Both local and remote computers should be able to create/read/write files in the folder/directory where the simulation dataset exists. GEM and FORGAS or the link DLL exchange data in that "work folder".

SSH Server

Linux and IBM AIX computers are equipped with OpenSSH which provides SSH Daemon (SSHD) as the SSH service.

Windows computers needs to install a third party software to run an SSH service. For all our tests, WinSSHD (http://www.bitvise.com/) was used to provide the SSH service for Windows. Tested Windows platform were Windows Server 2000, Server 2003 and XP 64 bit.

SSH Client

The SSH client must be on Windows, because the local machine needs to run RESOLVE for GAP or FGI for the FORGAS link, both Windows based programs. There are quite a few software vendors who provide SSH clients for Windows at a low cost. We developed/tested using CopSSH (http://www.itefix.no/phpws/index.php) which is an OpenSSH derived Windows version. The client program has been tested on Windows 2000, XP, XP 64 bit and Vista.

Note: After the installation of CopSSH, the user may need to edit the Windows environment variable "path" to add the SSH bin directory in the path, e.g.: C:\Program Files\copSSH\bin.

SSH Settings

There are various approaches that can be used to enable SSH communication. The following description is intended to give an example of one such approach that was employed in our testing process. For technical details, please refer to the SSH software manual.

The ultimate object of the SSH settings is to allow the user to invoke remote simulation by issuing a one line "ssh" command at a local command console. For example, the following command line should directly start the GEM with the dataset, dataset1.dat,

ssh –l *username_on_server* simsvr d:\cmgexe\gem.exe –f d:\cmgdata\dataset1.dat

where the "simsvr" is a remote Windows computer.

In order to do so, the remote computer (SSH server) should be set so as to accept the user login without password. Instead of password authentication, SSH client and server, by default, use key authentication to establish connection.

Generating Key Pair

The user can use ssh-keygen.exe available with CopSSH to generate a private/public key pair. Using ssh-keygen.exe, two key files, viz., id_rsa and id_rsapub, should be generated into the folder .ssh, which is created under the user's local Windows home directory. A blank "passphrase" is suggested when doing key pair generation. If the key pair is made with a non-blank passphrase, the passphrase will be asked for each time SSH attempts to login.

Registering Public Key in the Sever

For the UNIX/OpenSSH system, the user needs to copy the public key (e.g. from file `id_rsapub)` and append it into the file: $HOME/.ssh/authorized_keys.

For the Windows/WinSSHD system, the system administrator needs to start the "WinSSHD Control Panel/ Settings", and in the "Access Control/Windows accounts" tree view add the user and then import the user's public key.

For the first time to configure the installed WinSSHD, system administrator also needs to register the network domain name. That is, under the same "Settings" tree view, choose the "Domain order" and click the "Add" button to add the domain name.

Test SSH

After the above steps are done, the user can test SSH by trying a ssh command from the local command console, such as:

ssh simsvr set

If the above command gives out the user environment variables in "simsvr", SSH has been successfully set up. This command assumes the user's login name is the same on both the server and the local computer.

Please note that the first time a user tries to link to the SSH server, SSH will respond that it can not establish authenticity and will query if the user wants to continue. A 'yes' response is necessary to allow SSH to import the host key, i.e. the server's public key, to the client side.

Settings in Linux/AIX for the CMG Simulator Environment

In order to make sure the SSH login reads the UNIX shell startup file (.cshrc, .kshrc, etc.) which includes the values for `CMG_HOME`, `LSFORCEHOST` and `LD_LIBRARY_PATH`, the following should be done in the OpenSSH server:

a) In user's `$HOME/.ssh` directory, create a file with name `environment`. Add one line, `ENV=.kshrc`, into the file (assuming Kshell is used).

b) Make sure `/etc/ssh/sshd_config` file contains the next line:
`PermitUserEnvironment yes`

SSHD needs to be restarted to ensure the updated `sshd_config` file is read.

These settings ensure the `$HOME/.kshrc` is read when SSH logs in.

If the `$HOME` is a shared directory among UNIX machines and those hosts need different CMG variable settings, the shell startup file can include a `case` command to account for different hosts. For example,

```
simsvr | simsvr1 ) LD_LIBRARY_PATH=/usr/cmg/gem/Linux_x64/lib:\
/opt/intel/fce/9.1.036/lib; LSFORCEHOST=lserv; export\
LD_LIBRARY_PATH  LSFORCEHOST;;
aixsvr | aixsvr1) LSFORCEHOST=lserv; export LSFORCEHOST;;
esac
```

Location of the working folder

It is recommended to have the working folder on the local disk of the machine on which the simulation runs. This simplifies file access and reduces traffic through the network thus providing a better response between GEM and Resolve or between GEM and FORGAS. However, the working folder can also be created on a disk controlled by a third file server, if desired.

A “Local path to work folder” is required in the CMG case setting window, which tells Resolve how to access to the working folder. In the entry, the path should be addressed simply and directly. A path across multiple machines/mount points should be avoided.

Known Problems and Workaround

Samba Server

If the work folder/directory is located on a drive under a UNIX platform, and the user notices an unreasonable delay or even a hanging up on alternating text file communication between GEM and RESOLVE/GAP or GEM/FORGAS, there can be file sharing problems caused by Samba “opportunistic file locking”. The workaround is to switch off the file locking for all CMG-RESOLVE or CMG/FORGAS communication files. That is, in the Samba configuration file (e.g. /etc/samba/smb.conf), add a line for the share where the work folder is located:

```
veto oplock files = \
/*.LSImex/*.LSResolve/*.LSGem/*.LDImex/*.LDResolve/*.LDGem/
```

And restart the Samba daemon to make sure the altered smb.conf is used.

Linux Simulator Execution with Work Folder on a Windows PC (using Samba Client)

When a job is submitted to a Linux platform and the working folder is located on a Windows PC, GEM/Resolve or GEM/FORGAS may lose communication due to Windows file locking. If this occurs, the user needs to check how the “Local path to work folder” is defined. If the path does not start with the server-name of the Linux platform, the user must alter the server-name to the Linux platform name (e.g. \\Linux_CPU\...).

Coupling with FORGAS

The FORGAS calculation engine facilitates exchange of signal and data between GEM and FORGAS in addition to performing various other functionalities required for the running of the network model. The user provides the name of the GEM input file as well as the name and location of the GEM executable in the FORGAS data preparation program, FGI. In the FORGAS input data file, shared FORGAS/GEM wells should belong to a “GEM” reservoir. For each “GEM” reservoir defined in FORGAS, the user selects the name of the GEM input data file and an optional restart file (rst) or an optional index-results file (irf).

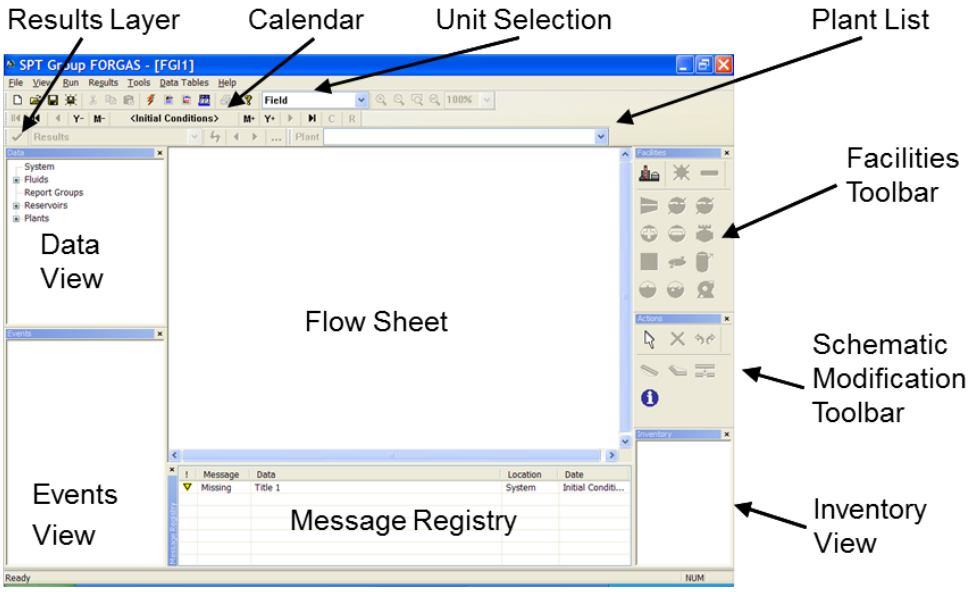
When the user selects to Run FORGAS inside the FORGAS interface, the coupled system will perform the following steps.

1. FORGAS will automatically start the specified GEM executable using the file names of the specified GEM input data files. The GEM log file will appear within a DOS window, which the user can view as the run progresses.
2. FORGAS will wait while GEM initializes its calculations and provides the composition and inflow performance data to FORGAS for each well defined in GEM.
3. FORGAS will read the GEM component definitions passed by GEM and will automatically create those components in FORGAS. FORGAS also checks the properties defined for H2S, CO2, N2, C1, C2 to C6 to determine if they are the same as those for FORGAS library components. If so, those components will be stored as FORGAS library components, to reduce the total number of components and to use the appropriate binary interaction coefficients built into FORGAS. FORGAS automatically eliminates duplicate components.
4. GEM will pass its current date. If this date is earlier than the forecast start date in FORGAS, FORGAS will end the forecast with an error message. GEM will recommend the length for the next timestep (in days). FORGAS will ensure that the first FORGAS timestep will be exactly one day in length, by providing the appropriate timestep information to GEM. Thereafter, the recommended timestep from GEM will be used, unless it needs to be shortened by FORGAS for user-specified changes in the FORGAS input data. The minimum timestep in FORGAS is one day. If the timestep recommended by FORGAS is too long for GEM, GEM will take as many timesteps as required to move forward the required number of days. When there are multiple GEM input data files coupled to FORGAS, FORGAS will look at the recommended timestep from each GEM file, and will use the smallest of each to determine the next timestep to take. FORGAS will pass that timestep length to each GEM file. Repeating FORGAS timesteps is not supported.
5. After the initialization phase, GEM will wait until FORGAS has finished the calculations for the first timestep. For production wells, FORGAS provides the oil, water and gas flow rates at the set point which are interpolated from the GEM IPR tables. For injection wells, the oil, water and gas flow rates are calculated by FORGAS at the outlet separator conditions using the calculated injection well composition. For injection wells, FORGAS provides GEM the hydrocarbon composition of each injection well, using the specified plant and import compositions. For each shared well, the well status (OPEN or SHUTIN) is passed by FORGAS to GEM. The flowing bottom-hole temperature will also be provided to GEM. For production wells, this temperature will be equal to the reservoir temperature. For injection wells, this temperature could be different than the reservoir temperature when temperature is computed in the wellbore. The next FORGAS timestep length (in days) is also provided.

6. Once FORGAS has finished its first timestep, it now waits for GEM to use the data provided by FORGAS to calculate the conditions at the start of the next FORGAS timestep. GEM now reads the data provided by FORGAS to determine which of its wells are shared with FORGAS. GEM performs its timestep(s) to reach the specified FORGAS timestep. GEM calculates the inflow performance and composition of each well and passes that to FORGAS. The current date and the recommended length of the next timestep are passed to FORGAS. Then GEM waits until FORGAS has retrieved and used the data to determine the well flow rates for the next timestep.
7. The forecast will continue until the end date is reached (namely the earliest date of those specified in the FORGAS and GEM input data files) or there is an error encountered by FORGAS or GEM.

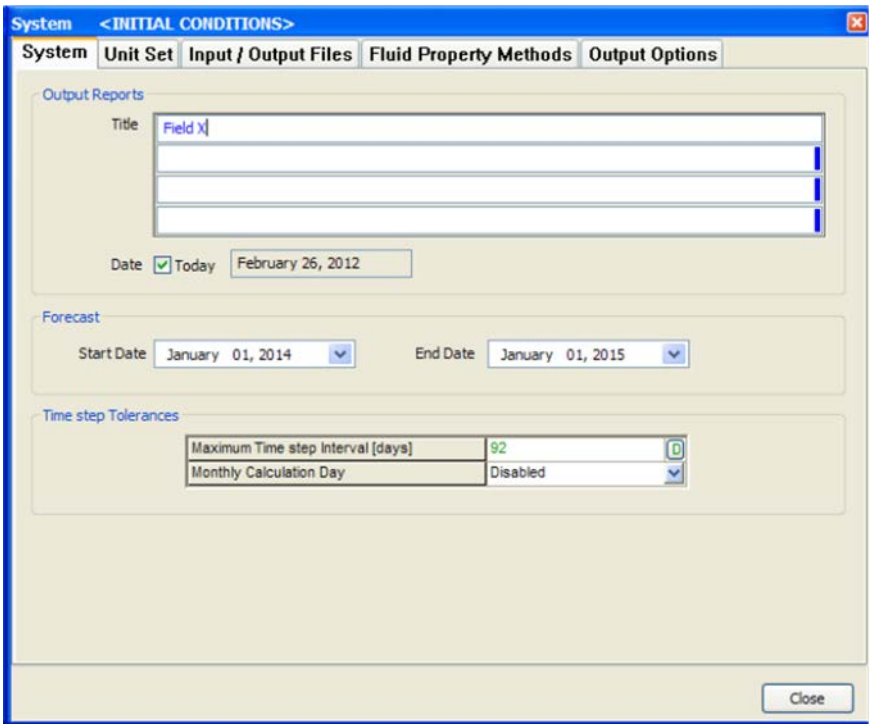
Getting Started

a) Start FORGAS. Note the various areas of the screen as described below.

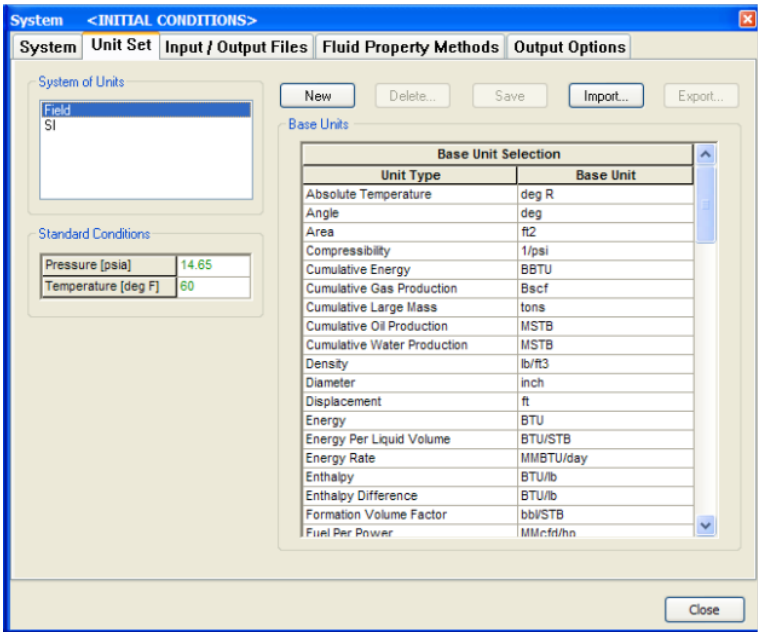


b) Select File => Save As and specify an appropriate file name. FORGAS will automatically append the extension ".fgw".

c) In the Message Registry (usually located at the bottom of the panel), double click on the message about the missing Title 1. This will take you to the System panel. Type in the Title. Specify the appropriate Forecast Start Date. Both FORGAS and GEM must start their forecast on the same date. Use a restart record in GEM to align the start dates. Enter the Forecast End Date. The FORGAS/GEM run will end on the earliest of the FORGAS and GEM end dates. The screen will appear similar to that shown below:



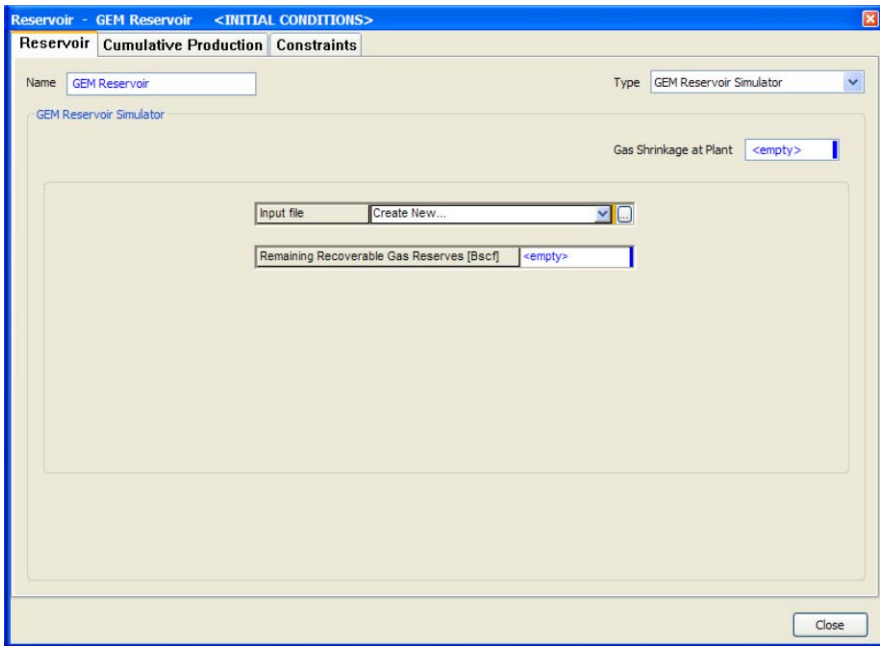
d) Click on the Unit Set tab and select the appropriate unit set, as shown below. It is not necessary to select the same unit set as GEM uses, since the parameters passed between GEM and FORGAS are always converted to imperial (field) units in their communication files.



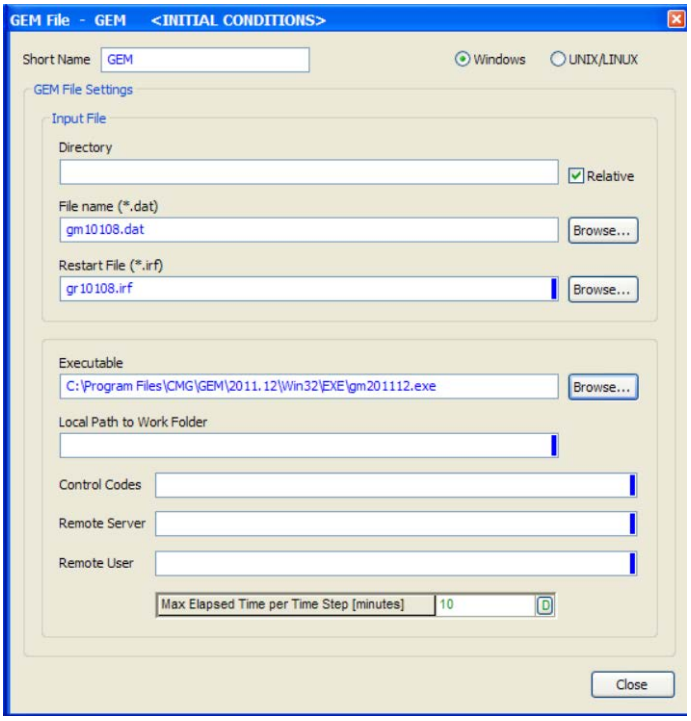
e) If desired, you can select the other tabs to select different options than the provided defaults for file creation, equation of state models and output tables. Close the System panel.

f) In the Data View, click on the + beside the word Reservoirs to open that area and then double click on Create New... to create a new reservoir.

g) Type in the appropriate name for the reservoir. For the reservoir Type, select GEM Reservoir Simulator. The panel should appear as shown below. Values marked with a blue bar are optional values, which are not required to run the case and thus can be left as <empty>. Values marked with a yellow bar are required parameters that must be specified for the forecast to run.

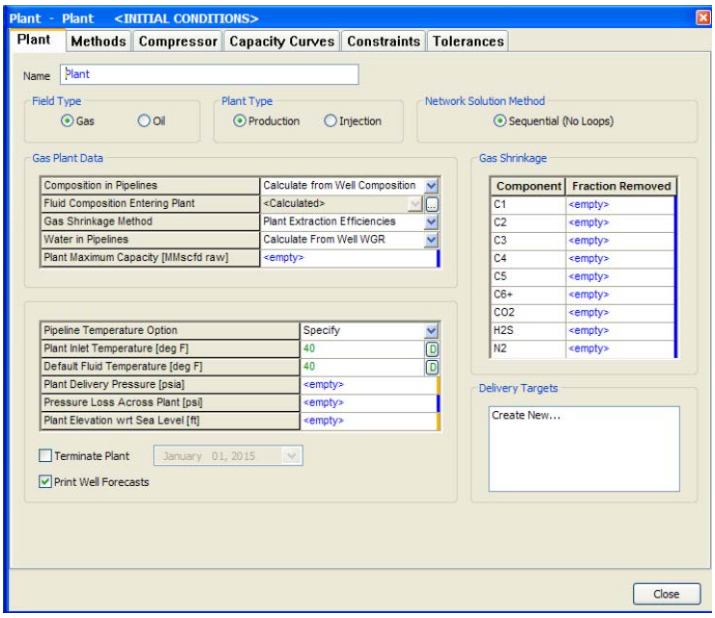


h) Click on the ... after Input file Create New... to specify the GEM input data file name. On the panel which appears, select the GEM input data file. If required to align the start dates with FORGAS, choose the GEM restart .IRF file. Use the Browse button to find the location of the GEM executable to be used. If running the GEM data file on a different computer than being used by FORGAS, specify the appropriate Remote Server information (see Appendix D in the GEM help for assistance). The panel should appear similar to that shown below:



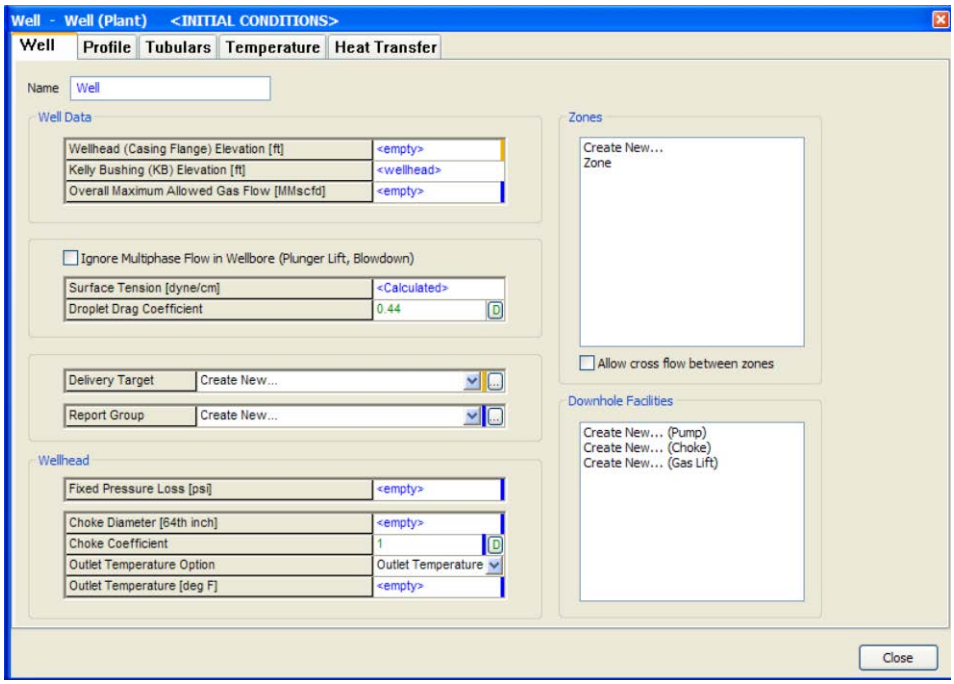
i) Select Close twice to close the reservoir data.

j) A plant is the delivery point at which gas is sold to the buyer. All systems must have one plant. The plant must be put on the Flow Sheet before the other parts of the system (e.g. wells, pipes) can be defined. Click on the plant icon in the Facilities Toolbar. FGI will automatically place the plant at the centre of the Flow Sheet. Double click on the plant icon to enter the plant data (see below).



k) After specifying all of the plant data (select each tab), close the plant data.

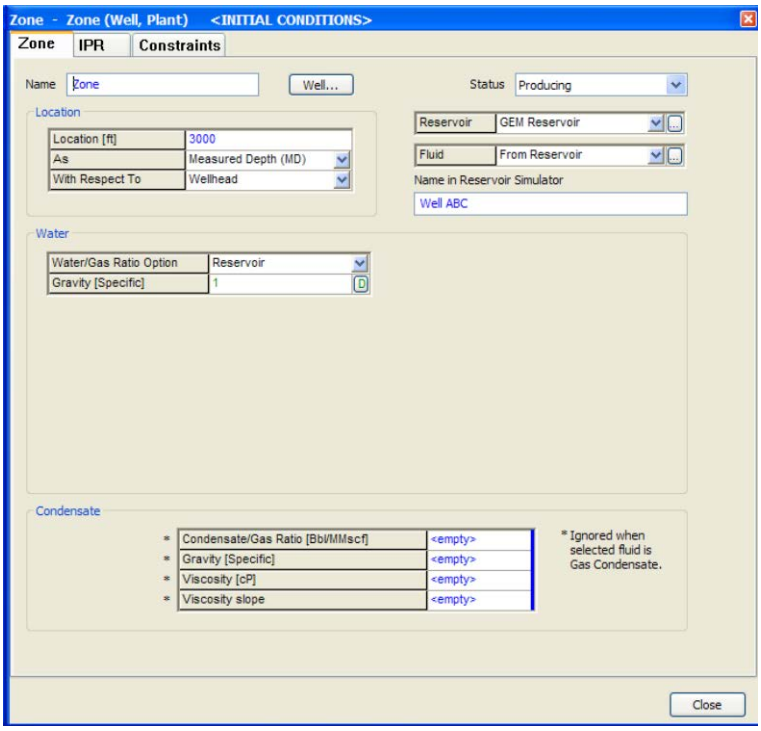
l) Click on the well icon in the Facilities Toolbar and then click on the Flow Sheet to place the well. Double click on the well to specify its data. A panel will appear similar to that shown below:



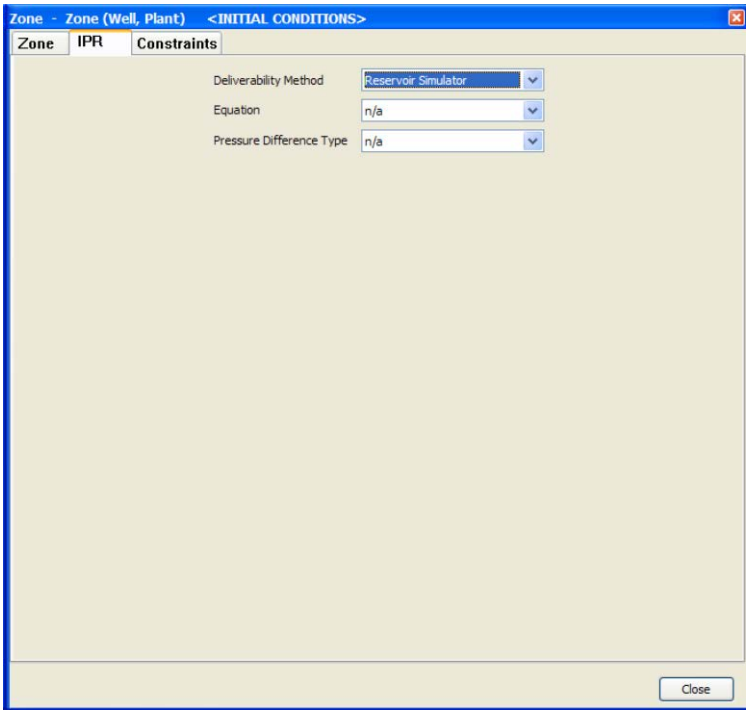
m) Zone is the term which refers to the perforations where the well connects to the reservoir. FORGAS automatically creates one Zone for a well. Double click on the word "Zone" in the Zones area. This will open the Zone data panel.

- Select the zone status. In both the GEM and FORGAS input data files, a well must be given the status of "producer" for FORGAS/GEM to allow the well to produce.
- Specify the location of the perforations.
- Select the appropriate reservoir name for Reservoir.
- The water/gas ratio can either be specified in the FORGAS input data file or obtained from GEM. To use the amount of water predicted for this individual well by GEM, select Reservoir for the Water/Gas Ratio Option.
- The fluid composition will automatically be supplied by GEM, so leave the Fluid selection as "From Reservoir".
- Type in the well name in the GEM input data file for the Name in Reservoir Simulator. It is important that this name exactly matches the well name in the GEM input data file, otherwise the forecast will be terminated.

The panel should appear similar to the following:



n) Select the IPR tab. GEM will always provide the inflow performance data for each well. Select Reservoir Simulator for the Deliverability Method, as shown below:



o) Specify the rest of the required data for each well, each pipeline and each facility.

p) To run the forecast, either click on the run icon or select Run => Current File... The Run Monitor of FORGAS will appear, showing the progress of the run. The GEM log file will appear in a separate DOS window. Note that an “f” alongside the GEM time step number in the GEM log file indicates completion of a FORGAS time step.

q) The results of the run will be available via GEM's output and SR2 files and through the FORGAS Results menu (View Output Files, View Plots..., Export to Spreadsheet...).

r) For more details on using FORGAS coupled with GEM, see the FORGAS Help => User Manual => FORGAS 10: Technical Reference => Reservoir Calculations => GEM Reservoirs.

s) More details on using FORGAS can be found in the FORGAS Help => User Manual => FORGAS 10: Getting Started => Using the FORGAS Graphic Interface (FGI).

GEM Results Accessible from RESOLVE Script

GEM variables that are accessible from Resolve script via DoGet() are described below. All names are case sensitive.

Total Numbers

Well count

instance_name.nWell

Group count

instance_name.nGroup

Sector count

instance_name.nSector

Example:

```
DoGet("GEM.nWell”)
DoGet("GEM.nGroup”)
```

Well Variables

Format:

instance_name.Well[{*well_name*}].*well_variable_name*
or
instance_name.Well[{*#well_number*}]. *well_variable_name*

Note: *well_number* is the well number used in GEM.

Example:

```
DoGet("GEM.Well[{OIL1}].OilRatSC")
DoGet("GEM.Well[{#1}].OilRatRC")
```

Well Variable List

Variable name	Descriptions	Units/Remarks
WelTyp	Well type; Indicates for producer or injector.	value = 1 : producer -1 : injector
WelLnk	Well link state. Indicates the well is linked to the GAP surface network or not. Please note at the moment of initial, all GEM wells are assumed to be linked.	value = 1 : linked 0 : unlinked
TmpBHF	Well bottom-hole temperature	deg.F
PrsBHF	Well bottom-hole pressure	psig
PrsBlkRef	Well reference grid block pressure	psig
PrsBlkMow	Well grid block mobility weighted pressure	psig
OilRatSC	Oil rate at stock-tank condition	STB/day
WatRatSC	Water rate at stock-tank condition	STB/day

GasRatSC	Gas rate at stock-tank condition	MMscf/day
OilRatRC	Oil rate at bottom-hole condition	RB/day
WatRatRC	Water rate at bottom-hole condition	RB/day
GasRatRC	Gas rate at bottom-hole condition	MMrcf/day
BHFRatRC	Bottom-hole fluid rate	RB/day
BHFCumRC	Cumulative bottom-hole fluid	RB
OnTimeFrc	Well on-time fraction	Fraction (0 ~ 1)

Note: Well rates above represent average rates.

Group Variables

Format:

instance_name.Grup[{*group_name*}].*group_variable_name*

Note: If there is no user group defined in GEM data set, group name FIELD is used to represent all the wells defined in GEM data set. Otherwise, a user given top node name can be used.

Example:

```
DoGet("GEM.Grup[{Gather-1}].OilRatSCPrd")
DoGet("GEM.Grup[{FIELD}].OilRatRCPrd")
```

Group Variable List

Variable name	Descriptions	Units/Remarks
OilRatSCPrd	Oil production rate at stock-tank condition	STB/day
WatRatSCPrd	Water production rate at stock-tank condition	STB/day
GasRatSCPrd	Gas production rate at stock-tank condition	MMscf/day
OilRatRCPrd	Oil production rate at bottom-hole condition	RB/day
WatRatRCPrd	Water production rate at bottom-hole condition	RB/day
GasRatRCPrd	Gas production rate at bottom-hole condition	MMrcf/day
BHFRatRCPrd	Bottom-hole fluid production rate	RB/day
BHFCumRCPrd	Cumulative bottom-hole fluid production	RB
WatRatSCInj	Water injection rate at stock-tank condition	STB/day
GasRatSCInj	Gas injection rate at stock-tank condition	MMscf/day
WatRatRCInj	Water injection rate at bottom-hole condition	RB/day
GasRatRCInj	Gas injection rate at bottom-hole condition	MMrcf/day
BHFRatRCInj	Bottom-hole fluid injection rate	RB/day
BHFCumRCInj	Cumulative bottom-hole fluid injection	RB

Sector Variables

Format:

instance_name.Sect[{*sector_name*}].*sector_variable_name*
or
instance_name.Sect[{#*sector_number*}].*sector_variable_name*

Note: Sector name with 'Entire Field' or sector number with 0 is used to represent the whole active blocks of the reservoir model.

Example:

```
DoGet("GEM.Sect[{#0}].OilInPSC")
DoGet("GEM.Sect[{Entire Field}].OilInPSC")
```

Sector Variable List

Variable name	Descriptions	Units/Remarks
OilInPSC	Stock-tank oil in-place	STB
WatInPSC	Stock-tank water in-place	STB
GasInPSC	Stock-tank gas in-place	MMscf
FrGInPSC	Stock-tank free gas in-place	MMscf
OilInPRC	Oil in-place at reservoir condition	RB
WatInPRC	Water in-place at reservoir condition	RB
GasInPRC	Gas in-place at reservoir condition	MMrcf
VolTPV	Total pore volume	RB
VolHPV	Hydrocarbon pore volume	RB
PrsTPV	Average pressure over the total pore volume	psig
PrsHPV	Average pressure over the hydrocarbon pore volume	psig

Example:

In the script for voidage replacement, replace the summation of well rates by using group rates:

```
If (InStr(ModuleList, "GAP") > 0) Then
  ' fallback values for FVFProd and FVFInj
  if (Resolve.Timestep = 0) then
    FVFProd0 = 1.3
    FVFInj0 = 1
  end if

  ' Estimate FVF values for producers and injectors
  Qres = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].BHFRatRCPrd")))
  QWsurf = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].WatRatSCPrd")))
  QOsurf = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].OilRatSCPrd")))

  if (Qres > 0 And QWsurf + QOsurf > 0) then
    FVFProd0 = Qres / (QOsurf + QWsurf)
  End If
```

```
  Qres = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].BHFRatRCInj")))
  QWsurf = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].WatRatSCInj")))

  if (Qres > 0 And QWsurf > 0) then
    FVFInj0 = Qres / QWsurf
  End If

  FVFRatio0 = FVFProd0 / FVFInj0

  ' Get the cumulative voidages so far
  VoidProd0 = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].BHFCumRCPrd")))
  VoidInj0 = Abs(CDbl(DoGet("GEM.Grup[{FIELD}].BHFCumRCInj")))

End If
```

Appendix E

Compositional Simulation for Sequestration of CO_2 and Other Greenhouse Gases in Saline Aquifers

The modeling of CO_2 storage in saline aquifers involves the solution of the component transport equations, the equations for thermodynamic equilibrium between the gas and aqueous phase, and the equations for geochemistry. The latter involve reactions between the aqueous species and mineral precipitation and dissolution.

The modeling of transport of reactive components in the aqueous phase has been a subject of intensive research in the hydro-geochemical field (Lichtner, 1988, 1991, 1992; Liu and Ortoleva, 1996a, 1996b; Sevougian et al., 1992, 1993; Steefel et al., 1992, 1994, 1996; Yeh and Tripathi, 1989, 1991; Pruess et al., 2003). There are essentially two approaches for solving the coupled system of equations: the sequential solution method and the simultaneous solution method. In the sequential solution approach, the flow equations and chemical-equilibrium equations are solved separately and sequentially. Iterations are applied between the two systems until convergence is achieved. The simultaneous solution approach solves all equations simultaneously with Newton's method. The simultaneous solution approach is also referred to as the fully-coupled approach. Although the fully-coupled method is recognized as the most stable approach, it has not been fully exploited as the required numerical solution method is very involved.

This Appendix describes the methods implemented in GEM for modeling of CO_2 storage in saline aquifers: adaptive-implicit multiphase multicomponent flow simulation with phase and chemical equilibrium and rate-dependent mineral dissolution/ precipitation using the fully-coupled approach. The success of this approach depends strongly on efficient sparse solution techniques and robust solution methods for multicomponent multiphase flow systems that were developed previously for an EOS compositional simulator (Nghiem and Rozon, 1988; Collins et al., 1992).

Thermodynamic Equilibrium

Henry's Law

It is assumed that the gaseous phase and the aqueous phase are in thermodynamic equilibrium. The equation for phase equilibrium is the equality of fugacities of the component in the gaseous and aqueous phase. i.e.

$$g_i = f_{ig} - f_{iw} = 0, \; i = 1, \ldots, n_c \qquad (1)$$

where

f_{ig}	=	fugacity of component i in the gas phase
f_{iw}	=	fugacity of component i in the aqueous phase
n_c	=	number of gaseous components

To be general, the gas is considered to be a multicomponent mixture. This is required if the injected CO_2 is impure or if flue gas (a mixture of CO_2 and N_2) is injected. The fugacity f_{ig} is calculated from an equation of state (EOS) (e.g. Peng and Robinson, 1976; Soave 1972). The fugacity f_{iw} of gaseous components soluble in the aqueous phase is calculated using Henry's law (Li and Nghiem, 1986), i.e.:

$$f_{iw} = y_{iw} \cdot H_i \qquad (2)$$

where

H_i	=	Henry's constant of component i
y_{iw}	=	mole fraction of component i in the aqueous phase

The Henry's constants H_i are functions of pressure, temperature and salinity. In versions of GEM prior to 2008.10, Henry's constants at a given pressure, temperature and salinity are calculated by WinProp. The molar volume of the gaseous component in the aqueous solution is also estimated by WinProp at a given temperature. The following equation is then used to calculate Henry's constants at other pressures:

$$\ln H_i = \ln H_i^* + \frac{\bar{v}_i \left(p - p^*\right)}{R\,T} \qquad (3)$$

where

H_i	=	Henry's constant for component i at pressure p and temperature T
H_i^*	=	Henry's constant for component i at pressure p^* and temperature T
P	=	Pressure
p^*	=	reference pressure
R	=	universal gas constant
T	=	temperature (K)
$\bar{v}_i$	=	partial molar volume of component i

The above approach assumes that the temperature and salinity do not change significantly throughout the aquifer and that a representative value for temperature and salinity could be used to calculate Henry's constants. This may not be applicable for a thick aquifer where the temperature difference between the top and bottom of the aquifer is substantial or for the case where CO_2 is injected at a temperature significantly different from that of the aquifer. In addition, the salinity of the aquifer may change due to water injection.

Thus an implementation of Henry's constant correlations will provide more flexibility to handle the above situations. Harvey (1996) published correlations for Henry's constants of many gaseous components including CO_2, N_2, H_2S, and CH4. The latter are particularly important in the sequestration of Greenhouse Gases (GHG). These correlations have been implemented in GEM and are described in the following.

Gas solubility depends on the salinity of the aqueous phase. The solubility of light gases normally decreases with increasing salinity, this phenomenon is referred to as the salting-out process. Models for salting-out coefficients have also been implemented in GEM.

Henry's Law Constant Correlations

The following correlation is due to Harvey (1996). Henry's constants *at the H_2O saturation pressure and T* are estimated from:

$$\ln H_i^s = \ln p_{H_2O}^s + A\left(T_{r,H_2O}\right)^{-1} + B\left(1 - T_{r,H_2O}\right)^{0.355}\left(T_{r,H_2O}\right)^{-1} + C\left[\exp\left(1 - T_{r,H_2O}\right)\right]\left(T_{r,H_2O}\right)^{-0.41} \quad (4)$$

where

H_i^s = Henry's constant for component i at the saturation pressure of H_2O (MPa)

$p_{H_2O}^s$ = saturation pressure of H_2O in MPa at T(K)

T_{c,H_2O} = critical temperature of H_2O (K)

T_{r,H_2O} = $T/T_{c,H_2O}$, reduced temperature of H_2O

Table 1 give values of the parameters A, B and C for the gases of interest.

Table 1 – Parameters for correlation of aqueous Henry's constants

Gaseous Solute	A	B	C
CO_2	–9.4234	4.0087	10.3199
N_2	–11.6184	4.9266	13.3445
H_2S	–5.7131	5.3727	5.4227
CH_4	–11.0094	4.8362	12.5220

The saturation pressure $p_{H_2O}^s$ of H_2O at T is calculated from the Saul and Wagner (1987) correlation (Equation 2.6). The Henry's law constant at p and T is then given by:

$$\ln H_i = \ln H_i^s + \frac{1}{R\,T}\int_{p^s_{H_2O}}^{p} \bar{v}_i \, dP \qquad (5)$$

where $\bar{v}_i$ is the partial molar volume of component i in the aqueous phase evaluated the T. The following correlations are used to estimate $\bar{v}_i$.

For CO_2, the correlation due to Garcia (2001) is used:

$$\bar{v}_{CO_2}\ (cm^3/mol) = 37.51 - 9.585\times10^{-2}\,\hat{T} + 8.740\times10^{-4}\,\hat{T}^2 - 5.044\times10^{-7}\,\hat{T}^3 \qquad (6)$$

where $\hat{T}$ is the temperature in °C.

For CH_4, the correlation due to Rettich et. al. (1981) is used:

$$\bar{v}_{CH_4}\ (cm^3/mol) = \exp\left(3.541 + 1.23\times10^{-3}\,\hat{T}\right) \qquad (7)$$

For N_2, the correlation due to Perez and Heidemann (2006) is used:

$$\bar{v}_{N_2}\ (cm^3/mol) = 15.732 + 6.60\times10^{-2}\,T \qquad (8)$$

For H_2S, the general approach described in Li and Nghiem (1986) is used. The molar volume $\bar{v}_i$ is calculated from the correlation of Lyckman et al. (1965), which was reported by Heidemann and Prausnitz (1977) in the form:

$$\left(\frac{p_{ci}\bar{v}_i}{RT_{ci}}\right) = 0.095 + 2.35\left(\frac{Tp_{ci}}{CT_{ci}}\right) \qquad (9)$$

p_{ci} and T_{ci} are respectively the critical pressure and temperature of the component. C is the cohesive energy density of water given by

$$C = (H^0_{H_2O} - H^s_{H_2O} + p^s_{H_2O}v^s_{H_2O} - RT)/v^s_{H_2O} \qquad (10)$$

where

$p^s_{H_2O}$	=	saturation pressure of H_2O at temperature T
$v^s_{H_2O}$	=	saturated liquid molar volume of H_2O at temperature T
$H^0_{H_2O} - H^s_{H_2O}$	=	enthalpy departure of liquid H_2O at $p^s_{H_2O}$ and T

The saturation pressure and molar volume at saturation for water can be calculated using the correlations given previously. The enthalpy departure of liquid water is determined using the Yen-Alexander correlation as reported in Reid et al. (1977):

$$\frac{H^0_{H_2O} - H^s_{H_2O}}{T_{c,H_2O}} = \frac{7.0 + 4.5688\,[-\ln(p^s_{H_2O}/p_{c,H_2O})]^{0.333}}{1.0 + 0.004\,[\ln(p^s_{H_2O}/p_{c,H_2O})]} \qquad (11)$$

where

T_{c,H_2O}	=	critical temperature of water
p_{c,H_2O}	=	critical pressure of water

Effect of Salinity

The salting out coefficient implemented in GEM is defined by the following relation between the Henry's constant in pure water and the Henry's constant in brine:

$$\ln\left(\frac{H_{salt,i}}{H_i}\right) = k_{salt,i}\ m_{salt} \qquad (12)$$

where

$H_{salt,i}$	=	Henry's constant of component i in brine (salt solution)
H_i	=	Henry's constant of component i at zero salinity
$k_{salt,i}$	=	salting-out coefficient for component i
m_{salt}	=	molality of the dissolved salt (mol/kg H_2O)

Bakker (2003) gives the following correlations for the salting-out coefficients for CO_2 and CH_4:

$$k_{salt,CO_2} = 0.11572 - 6.0293\times10^{-4}\ \hat{T} + 3.5817\times10^{-6}\ \hat{T}^2 - 3.7772\times10^{-9}\ \hat{T}^3 \qquad (13)$$

where $\hat{T}$ is the temperature in °C.

$$k_{salt,CH_4} = 3.38828 - 0.0318765\,T - 2.31891\times10^{-7}\ T^3 + 2.22938\times10^{-10}\ T^4 - 8.83764\times10^{-14}\ T^5 \qquad (14)$$

where T is the temperature in K.

Perez and Heidemann (2006) give the following correlation for the salting-out coefficient for N_2 and CH_4:

$$k_{salt,N_2} = 3.1150 - 0.01690\,T + 2.4950\times10^{-5}\ T^2 \qquad (15)$$

Finally, Suleimenov and Krupp (1994) give the following correlation for the salting-out coefficient for H_2S:

$$k_{salt,H_2S} = 8.37106265\times10^{-2} - 5.135608863\times10^{-4}\ T + 6.387039005\times10^{-6}\ T^2 - 2.217360319\times10^{-8}\ T^3 - 5.069412169\times10^{-11}\ T^4 + 2.827486651\times10^{-13}\ T^5 \qquad (16)$$

H_2O Vaporization

Phase Equilibrium Equation

Another important aspect is the vaporization of H_2O. It has been reported that complete vaporization of H_2O could occur during gas injection, which would in turn affect the gas injectivity.

To model H_2O vaporization, the equation for thermodynamic equilibrium (1) is applied to the H_2O component, i.e.

$$g_{n_c} = f_{H_2O,g} - f_{H_2O,w} = 0 \qquad (17)$$

The fugacity $f_{H_2O,g}$ is calculated from a cubic EOS as in the case of other gaseous components. The fugacity $f_{H_2O,w}$ is calculated from the procedure described below.

Calculation of H_2O Fugacity in Aqueous Phase

1. Calculation of fugacity of saturated H_2O at reservoir temperature T

The equation due to Canjar and Manning (1967) is used:

$$\frac{f^s_{H_2O}}{p^s_{H_2O}} = \begin{cases} 0.9958 + 9.68330\times10^{-5}\,T' - 6.1750\times10^{-7}\,T'^2 - 3.08333\times10^{-10}\,T'^3, T' > 90°F \\ 1, \text{otherwise} \end{cases} \qquad (18)$$

where

$f^s_{H_2O}$ = fugacity of H_2O at saturation pressure and temperature T' (°F)

$p^s_{H_2O}$ = saturation pressure of H_2O at temperature T' (°F)

T' = temperature in °F corresponding to T

2. Calculation of saturation pressure of H_2O at reservoir temperature T

The equation due to Saul and Wagner (1987) is used:

$$\ln\left(\frac{p^s_{H_2O}}{p_{c,H_2O}}\right) = \frac{T_{c,H_2O}}{T}\left(a_1\tau + a_2\tau^{1.5} + a_3\tau^{3.5} + a_4\tau^{3.5} + a_5\tau^4 + a_6\tau^{7.5}\right) \qquad (19)$$

with

$a_1 = -7.85823$, $a_4 = +22.6705$

$a_2 = +1.83991$, $a_5 = -15.9393$

$a_3 = -11.7811$, $a_6 = +1.77516$

p_{c,H_2O} = critical pressure of H_2O (22.064×10^6 kPa)

T_{c,H_2O} = critical temperature of H_2O (647.14 K)

and where

$$\tau = 1 - \frac{T}{T_{c,H_2O}} \qquad (20)$$

3. Calculation of molar volume of H_2O at temperature T

The Rowe and Chou (1970) equation is used to calculate the specific volume of H_2O:

$$\hat{v}_{H_2O} = A(T) - \hat{p}\,B(T) - \hat{p}^2\,C(T) \qquad (21)$$

where

$\hat{v}_{H_2O}$ = specific volume of H_2O (cm^3/g)

$\hat{p}$ = absolute pressure (kg/cm^2)

The other parameters in Equation (21) are given by:

$$A(T) = 5.916365 - 0.010357694\,T + 0.9270048 \times 10^{-5}\,T^2 - 1127.522/T + 100674.1/T^2 \qquad (22a)$$

$$B(T) = 0.5204914 \times 10^{-2} - 0.10482101 \times 10^{-4}\,T + 0.8328532 \times 10^{-8}\,T^2 - 1.1702939/T + 102.2783/T^2 \qquad (22b)$$

$$C(T) = 0.118547 \times 10^{-7} - 0.6599143 \times 10^{-10}\,T \qquad (22c)$$

The molar volume of H_2O (m^3/mol) is then given by

$$v_{H_2O} = 10^{-6}\,\hat{v}_{H_2O} \times M_{H_2O} \qquad (23)$$

where M_{H_2O} = 18.015 is the molecular weight of H_2O.

4. Calculation of fugacity of pure H_2O at pressure p and temperature T

The fugacity of pure H_2O at p and T is calculated from:

$$f_{H_2O} = f^s_{H_2O} \exp\left(\int_{p^s_{H_2O}}^{p} \frac{v_{H_2O}}{R\,T}\,dp\right) \qquad (24)$$

where $f^s_{H_2O}$ and v_{H_2O} are estimated from Equations (18) and (23) respectively.

5. Calculation of fugacity of H_2O at p and T in aqueous solution

Finally the fugacity of H_2O is the aqueous solution is calculated from:

$$f_{H_2O,w} = y_{H_2O,w}\,f_{H_2O} \qquad (25)$$

where $y_{H_2O,w}$ is the mole fraction of H_2O in the aqueous phase.

Gas and Aqueous Phase Properties

The gas density is calculated with the Peng-Robinson EOS. The gas viscosity is estimated from the Jossi, Stiel and Thodos correlation (Reid et al., 1977). The aqueous phase density and viscosity are calculated respectively from the Rowe and Chou (1970) correlation and the Kestin et al. (1981) correlation.

Reaction Stoichiometry

Chemical reactions occur between components in the aqueous phase and between minerals and aqueous components. Let R_{aq} be the number of reactions between aqueous components and R_{mn} be the number of reactions between minerals and aqueous components. The components in the aqueous phase comprise gaseous components that are soluble in the aqueous phase (n_c components) as well as components that exist only in the aqueous phase (n_a components). Let n_m be the number of mineral components, n_{aq} be the total number of components in the aqueous phase ($n_{aq} = n_c + n_a$) and n_{ct} be the total number of components ($n_{ct} = n_c + n_a + n_m$).

A reaction between species in the aqueous phase has the following stoichiometry:

$$\sum_{k=1}^{n_{aq}} \nu_{k\alpha}^{a} A_k = 0, \quad \alpha = 1,\ldots,R_{aq} \qquad (26)$$

where A_k is the chemical symbol for the k-th aqueous species. The dissolution or precipitation chemical reaction for a mineral has the following stoichiometry:

$$\sum_{k=1}^{n_{ct}} \nu_{k\beta}^{m} A_k = 0, \quad \beta = 1,\ldots,R_{mn} \qquad (27)$$

An example of the reaction (26) is the dissociation of CO_2(aq), i.e.

$$H^+ + HCO_3^- - CO_2(aq) - H_2O = 0 \qquad (28)$$

An example of the reaction (27) is dissolution of Calcite ($CaCO_3$):

$$Ca^{++} + CO_3^{--} - \text{Calcite} = 0 \qquad (29)$$

The chemical reactions (26) are homogeneous reactions that involve only components in the aqueous phase. They do not involve any minerals and therefore the summation involves only n_{aq} components. The chemical reactions in (27) are heterogeneous reactions that involve mineral species and aqueous species. Normally, a mineral reacts only with aqueous species and not with other minerals. Therefore the stoichiometric coefficients associated with mineral species in Equation (27) are all zero except for the one corresponding to the reacting mineral. Also, in this paper, all mineral reactions are written as dissolution reactions with the stoichiometric coefficient for the reacting mineral equal to -1. There is no loss of generality by adopting this convention, since precipitation can also be modeled as discussed later.

Reactions between components in the aqueous phase are fast relative to mineral dissolution/precipitation reactions. Therefore intra-aqueous reactions are represented as chemical-equilibrium reactions whereas mineral dissolutions/precipitations are represented as rate-dependent reactions.

Chemical Equilibrium

The chemical equilibrium reactions (26) are modeled with chemical equilibrium constants (Bethke, 1996). The governing equations for the chemical equilibrium reactions are:

$$Q_\alpha - K_{eq,\alpha} = 0, \ \alpha = 1,...,R_{aq} \qquad (30)$$

with

$$Q_\alpha = \prod_{k=1}^{n_{aq}} a_k^{\nu_{k\alpha}} \qquad (31)$$

where $K_{eq,\alpha}$ is the chemical equilibrium constant for the aqueous reaction α, a_k is the activity of component k, $\nu_{k,\alpha}$ are the stoichiometry coefficients and Q_α is the activity product. Kharaka et al. (1989) and Delany and Lundeen (1990) provide tables of values of $K_{eq,\alpha}$ as a function of temperature for many aqueous reactions.

The activities a_k are related to the molality m_k (moles per kg of H_2O) as follows:

$$a_i = \gamma_i \, m_i, \ i = 1,..,n_{aq} \qquad (32)$$

where γ_i is the activity coefficient. For ideal solution, $\gamma_i = 1$, and the activity is equal to the molality. However, for most cases, the solution is non-ideal and the preferred model for ionic activity coefficients is the B-dot model (Bethke, 1996):

$$\log \gamma_i = -\frac{A_\gamma \, z_i^2 \sqrt{I}}{1 + \mathring{a}_i \, B_\gamma \sqrt{I}} + \dot{B} \, I \qquad (33)$$

where A_γ, B_γ and $\dot{B}$ are temperature dependent parameters, $\mathring{a}_i$ is the ion size parameter. I is the ionic strength given by:

$$I = \frac{1}{2} \sum_{k=1}^{n_{aq}} m_k \, z_k^2 \qquad (34)$$

with z_k being the charge of the k-th ion.

The activities of CO_2(aq) and of H_2O can be assumed to be equal to 1 without introducing significant errors.

Mineral Dissolution and Precipitation Reactions

The rate law for the mineral dissolution and precipitation reaction (27) is (Bethke, 1996):

$$r_\beta = \hat{A}_\beta \, k_\beta \left(1 - \frac{Q_\beta}{K_{eq,\beta}}\right), \quad \beta = 1,\ldots,R_{mn} \qquad (35)$$

where r_β is the rate, $\hat{A}_\beta$ is the reactive surface area for mineral β, k_β is the rate constant of mineral reaction β, $K_{eq,\beta}$ is the chemical equilibrium constant for mineral reaction β and Q_β is the activity product of mineral reaction β. The activity product Q_β is analogous to the activity product for aqueous chemical equilibrium reactions (Equation 31), i.e.

$$Q_\beta = \prod_{k=1}^{n_{aq}} a_k^{\nu_{k\beta}} \qquad (36)$$

The product does not involve the activities of the minerals as they are equal to unity. The chemical equilibrium constants $K_{eq,\beta}$ for many minerals are also available in the literature (Kharaka et al., 1989) and Delany and Lundeen, 1990). The ratio ($Q_\beta/K_{eq,\beta}$) is called the *saturation index* of the reaction. If ($Q_\beta/K_{eq,\beta}$) > 1, mineral dissolution occurs and if ($Q_\beta/K_{eq,\beta}$) < 1, mineral precipitation occurs. In this document, the convention adopted is that the rate r_β is negative for dissolution and positive for precipitation. The rate is zero when ($Q_\beta/K_{eq,\beta}$) = 1.

The rate of dissolution/precipitation in Equation (35) applies to the mineral. The rate of formation/consumption of the different aqueous species is obtained by multiplying r_β by the respective stoichiometry coefficient:

$$r_{k\beta} = \nu_{k\beta} \cdot r_\beta \qquad (37)$$

The reaction rate constants, k_β, are normally reported in the literature at a reference temperature T_0 (usually 298.15 K or 25°C). The following equation is used to calculate the rate constant at a different temperature T:

$$k_\beta = k_{0\beta} \exp\left[-\frac{E_{a\beta}}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right)\right] \qquad (38)$$

where $E_{a\beta}$ is the activation energy for reaction β [J/mol] and $k_{0\beta}$ is the reaction rate constant for reaction β at T_0 [mol/(m^2 s)].

The reactive surface area $\hat{A}_\beta$ is another important parameter in the calculation of the rate. The following equation is used to calculate the reactive surface area with change in the moles of minerals through dissolution or precipitation:

$$\hat{A}_\beta = \hat{A}_\beta^0 \cdot \frac{N_\beta}{N_\beta^0} \qquad (39)$$

where $\hat{A}_\beta^0$ is the reactive surface area at time 0, N_β is the mole number of mineral β per unit grid block volume at current time and N_β^0 is the mole number of mineral β per unit grid block bulk volume at time 0.

Mineral dissolution and precipitation change the void volume of the porous medium. The porosity is calculated as follows:

$$\hat{\phi}^* \equiv \phi^* - \sum_{\beta=1}^{n_m} \left(\frac{N_\beta}{\rho_\beta} - \frac{N_\beta^0}{\rho_\beta} \right) \qquad (40)$$

$$\phi = \hat{\phi}^* [1 + c_\phi (p - p^*)] \qquad (41)$$

where ϕ is the porosity, ϕ^* is the reference porosity without mineral precipitation/dissolution, $\hat{\phi}^*$ is the reference porosity including mineral precipitation/dissolution, N_β is the total moles of mineral β per bulk volume at the current time, N_β^0 is the total moles of mineral β per bulk volume at time 0, ρ_β is the mineral molar density and c_ϕ is the rock compressibility, p^* is the reference pressure.

The absolute permeability also changes with mineral dissolution and precipitation. The Kozeny-Carman equation is used:

$$\frac{k}{k^0} = \left(\frac{\phi}{\phi^0} \right)^3 \cdot \left(\frac{1-\phi^0}{1-\phi} \right)^2 \qquad (42)$$

where k^0 and ϕ^0 are the initial permeability and porosity respectively.

Component Material Balance Equation

Darcy's law governs the flow in the reservoir. The dispersion and diffusion of components in the aqueous phase also contribute to the movement of components in the aqueous phase. Let n_{ct} be the total number of species. The components are ordered as follows: n_c gaseous components, n_a aqueous components and n_m mineral components The conservation equations for the different species are:

$$\begin{aligned}\psi_i \equiv\ & \Delta T_g^u y_{ig}^u \left(\Delta p^{n+1} + \Delta P_{cwg}^u - \tilde{\rho}_g^u g \Delta d\right) + \\ & \Delta T_w^u y_{iw}^u \left(\Delta p^{n+1} - \tilde{\rho}_w^u g \Delta d\right) + \sum_{q=g,w} \Delta D_{iq}^u \Delta y_{iq}^u + \\ & V\sigma_{i,aq}^{n+1} + q_i^{n+1} - \frac{V}{\Delta t}\left(N_i^{n+1} - N_i^n\right) = 0, \quad i = 1,\ldots,n_c\end{aligned} \tag{43}$$

$$\begin{aligned}\psi_j \equiv\ & \Delta T_w^u y_{jw}^u \left(\Delta p^{n+1} - \tilde{\rho}_w^u g \Delta d\right) + \Delta D_{iw}^u \Delta y_{iw}^u + \\ & V\sigma_{j,aq}^{n+1} + V\sigma_{j,mn}^{n+1} + \\ & q_j^{n+1} - \frac{V}{\Delta t}\left(N_{ja}^{n+1} - N_{ja}^n\right) = 0, \quad j = n_c + 1,\ldots,n_c + n_a\end{aligned} \tag{44}$$

$$\psi_k \equiv V\sigma_{k,mn}^{n+1} - \frac{V}{\Delta t}\left(N_k^{n+1} - N_k^n\right) = 0, \quad k = n_c + n_a + 1,\ldots,n_{ct} \tag{45}$$

In the above equation, the superscript n denotes the old time level and n+1 denotes the new time level. Equations (43) and (44) are discretized in an adaptive-implicit manner (Nghiem and Rozon, 1988 & 1989, Collins et al., 1992). The superscript u = n for explicit grid blocks and n+1 for implicit grid blocks. The terms ($V\sigma_{i,aq}$) and ($V\sigma_{i,mn}$) in Equations (43) to (44) correspond respectively to the intra-aqueous reaction rates and mineral dissolution/precipitation rates. $\sigma_{i,mn}$ is related to the rates in Equation (35) as follows:

$$V\sigma_{i,mn} = VS_w \sum_{\beta=1}^{R_{mn}} r_{i\beta} = VS_w \sum_{\beta=1}^{R_{mn}} \nu_{i\beta} r_\beta \tag{46}$$

As r_β [mol/(m^3 s)] is the reaction rate per unit bulk volume of rock when the pore space is filled with water, multiplication by the factor VS_w is required in the above equation to obtain the proper rate when part of the pore space is occupied by gas. The ($V\sigma_{i,aq}$) correspond to rates of the chemical equilibrium reactions. There is no accurate means available to compute these rates, but they can be eliminated as described in the following section.

Elimination of the Chemical Equilibrium Reaction Rates

The approach used is similar to those described in Lichtner (1985, 1988) and Sevougian et al. (1992). Consider the chemical-equilibrium reactions in the aqueous phase (Equation 30). Here $n_{aq} = n_c + n_a$ is the total number of components in the aqueous phase (gaseous solutes and aqueous components). It can be shown that for R_{aq} independent chemical reactions involving n_{aq} components ($n_{aq} > R_{aq}$), a canonical stoichiometric matrix $\mathbf{V}_c$ of dimension $n_{aq} \times R_{aq}$ that has the following form can be defined:

$$\mathbf{V}_c = \begin{bmatrix} \nu_{11} & \nu_{12} & \cdots & \nu_{1R_{aq}} \\ \vdots & \vdots & & \vdots \\ \nu_{n_{cp}1} & \nu_{n_{cp}2} & \cdots & \nu_{n_{cp}R_{aq}} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (47)$$

Column α ($\alpha = 1,\ldots,R_{aq}$) of $\mathbf{V}_c$ corresponds to the stoichiometry of reaction α, i.e.

$$\sum_{j=1}^{n_c} \nu_{j\alpha} A_j + A_{n_c+\alpha} = 0 \qquad (48)$$

Equations (47) and (48) show that each of the n_{cp} components corresponding the first n_{cp} rows of $\mathbf{V}_c$ can participate in more than one of the independent chemical reactions (30). Each of the components corresponding to the last $R_{aq} = n_{aq} - n_{cp}$ rows of $\mathbf{V}_c$ participates only in one independent chemical reaction. The first n_{cp} components are called primary components and the last R_{aq} components are called secondary components. The gaseous components are always primary components.

The canonical stoichiometric matrix in Equation (47) can be partitioned as follows:

$$\mathbf{V}_c = \begin{bmatrix} \left(\tilde{\mathbf{V}}\right)_{n_{cp} \times R_{aq}} \\ \mathbf{I}_{R_{aq}} \end{bmatrix} \qquad (49)$$

where $\mathbf{I}_{R_{aq}}$ is the identity matrix of dimensions $R_{aq} \times R_{aq}$. For the R_{aq} chemical-equilibrium reactions, the forward rates are equal to the backward rates. There exists an *Equilibrium-Rate-Annihilation* (ERA) matrix (Sevougian et al., 1992) **E** of dimension $n_c \times n_{aq}$ such that the left n_c x n_c submatrix of **E** is non-singular and:

$$\mathbf{E}\,\mathbf{V}_c = \mathbf{0}_{n_{cp} \times R_{aq}} \qquad (50)$$

It can be seen that

$$\mathbf{E} = \left[\mathbf{I}_{n_{cp}} \quad \left(-\tilde{\mathbf{V}}\right)_{n_c \times R_{aq}}\right] \qquad (51)$$

satisfies Equation (50), and $\mathbf{I}_{n_{cp}}$ is non-singular. In simple terms, there are independent linear combinations of the component concentrations that have zero net reaction rates due to the chemical equilibrium reactions. This property can be used to reduce the number of component balance equations.

Let $\vec{\psi}$ be the vector of material balance equations for the gaseous and aqueous species (Equations 43 and 44). Using Equation (50), it can be shown that the product $\mathbf{E}\,\vec{\psi}$ gives combinations of component material balance equations that do not have terms with explicit chemical-equilibrium reactions rates $\sigma_{j,aq}$. The application of the ERA matrix reduces the number of flow equations from n_{aq} to n_{aq} -R_{aq}.

The elimination of the chemical-equilibrium reactions is important as it reduces the number of flow equations. If these chemical-equilibrium reactions are not eliminated but actually modeled with fast chemical reaction rates, the equations are more difficult to solve (Litchner, 1991, 1992, 1993). The application of the ERA matrix reduces the stiffness of the equations and thus makes the solution easier to obtain.

Solution Method

This section describes the method used to solve the equations. The method is based on the simultaneous solution of all equations using Newton's method. In compositional simulation, a volume constraint equation is normally also part of the equation set:

$$\psi_p \equiv \sum_q \left(\frac{N_q^{n+1}}{\rho_q^{n+1}} \right) - \phi^{n+1} = 0, \quad q = g, w \qquad (52)$$

Equation (52) equates the sum of phase volumes per unit reservoir volume to the pore volumes. Let $\vec{\psi}$ be the vector of the material balance equations for all gaseous and aqueous components (Equations 43 and 44). Let $\vec{\tilde{\psi}} = \mathbf{E}\,\vec{\psi}$ be the n_{cp} reduced set of "primary component" material balance equations ($n_{cp} = n_{aq} - R_{aq}$ with $n_{aq} = n_c + n_a$).

The following equations are taken as primary equations:

The volume constraint equation [Equation (52)],

The n_{cp} reduced aqueous component material equations $\vec{\tilde{\psi}}$,

The n_c phase equilibrium equations [Equation (1)],

The R_{aq} chemical equilibrium equations [Equation (30)],

The n_m mineral balance equations [Equation (45)].

In total, there are $(1 + n_{cp} + n_c + R_{aq} + n_m)$ equations. The primary unknowns are:

Pressure,

The summed mole numbers of gaseous primary components in the gas and aqueous phases, N_i, $i = 1,\ldots n_c$,

The mole numbers of aqueous components (excluding the gaseous components that are soluble in the aqueous phase), N_j, $j = n_c + 1,\ldots, n_c+n_a$,

The mole numbers of minerals, N_k, $k = n_c + n_a + 1,\ldots, n_{ct}$,

The aqueous mole numbers of gaseous components soluble in the aqueous phase, $N_{i,w}$, $i = 1,\ldots, n_c$.

Let $\vec{\Psi}$ be the vector comprising all equations for all grid blocks and $\vec{\xi}$ the vector of all primary variables. The well constraint equations and flowing bottomhole pressures are also included in $\vec{\Psi}$ and $\vec{\xi}$, after the reservoir variables. The whole system of equations is solved with Newton's method, i.e.:

$$\vec{\xi}^{(k+1)} = \vec{\xi}^{(k)} - \left[\left(\frac{\partial \vec{\Psi}}{\partial \vec{\xi}} \right)^{(k)} \right]^{-1} \vec{\Psi}^{(k)} \qquad (53)$$

where $\left(\frac{\partial \vec{\Psi}}{\partial \vec{\xi}} \right)$

is the Jacobian matrix, and the superscript (k) denotes the iteration level. The Jacobian matrix in Equation (53) is a sparse matrix that is solved by Incomplete LU (ILU) factorization followed by the GMRES iterative method. As the phase equilibrium equations (1), the chemical equilibrium equations (30) and the mineral material balance equations (45) do not involve variables in adjacent grid blocks, they are eliminated in a pre-processing step prior to applying the ILU factorization (Nghiem and Rozon, 1988 & 1989).

The above solution method allows the simultaneous convergence of the flow equations, the phase equilibrium equations, the chemical equilibrium equations and the mineral dissolution and precipitation rates. It is the most robust approach for solving these systems of strongly coupled nonlinear equations.

References

Bakker, R.J.: "Package *FLUIDS* 1. Computer Programs for Analysis of Fluid Inclusion Data and for Modelling Bulk Fluid Properties," *Chemical Geology*, Vol. 194 (2003) 3-23.

Bethke, C.M.: *Geochemical Reaction Modelling*, Oxford University Press, 1996, New York.

Canjar, L.N. and Manning, F.S.: *Thermodynamic Properties and Reduced Correlations for Gases*, Gulf Publishing Co., Houston, Texas, 1967.

Delany, J.M. and Lundeen, S.R.: "The LLNL Thermochemical Database," Lawrence Livermore National Laboratory Report UCRL-21658, 1990.

Garcia, J.E.: "Density of Aqueous Solutions of CO_2," *Lawrence Berkeley National Laboratory*, October 2001.

Harvey, A.H.: "Semiempirical Correlation for Henry's Constants over Large Temperature Ranges," *AIChE J*, Vol. 42, No. 5 (May 1996) 1491-1494.

Heidemann, R.A., and Prausnitz, J.M.: "Equilibrium Data for Wet-Air Oxidation. Water content and Thermodynamic Properties of Saturated Combustion Gases," *Ind. Eng. Process Des. Dev.*, Vol. 16 (1977) 375-381.

Kharaka, Y.K., Gunter, W.D., Aggarwal, P.K., Perkins, E. and DeBraal, J.D.: "SOLMINEQ.88: A Computer Program for Geochemical Modelling of Water-Rock Reactions," U.S. Geological Survey Water-Resources Investigations Report 88-4227, Menlo Park, California, 1989.

Kestin, J., Khalifa, H.E. and Correia, R.J.: "Tables of the Dynamic and Kinematic Viscosity of Aqueous NaCl Solutions in the Temperature Range 20-150°C and Pressure Range 0.1–35 MPa," *J. Phys. Chem. Ref. Data*, Vol. 10 (1981) 71-87.

Li, Y.-K. and Nghiem, L.X.: "Phase Equilibria of Oil, Gas and Water/Brine Mixtures from a Cubic Equation of State and Henry's Law," *Can. J. Chem. Eng.* (June 1986) 486-496.

Lichtner, P.C.: "Continuum Model for Simultaneous Chemical Reactions and Mass Transport in Hydrothermal Systems," *Geochim. Cosmochim. Acta*, Vol. 49 (1985) 779-800.

Lichtner, P.C.: "The Quasi-Stationary State Approximation to Coupled Mass Transport and Fluid-Rock Interactions in a Porous Medium, *Geochim. Cosmochim. Acta*, Vol. 52 (1988) 143-165.

Lichtner, P.C.: "The Quasi-Stationary State Approximation of Fluid/Rock Interaction: Local Equilibrium Revisited," *Diffusion, Atomic Ordering and Mass Transport*, J. Ganguly Editor; Advances in Physical Geochemistry 8; Springer-Verlag, New York (1991) 452-557.

Lichtner, P.C.: "Time-Space Continuum Description of Fluid/Rock Interaction in Permeable Media, *Water Resour. Res.*, Vol. 28 (1992) 3135-3155.

Lichtner, P.C.: "Scaling Properties of Time-Space Kinetic Mass Transport Equations and the Local Equilibrium Limit," *Ameri. J. Sci.*, Vol. 293 (1993) 257-296.

Liu, X. and Ortoleva, P.: "A General-Purpose Geochemical Reservoir Simulator," paper SPE 36640, *Proceedings* SPE Annual Technical Conference and Exhibition, Denver, Colorado, 6-9 October 1996 (a).

Liu, X. and Ortoleva, P.: “A Coupled Reaction and Transport Model for Assessing the Injection, Migration and Fate of Waste Fluids,” paper SPE 36640, *Proceedings* SPE Annual Technical

Lyckman, E.W., Eckert, C.A., and Prausnitz, J.M.: “Generalized Reference Fugacities for Phase Equilibrium Thermodynamics,” *Chem. Eng. Sci.*, Vol. 20 (1965) 685-691.

Nghiem, L.X. and Rozon, B.J.: “A Unified Approach for Handling and Solving Large Systems of Equations in Reservoir Simulation,” *Proceedings* of the First and Second International Forum on Reservoir Simulation, Alpbach, Austria, September 1988 & 1989, Paul Steiner Publisher, Leoben, Austria.

Peng, D.Y. and Robinson, D.B.: “A New Two-Constant Equation of State,” *Ind. Eng. Chem. Fundam.*, Vol. 15 (1976) 59-64.

Perez, R.J. and Heidemann, R.A.: “Coupling an equation of state and Henry's Law to model the phase equilibria of gases and brines: Examples in the N2–H2O–NaCl system,” *J. Geochemical Exploration*, Vol. 89, No. 1-3 (April-June 2006) 331-334.

Pruess, K., Xu T., Apps, J. and Garcia, J.: “Numerical Modeling of Aquifer CO_2 Disposal,” *SPEJ*, Vol. 8, No. 1 (March 2003) 48-60.

Reid, R.C., Prausnitz, J.M., and Sherwood, T.K.: *The Properties of Gases and Liquids, 3rd Edition* (1977), McGraw-Hill, New York.

Rettich, T.R., Handa, Y.P., Battino, R. and Wilhelm, E.: “Solubility of Gases and Liquids. 13. High-Precision Determination of Henry’s Constants for Methane and Ethane in Liquid Water at 275 to 328 K,” *J. Phys. Chem.*, Vol. 85 (1981) 3230-3237.

Rowe, A.M. and Chou, J.C.S.: “Pressure-Volume-Temperature-Concentration Relation of Aqueous NaCl Solutions”, *J. Chem. Eng. Data*, Vol. 15 (1970) 61-66.

Saul, A. and Wagner, W.: “International Equations for the Saturated Properties of Ordinary Water Substance,” *J. Phys. Chem. Ref. Data*, Vol. 16, No. 4 (1987) 893-901.

Sevougian, S.D., Lake, L.W. and Schechter, R.S.: “A New Geochemical Simulator to Design More Effective Sandstone Acidizing Treatments,” paper SPE 24780, *Proceedings* 67th SPE Annual Technical Conference and Exhibition, Washington, DC, October 4-7, 1992.

Sevougian S.D., Schechter, R.S. and Lake, L.W.: “Effect of Partial Local Equilibrium on the Propagation of Precipitation/Dissolution Waves,” *Ind. Eng. Chem. Res.*, Vol. 32 (1993) 2281-2304.

Soave, G.: “Equilibrium Constants from a Modified Redlich-Kwong Equation of State,” *Chem. Eng. Sci.*, Vol. 27 (1972) 1197-1203.

Steefel, C.I. and Lasaga, A.C.: “Putting Transport Into Water-Rock Interaction Models,” *Geology*, Vol. 20 (1992) 680-684.

Steefel, C.I. and Lasaga, A.C.: “A Coupled Model for Transport of Multiple Chemical Species and Kinetic Precipitation/Dissolution Reactions with Application to Reactive Flow in Single-Phase Hydrothermal Systems,” *American J. Science*, Vol. 294 (1994) 529-592.

Steefel, C.I. and MacQuarrie, K.T.: “Approaches to Modelling Reactive Transport in Porous Media,” *Reactive Transport in Porous Media, Reviews in Mineralogy 34*, Lichtner, P.C., Steefel, C.I., Oelkers, E.H. Editors, Mineral. Soc. Am. (1996) 83-129.

Suleimenov, O.M. and Krupp, R.E.: “Solubility of hydrogen sulfide in pure water and in NaCl solutions from 20 to 320 C and at saturation pressure,” *Geochimica et Cosmochimica Acta*, Vol. 58, No. 11 (1994) 2433-2444.

Yeh, G.T. and Tripathi, V.S.: “A Model for Simulating Transport of Reactive Multispecies Components: Model Development and Demonstration, *Water Resources Res.*, Vol. 27 (1991) 3075-3094.

Yeh, G.T. and Tripathi, V.S.: “A Critical Evaluation of Recent Developments in Hydrogeochemical Transport Models of Reactive Multichemical Components,” *Water Resources Res.*, Vol. 25 (1989) 93-108.

Appendix F

Thermal Option in GEM

This section describes the development of an energy equation to calculate temperature distribution in the reservoir for the compositional processes where reservoir temperature could change with time, for instance when the injected fluid is at a different temperature than the one prevailing in the reservoir. The equations to be solved are:

- Volume constraint equation: the volume of fluids must equal the pore volume.
- Component flow equations: material balance equations for oil, gas and water components.
- Energy balance equation including convection, conduction and heat losses.
- Phase equilibrium equations.

The volume constraint, component flow and phase equilibrium equations are described in Appendix-A. Here we focus on the energy balance equation and related aspects.

Energy Balance Equation

To calculate the temperature distribution for the thermal cases, the following energy balance equation is added to the equation set.

$$\begin{aligned}\psi_T \equiv\ & \Delta T_o^m\ H_o^m \left(\Delta p_o^{n+1} - \tilde{\rho}_o^m g \Delta D\right) + \Delta T_g^m H_g^m \left(\Delta p_o^{n+1} + \Delta P_{cog}^m - \tilde{\rho}_g^m g \Delta D\right) + \\ & \Delta T_w^m\ H_w^m \left(\Delta p_o^{n+1} - \Delta P_{cwo}^m - \tilde{\rho}_w^m g \Delta D\right) + \Delta \tau_c^m\ \Delta T^{n+1} + Q_{loss} + \sum_k H_k^{n+1}\ q_k^{n+1} - \\ & \frac{V}{\Delta t}\left[\phi^{n+1}\left(\sum_k \rho_k^{n+1}\ S_k^{n+1}\ U_k^{n+1}\right) - \phi^n\left(\sum_k \rho_k^n\ S_k^n\ U_k^n\right)\right] - \\ & \frac{V}{\Delta t}\left[\left(1-\phi_0\right) c_R\ \tilde{\rho}_R\ \left(T^{n+1} - T^n\right)\right] = 0, \quad k = o, g, w\end{aligned} \tag{F.1}$$

where

c_R	=	heat capacity of rock
H_k	=	molar enthalpy of Phase k (k = o, g, w)
S_k	=	saturation of Phase k (k = o, g, w)
Q_{loss}	=	heat loss rate to the (over/underburden) surroundings
T	=	temperature
ϕ	=	Porosity (0=initial; n+1=current timestep; n=previous timestep)

U_k	=	$H_k - p/\rho_k$; molar internal energy of Phase k
$\tilde{\rho}_R$	=	Rock mass density
τ_c	=	total thermal conductivity of rock and fluids

Equation (F.1) is an energy balance equation involving convection, conduction and heat losses to the surroundings.

Enthalpy Calculations

The enthalpy of the water phase is calculated from a look-up of the steam table. The oil and gas enthalpies are calculated from an EOS as follows.

The excess enthalpy for a fluid, which is the difference of the enthalpy at p and T and the ideal gas enthalpy at zero pressure and T, can be calculated from an EOS:

$$\Delta H^E = H - H^* = R\,T\left(Z-1\right) + \int_{\infty}^{v}\left[T\left(\frac{\partial p}{\partial T}\right)_v - p\right]dv \qquad \text{(F.2)}$$

where

R	=	universal gas constant
v	=	molar volume
Z	=	compressibility factor

The above quantity is also referred to as enthalpy departure. Using equation F.2, the following equation can be derived to calculate enthalpy departure for the SRK or PR EOS:

$$\Delta H^E = RT(Z-1) + \frac{T(\partial a/\partial T) - a}{b(\delta_2 - \delta_1)}\ln\left(\frac{v+\delta_2 b}{v+\delta_1 b}\right) \qquad \text{(F.3)}$$

where

$\delta_1 = 1-\sqrt{2};\ \ \delta_2 = 1+\sqrt{2}$ for the PR EOS, and

$\delta_1 = 0;\ \ \delta_2 = 1$ for the SRK EOS

and a and b are EOS parameters.

The *API Technical Data Book -- Petroleum Refining V. 2 (1983)* provides polynomials for estimating pure component enthalpies, H_i^*, in the form:

$$H_i^* = \sum_{j=0}^{5} a_{ji}\,T^j \qquad \text{(F.4)}$$

where a_{ji} are the polynomial coefficients for Component i. Knowing ΔH^E and H^*, the enthalpy H and the internal energy can be calculated from:

$$U_k = H_k - p\,v_k = H_k - p/\rho_k \qquad \text{(F.5)}$$

Calculation of Enthalpies for Pseudo Components

Kesler and Lee (1976) provide the following correlation for calculating ideal gas heat capacities for heavy-fraction pseudo components from their specific gravity and normal boiling point:

$$\begin{aligned} c_{pi}^{*} \;=\; & -0.32646 \;+ 0.02678\, K_{uop} \\ & -\left(1.3892 - 1.2122\, K_{uop} + 0.03803\, K_{uop}^{2}\right)\cdot 10^{-4}\, T - 1.5393\cdot 10^{-7}\, T^{2} \\ & - C_F \left[0.084773 - 0.080809\, \rho' - \left(2.1773 - 2.0826\, \rho'\right)\cdot 10^{-4}\, T\right] \\ & + C_F \left[\left(0.78649 - 0.70423\, \rho'\right)\cdot 10^{-7}\, T^{2}\right] \end{aligned} \qquad \text{(F.6)}$$

where ρ' is the specific gravity, T_b is the normal boiling point in °R, K_{uop} is the Watson characterization factor given by:

$$K_{uop} \;=\; \left(T_b\right)^{\frac{1}{3}} / \rho' \qquad \text{(F.7)}$$

and

$$C_F \;=\; \left[\left(\frac{12.8}{K_{uop}} - 1\right)\times\left(\frac{10}{K_{uop}} - 1\right)\times 100\right]^{2} \qquad \text{(F.8)}$$

The ideal gas enthalpy is then obtained by integrating c_p in Equation (6), i.e.

$$H_i^{*} \;=\; \int_0^{T} c_{pi}\, dT' \qquad \text{(F.9)}$$

Note that Equation (6) is of the form

$$c_{pi}^{*} \;=\; \sum_{j=0}^{2} c_{ij}\, T^{j} \qquad \text{(F.10)}$$

and therefore H_i^{*} in Equation (10) is of the form

$$H_i^{*} \;=\; \sum_{j=1}^{3} h_{ij}\, T^{j} \qquad \text{(F.11)}$$

Equation (11) is similar to Equation (4) with $a_{0i} = a_{4i} = a_{5i} = 0$.

Heat Loss Calculation

The heat loss to the overburden and underburben is calculated using the method of Vinsome and Westerveld (1980). They assumed a temperature profile in the overburden and underburden of the form:

$$T(t,z) \;=\; \left(\theta - \theta^{0} + b_1\, z + b_2\, z^{2}\right)\exp(-z/d) \;+\; \theta^{0} \qquad \text{(F.12)}$$

where

$T(t,z)$	=	over/underburden temperature at time t at a distance z from the reservoir boundary
b_1, b_2	=	time-dependent parameters
d	=	thermal diffusion length
θ	=	temperature in the boundary grid block
θ^0	=	initial temperature in boundary grid block

The diffusion length is taken as:

$$d = \frac{\sqrt{\eta t}}{2} \qquad \text{(F.13)}$$

where η is the thermal diffusivity

$$\eta = \frac{\kappa_R}{c_R \tilde{\rho}_R} \qquad \text{(F.14)}$$

and where

c_R	=	rock heat capacity
$\tilde{\rho}_R$	=	mass density of rock
κ_R	=	rock thermal conductivity

Vinsome and Westerveld (1980) derived the following expression for b_1 and b_2 and the heat loss rate:

$$b_1^{n+1} = \frac{\dfrac{\eta\,\Delta t\left(\theta^{n+1} - \theta^0\right)}{d^{n+1}} + \xi^n - \dfrac{\left(d^{n+1}\right)^3\left(\theta^{n+1} - \theta^n\right)}{\eta\,\Delta t}}{3\left(d^{n+1}\right)^2 + \eta\,\Delta t} \qquad \text{(F.15)}$$

$$b_2^{n+1} = \frac{2\,b_1^{n+1}\left(d^{n+1}\right) - \left(\theta^{n+1} - \theta^0\right) + \dfrac{\left(d^{n+1}\right)^2\left(\theta^{n+1} - \theta^n\right)}{\eta\,\Delta t}}{2\left(d^{n+1}\right)^2} \qquad \text{(F.16)}$$

$$\xi^n = \left[\left(\theta - \theta^0\right)d + b_1 d^2 + 2 b_2 d^3\right]^n \qquad (17)$$

$$Q_{loss} = \kappa_R A\left[\frac{\left(\theta^{n+1} - \theta^0\right)}{d^{n+1}} - b_1^{n+1}\right] \qquad \text{(F.18)}$$

where A is the cross-sectional area for heat loss to the overburden/underburden.

References:

Kenyon, D.E. and Behie, G.A.: "Third SPE Comparative Solution Project: Gas Cycling of Retrograde Condensate Reservoirs," *JPT* (August 1987), 981-997.

Kesler, M., and Lee, B.I.: "Improve Prediction of Enthalpy of Fractions," *Hydro. Proc.* (March 1976) 153-158.

Peaceman, D.W.: "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation," *Soc. Pet. Eng. J.* (June 1978), 183-194.

Peaceman, D.W.: "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability," *Soc. Pet. Eng. J.* (June 1983), 531-543.

Peng, D.Y., and Robinson, D.B.: "A New Two-Constant Equation of State," *Ind. Eng. Chem. Fundam.*, Vol. 15 (1976), 59-64.

Soave, G.: "Equilibrium Constants from a Modified Redlich-Kwong Equation of State," *Chem. Eng. Sci.*, Vol. 27, 1972, 1197-1203.

Vinsome, P.K.W. and Westervled, J.: "A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators," *J. Can. Petrol. Eng.* (July-September 1980), 87-90.

Appendix G

Ion Exchange in GEM

Multiple ion exchanges along with geochemical reactions are key to low-salinity waterflood (LSW process). In a reservoir environment, there is a chemical equilibrium between ions in solution and ions that sorb on the mineral (mainly clay) surface. As water with an ion concentration different from the connate water ion concentration is injected, ion exchange reactions occur. In LSW process, two typical ion exchange reactions that involve sodium, calcium and magnesium ions are:

$$Na^{+} + \frac{1}{2}\left(Ca - X_2\right) \leftrightarrow \left(Na - X\right) + \frac{1}{2}Ca^{2+} \qquad \text{(G.1)}$$

$$Na^{+} + \frac{1}{2}(Mg - X_2) \leftrightarrow (Na - X) + \frac{1}{2}Mg^{2+} \qquad \text{(G.2)}$$

where, X denotes the clay mineral in the reservoir rock

Note that the above reaction is reversible and indicates that Na^{+} is taken up by the exchanger and Ca^{2+}/Mg^{2+} are released if we inject high-salinity brine into reservoir, while the reversible is true in LSW or when there is a surplus of Ca^{2+}/Mg^{2+}. Like chemical equilibrium reactions, ion-exchange reactions are characterized by equilibrium constant defined as:

$$K_{Na\backslash Ca} = \frac{\left[\alpha\left(Ca^{2+}\right)\right]^{0.5}\alpha\left(Na - X\right)}{\alpha\left(Na^{+}\right)\left[\alpha\left(Ca - X_2\right)\right]^{0.5}} \qquad \text{(G.3)}$$

$$K_{Na\backslash Mg} = \frac{\left[\alpha\left(Mg^{2+}\right)\right]^{0.5}\alpha\left(Na - X\right)}{\alpha\left(Na^{+}\right)\left[\alpha\left(Mg - X_2\right)\right]^{0.5}} \qquad \text{(G.4)}$$

where $\alpha(.)$ denotes the activity. The activity α_i of ith component is related to the activity coefficient γ_i through the equation:

$$\alpha_i = \gamma_i m_i \qquad \text{(G.5)}$$

where m_i is the molality of the component i. Substituting Equation (G.5) into Equations (G.3) and (G.4) gives:

$$K_{Na\backslash Ca} = \frac{m(Na - X)\left[m\left(Ca^{2+}\right)\right]^{0.5}}{\left[m\left(Ca - X_2\right)\right]^{0.5} m\left(Na^{+}\right)} \times \frac{\gamma\left(Na - X\right)\left[\gamma\left(Ca^{2+}\right)\right]^{0.5}}{\left[\gamma\left(Ca - X_2\right)\right]^{0.5}\gamma\left(Na^{+}\right)} \qquad \text{(G.6)}$$

$$K_{Na\backslash Ca} = \frac{m(Na-X)\left[m\left(Mg^{2+}\right)\right]^{0.5}}{\left[m\left(Mg-X_2\right)\right]^{0.5}\, m\left(Na^{+}\right)} \times \frac{\gamma(Na-X)\left[\gamma\left(Mg^{2+}\right)\right]^{0.5}}{\left[\gamma\left(Mg-X_2\right)\right]^{0.5}\, \gamma\left(Na^{+}\right)} \qquad (G.7)$$

The activity coefficients of sodium and calcium ions in the aqueous solution can be assumed ideal (i.e. equal to molality) or calculated either by the Debye-Huckel model or by the B-dot model. The activity coefficients of Na-X, Ca-X_2, and Mg-X_2 which correspond to Na^+ and Ca^{2+}/Mg^{2+} on the exchanger surfaces are difficult to evaluate. For many ion exchange applications, the following selectivity coefficient is used instead of the equilibrium constant defined in Equations (G.6) and (G.7):

$$K'_{Na\backslash Ca} = \frac{\zeta(Na-X)\left[m\left(Ca^{2+}\right)\right]^{0.5}}{\left[\zeta\left(Ca-X_2\right)\right]^{0.5}\, m\left(Na^{+}\right)} \times \frac{\left[\gamma\left(Ca^{2+}\right)\right]^{0.5}}{\gamma\left(Na^{+}\right)} \qquad (G.8)$$

$$K'_{Na\backslash Mg} = \frac{\zeta(Na-X)\left[m\left(Mg^{2+}\right)\right]^{0.5}}{\left[\zeta\left(Mg-X_2\right)\right]^{0.5}\, m\left(Na^{+}\right)} \times \frac{\left[\gamma\left(Mg^{2+}\right)\right]^{0.5}}{\gamma\left(Na^{+}\right)} \qquad (G.9)$$

where $\zeta(Na-X)$, $\zeta\left(Ca-X_2\right)$, and $\zeta\left(Mg-X_2\right)$ are the equivalent fraction of Na+/ Ca2+ and Na+/ Mg2+ on the exchanger respectively. The use of the selectivity coefficient in Equations (G.8) and (G.9) follows the Gaines-Thomas convention (Appelo and Postma, 2005). Note that the selectivity coefficient $K'_{Na\backslash Ca}$ and $K'_{Na\backslash Mg}$ are operational variables and not thermodynamic variables like the equilibrium constant $K_{Na\backslash Ca}$ and $K_{Na\backslash Mg}$ in Equations (G.6) and (G.7). In other words, $K'_{Na\backslash Ca}$ is estimated from experimental measurements and used in the calculations. Appelo and Postma (2005) reported selectivity coefficients between Na+ and many ions. The selectivity coefficients (like chemical equilibrium constants) are functions of temperature.

Another property of the exchanger is the amount of ion that can be adsorbed on its surface. This amount is called cation exchange capacity (CEC) and is measured in eq/m^3, where eq/m^3= (mol/m^3) × (charge of ion). In GEM, all component moles are represented as moles per gridblock bulk volume. Thus the total moles of Na-X, Ca-X_2, and Mg-X_2 in a gridblock would be $VN_{Na\text{-}X}$, $VN_{Ca\text{-}X2}$, and $VN_{Mg\text{-}X2}$, where V is the gridblock bulk volume. For a given value of CEC in the gridblock, the following equation must be satisfied

$$VN_{Na-X} + 2\,VN_{Ca-X_2} + 2\,VN_{Mg-X_2} = V\,\phi(CEC) \qquad (G.10)$$

or,

$$N_{Na-X} + 2\,N_{Ca-X_2} + 2\,N_{Mg-X_2} = \phi(CEC) \qquad (G.11)$$

The equivalent fractions in Equations (G.8) and (G.9) are given by:

$$\zeta(Na-X) = \frac{N_{Na-X}}{N_{Na-X} + 2\,N_{Ca-X_2} + 2\,N_{Mg-X_2}} \qquad (G.12)$$

$$\zeta(Ca-X_2) = \frac{2\,N_{Ca-X_2}}{N_{Na-X} + 2\,N_{Ca-X_2} + 2\,N_{Mg-X_2}} \qquad \text{(G.13)}$$

$$\zeta(Mg-X_2) = \frac{2\,N_{Mg-X_2}}{N_{Na-X} + 2\,N_{Ca-X_2} + 2\,N_{Mg-X_2}} \qquad \text{(G.14)}$$

The material balance equation for the ion of charge i^+ that includes ion exchange with an exchanger X in the aqueous phase can now be written as below:

$$\Delta T^u_{aq} y^u_{i,aq}\left(\Delta p^{n+1} - \tilde{\rho}^u_{aq} g \Delta\Delta\right) + \Delta D^u_{i,aq} \Delta^u_{i,aq} + V\sigma^{n+1}_{i,aq} + V\sigma^{n+1}_{i,mn} + q^{n+1}_i$$
$$- \frac{V}{\Delta t}\left[\left(N^{n+1}_{i,aq} + N^{n+1}_{i-X}\right) - \left(N^{n}_{i,aq} + N^{n}_{i-X}\right)\right] = 0 \qquad \text{(G.15)}$$

The chemical equilibrium reaction terms $\sigma^{n+1}_{i,aq}$ are eliminated by applying the Equilibrium-Rate-Annihilation matrix (Nghiem et al., 2004, 2011). These balance equations can be solved simultaneously with the phase, chemical and ion-exchange equations for all the moles of components.

Modelling of Wettability Alteration in Low Salinity Water Flooding

The effect of wettability alteration is modeled by shifting the relative permeability curves as schematically shown in Figure G.1. Multiple relative permeability tables can be defined for a reservoir rock type, where each table corresponds to one value of a specified interpolant. Typically, two sets of relative permeability curves representing respectively low salinity and high salinity conditions are considered, an interpolation between these two curves will be carried out. The interpolant is the equivalent fraction of an ion on the rock surfaces (usually Ca++). Relative permeability curves are usually measured from laboratory experiments and served as the input data for numerical simulation.

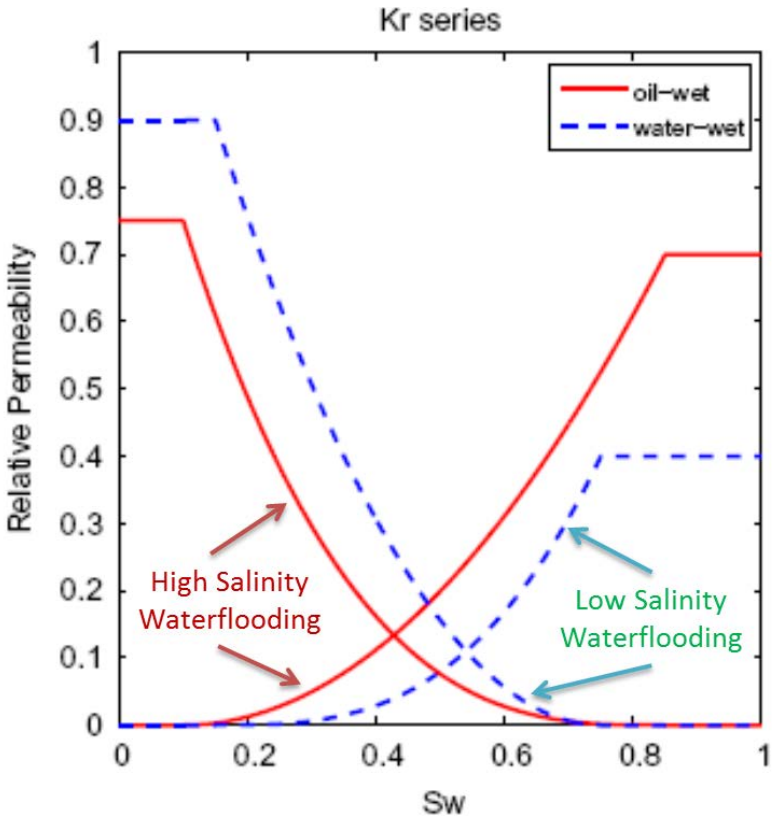


Figure G.1: Shifting of Relative Permeability Curves in LSW

References:

Appelo, C.A.J. and Postma, D.: Geochemistry, Groundwater and Pollution, 2nd Edition, CRC Press, New York, 2010.

Nghiem, L.X., Sammon, P., Grabenstetter, J., Ohkuma, H. 2004a. Modeling CO2 Storage in Aquifers with Fully-Coupled Geochemical EOS Compositional Simulator. Paper SPE 89474 presented at the SPE Fourteenth Symposium on Improved Oil Recovery, Tulsa, OK, USA, April 17-21.

Nghiem, L.X., Shrivastava, V., Kohse, B. 2011. Modeling Aqueous Phase Behavior and Chemical Reactions in Compositional Simulation. Paper SPE 141417 SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, February 21-23.

Appendix H

Polymer Option in GEM

Polymer flooding, a chemical EOR process, has been studied for several decades, primarily as a tertiary treatment after water flooding for mobility control. During water flooding in reservoirs containing heterogeneities, water channels form in high permeability zones resulting in early water breakthroughs and reduced oil recovery due to by-passed oil (incomplete sweeps). The addition of a polymer to water increases its viscosity, thus reducing the water mobility ratio which is defined as

$$M_{w-o} = \frac{\lambda_w}{\lambda_o} = \frac{k_{rw}/\mu_w}{k_{ro}/\mu_o} \qquad \text{(H.1)}$$

where λ_w and λ_o are the water and oil mobilities; k_{rw} and k_{ro} are the relative permeabilities to water and oil; and μ_w and μ_o are the viscosities of water and oil phases. Reduced mobility ratio is more favorable to oil displacement. During polymer flood, a fraction of the polymer may get adsorbed on the rock. While this results in a loss of the polymer, adsorbed polymer is believed to cause a selective reduction in the permeability to water.

Similar to most other chemical EOR processes, polymer flooding is generally employed at the tertiary recovery stage after water flood. Therefore, a measure of the performance of a polymer flood is incremental oil recovery post-water flood. Two other measures which are mentioned only infrequently in the literature are reduced water usage and delayed water break-through. Polymer flooding is employed on a commercial level in the Daqing field in northern China and Sanand field in India. Typically, polymer flood is carried out as a slug followed by fresh or saline water flood.

Two principal types of polymers are currently being used in polymer floods. The first type is the hydrolyzed polyacrylamide (HPAM) which is a synthetic polymer. Polysaccharide biopolymers are the second type of widely used polymers. Associative water-soluble polymers and salinity-tolerant polyacrylamide polymers have recently been developed.

To be effective, the polymer and its properties must remain stable throughout its passage through the reservoir. The following factors influence polymer flooding:

- Aqueous phase viscosity. The viscosity of a polymer solution, which affects the aqueous phase viscosity, depends on the polymer concentration, salinity, shear as well as temperature.
- Salinity in the formation waters. It reduces the viscosity of some polymer solutions.

- Shear. In pseudo-plastic polymer solutions, the viscosity reduces as the shear rate is increased. This may be a beneficial effect near well bores but in the rest of the reservoir, it results in reduced effectiveness. In viscoelastic polymers, the viscosity of the aqueous phase increases with an increase in the shear rate.
- Polymer retention
    - Adsorption on the rock. The adsorbed polymer is not available for modifying the viscosity of the aqueous phase. Level of adsorption depends primarily on type of polymer and rock and polymer concentration but may also depend on temperature and pressure.
    - Entrapment. Long-chain, high molecular weight polymers may not be able to enter small pores. The polymer molecules may get stuck in some of the pores that have big enough "inlet" that allow the polymer to enter but the "outlet" is smaller than required. Entrapment causes large retention at low permeabilities.
    - Retention of bio-polymers is generally less than that of polyacrylamides.
- Residual adsorption level. When polymer flood is followed by fresh or saline water flood, some of the adsorbed polymer may desorb.
- Inaccessible pore volume (IPV). The polymer molecules may not be able to pass through 100% of the pores due to size restrictions. It has been observed that when there is no adsorption taking place, the polymers travel faster than tracers. The fraction of pores not contacted by the polymer is called IPV. IPV can be experimentally determined. IPV is generally modelled as "reduced porosity" in the simulator.
- Aqueous phase effective permeability reduction. It has been observed that polymer retention leads to a reduction in the permeability of the rock to the aqueous phase. The reduction depends on the polymer type and rock characteristics. The permeability reduction usually persists even after the polymer has been displaced from the pores with fresh/saline water. Polyacrylamides usually cause larger reduction than bio-polymers. The reduction is decreases at higher salinity.
- Chemical, thermal, mechanical and biological degradation of the polymer. Chemical degradation occurs due to presence of divalent ions and oxygen. Mechanical degradation occurs due to the high shear rates near the injector or whenever the polymer is forced through at high flow rates through the porous media. Biological degradation of biopolymers occurs due to the microbial breakdown of the macromolecules of the polymer in the presence of bacteria. This phenomenon is generally modeled via reactions.
- Temperature. Polymer solution viscosity, adsorption and degradation may be affected by temperature.

Not all polymers are affected by the above factors in the same way and to the same extent. For example, HPAM polymers are more resistant to thermal degradation than biopolymers. Biopolymers, on the other hand, can withstand salinity better than the HPAM. Associative polymers have a higher tolerance for salinity and shear than HPAM.

Implementation in GEM

The factors that affect polymer floods as discussed above are implemented in GEM with the following assumptions:

- A polymer can exist only in the aqueous phase or be adsorbed on the reservoir rock. A polymer component must be defined as one of the primary aqueous component, via *COMPNAME-POLYMER.
- A polymer component can exist with other polymer components as well as other aqueous components.
- In the presence of one or more polymer components, the viscosity of the aqueous phase is computed using special keywords. These keywords are used to input data for the water component (H_2O) and polymers. Non-polymeric aqueous components and gas solutes do not contribute towards the aqueous phase viscosity. Effect of salinity is computed as described later in this section.
- Simultaneous use of gas and aqueous phase adsorption is not allowed.
- Adsorbed polymer in a grid block is at equilibrium with the aqueous phase.

Aqueous phase viscosity in presence of polymers

GEM provides the three options for computing the aqueous phase viscosity. Mixing rule and the computation of a "pure" polymer component viscosity are the primary items that differ in the three options.

Mixing rule specifies how the contributions from the aqueous components and the water component (H_2O) will be combined to arrive at the aqueous phase viscosity.

A "pure" polymer component can be viewed as the binary polymer-water solution with the maximum polymer concentration anticipated, usually the concentration of the injected solution. It is the concentration of the polymer solution for which the mixing function, $f_{i,p}$ (see below), is unity.

Model 1(*AQUEOUS-VISCOSITY *POLYMER *NONLIN1)

Two options are available for entering data for pure component viscosities.

1. In the first option, the viscosities are computed from the correlation,

   $$\ln(\mu_i) = \ln(\text{avisc}_i) + \text{bvisc}_i/T \quad \text{(H.2)}$$

   where T is the absolute temperature and avisc_i and bvisc_i are (input) correlation parameters for component i. μ_i is the viscosity of component i in the aqueous phase. avisc_i has the units of viscosity. If bvisc_i is set to zero, the component viscosity is equal to avisc_i.
2. In the second option, the viscosity is interpolated from a look-up table of viscosity against temperature values. Pressure dependence can also be specified by entering one temperature-viscosity table for each pressure value.

The above two options are available for the aqueous components as well as the water component (H_2O). Refer to the keywords *AQUEOUS-VISCOSITY, *AVISC-AQUEOUS, *BVISC-AQUEOUS, *VISCTABLE-AQUEOUS, *AVISC-H2O and *BVISC-H2O.

The following mixing rule is used to compute the aqueous phase viscosity:

$$\ln(\mu_{aq}) = f_{H_2O}.\ln(\mu_{H_2O}) + \sum_{i=1}^{nc_plm} f_{i,p}.\ln(\mu_i) + \sum_{i=nc_plm+1}^{nc_aqu} f_{i,np}.\ln(\mu_i) \qquad \text{(H.3)}$$

In the above expression,

f_{H_2O}	=	the weighting factor for water component (H2O) in the aqueous phase
μ_{H_2O}	=	the viscosity of H_2O
μ_i	=	the viscosity of pure component i
$f_{i,p}$ and $f_{i,np}$	=	the weighting factors for the component i depending on whether it is a polymer or a non-polymer. nc_plm is the number of polymers and nc_aqu is the total number of aqueous components, excluding H_2O

Current implementation ignores the third term on the right hand side of the above expression.

The weighting factors are computed as follows:

For water component (H_2O): $f_{H_2O} = N.x_{H_2O}$

For non-polymers: $f_{i,np} = N.x_{i,np}$

For polymers: $f_{i,p} = \alpha_{i,p}$

where N is a normalizing factor to ensure that the weighting factors sum to 1 and is defined as

$$N = \frac{1 - \sum_i \alpha_{i,p}}{x_{H_2O} + \sum_i x_{i,np}} \qquad \text{(H.4)}$$

with,

x_{H_2O}	=	the mole fraction of water component in aqueous phase;
$x_{i,np}$	=	the mole fraction of a non-polymeric component i; and
$\alpha_{i,p}$	=	the mixing function computed for polymer component i

The mixing function for a polymer component is interpolated from a look-up table of mixing function against polymer mole fraction. This table is generated from data entered via two keywords (*VSMIXENDP and *VSMIXFUNC).

Model 2(*AQUEOUS-VISCOSITY *POLYMER *NONLIN2)

This model is identical to Model1 described above except for the computation of the mixing function $\alpha_{i,p}$ for polymers. In Model2, the mixing function is computed as

$$\alpha_{i,p} = \frac{CP_i}{CP_{Ref,i}} \quad \text{(H.5)}$$

where,

CP_i	=	the local concentration of polymer component i and
$CP_{Ref,i}$	=	the reference concentration of polymer component i entered via PREFCONC keyword

For an aqueous phase containing only one polymer component (nc_aqu = 1, nc_plm = 1), the viscosity expression reduces to

$$\ln(\mu_{aq}) = (1 - \alpha_{i,p}) . \ln(\mu_{H_2O}) + \alpha_{i,p} . \ln(\mu_p) \quad \text{(H.6)}$$

where, μ_p is the viscosity of the polymer including the effects of temperature, salinity and shear.

In the special case where the effects of temperature, salinity and shear are absent, the above expression is identical to the *NONLINEAR model in IMEX. In this case, μ_p can be viewed as the "reference viscosity" (*PVISC) at the "reference concentration" (*PREFCONC) in IMEX.

Model 3(*AQUEOUS-VISCOSITY *POLYMER *LINEAR)

Model 3 is identical to Model 2 described above except for the mixing rule which is linear, rather than non-linear:

$$\mu_{aq} = f_{H_2O}.\mu_{H_2O} + \sum_{i=1}^{nc_plm} f_{i,p}.\mu_i + \sum_{i=nc_plm+1}^{nc_aqu} f_{i,np}.\mu_i \quad \text{(H.7)}$$

For an aqueous phase containing only a polymer component (nc_aqu = 1, nc_plm = 1), the viscosity expression reduces to

$$\mu_{aq} = (1 - \alpha_{i,p}) . \mu_{H_2O} + \alpha_{i,p} . \mu_p \quad \text{(H.8)}$$

where μ_p is the viscosity of the polymer including the effects of temperature, salinity and shear.

In the special case where the effects of temperature, salinity and shear are absent, the above expression is identical to the *LINEAR model in IMEX. In this case, μ_p can be viewed as the "reference viscosity" (*PVISC) at the "reference concentration" (*PREFCONC) in IMEX.

Effect of salinity on polymer viscosity

Salinity will generally reduce the viscosity of the polymer solution. Hydrolyzed polyacrylamides are less resistant to salinity than bio polymers.

GEM provides a model to compute the effect of the salinity on polymer solution viscosity. It is given by the following expression:

$$\mu_p = \mu_p^o \left(\frac{x_{NaCl}}{x_{min}} \right)^{sp} \quad \text{for } x_{NaCl} > x_{min} \quad \text{(H.9)}$$

$$\mu_p = \mu_p^o \quad \text{for } x_{NaCl} \le x_{min} \quad \text{(H.10)}$$

where

μ_p^0 is the viscosity of the polymer before applying the effect of salinity (i.e., effects of concentration, temperature and shear are already taken in to account); x_{NaCl} is the concentration of salt, x_{min} is the concentration of salt below which the polymer viscosity is unaffected by salinity and sp is the slope of a log-log plot of polymer viscosity against the ratio

$$\left(\frac{x_{NaCl}}{x_{min}}\right).$$

The effect of salinity is computed only if *VSSALTCMP is specified in the dataset.

The following describes how x_{NaCl} is computed: The total mole fraction of Cl^- contributed by aqueous components other than NaCl is added to the mole fraction of NaCl (which could be zero) and this sum is taken as x_{NaCl}. If there are no Cl^- ions but if Na^+ are present, then total mole fraction of Na^+ is added to the mole fraction of NaCl. If the mole fraction thus computed is zero but *SALINITY keyword is entered, then x_{NaCl} is computed using the data entered for *SALINITY.

Effect of shear on polymer viscosity

For Newtonian fluids, a plot of shear stress versus shear rate is a straight line passing through the origin. The slope of this line is the viscosity of the fluid. Viscosity of a Newtonian fluid is constant and is independent of the shear.

Fluids that do not obey this relationship between shear stress and shear rate are called non-Newtonian. The relationship between shear stress and shear rate can be expressed as

$$\tau = \eta\dot{\theta} \quad \text{(H.11)}$$

In the above expression η is called the apparent viscosity to distinguish it from the Newtonian viscosity which is a constant. η is a function of $\dot{\theta}$.

Many non-Newtonian fluids (including polymers) exhibit Newtonian behavior at the extremes of shear rate – i.e., regions where the apparent viscosity is a constant. On a log-log plot of apparent viscosity against shear rate, these two regions are separated by a region a region where the viscosity is increasing or decreasing linearly with the shear rate. This power-law region can be represented by the following expression:

$$\eta = K\dot{\theta}^{n-1} \quad \text{(H.12)}$$

Substituting for $\dot{\theta}$ from (H.11),

$$\tau = K\dot{\theta}^{n} \quad \text{(H.13)}$$

In the above expression K is called the fluid consistency index and n is the power law exponent.

The quantity (n – 1) in Equation H.11 is the slope of the linear curve in the power-law region on the log-log plot. When n is < 1, the slope is negative and the fluid exhibits shear thinning properties (apparent viscosity decreasing with increasing shear rate) as shown in Figure H-1.

When n > 1, the slope is positive and the fluid exhibits shear thickening behavior (apparent viscosity increasing with increasing shear rate) as shown in Figure H-2. When n is equal to 1, the fluid exhibits Newtonian behavior since $\eta = K$, a constant.

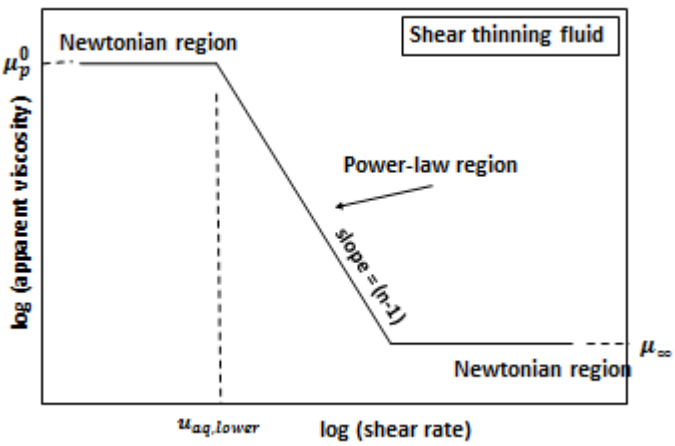


Figure H- 1

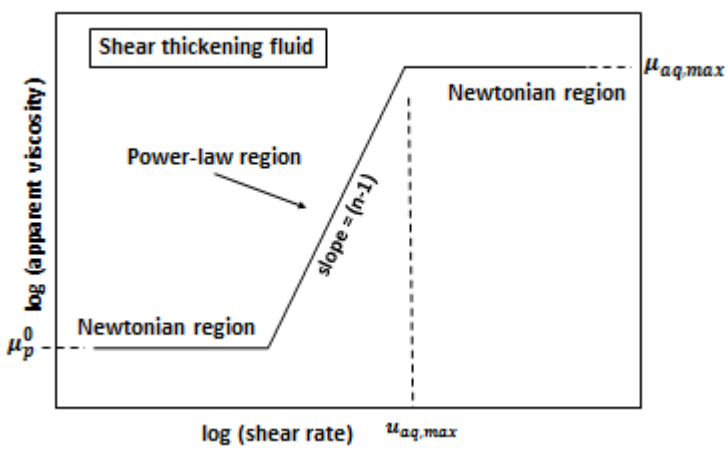


Figure H- 2

In many fluids, the viscosity may decrease and then start increasing as the shear rate is increased, thus exhibiting a combined effect of shear thinning and thickening as shown in Figure H-3.

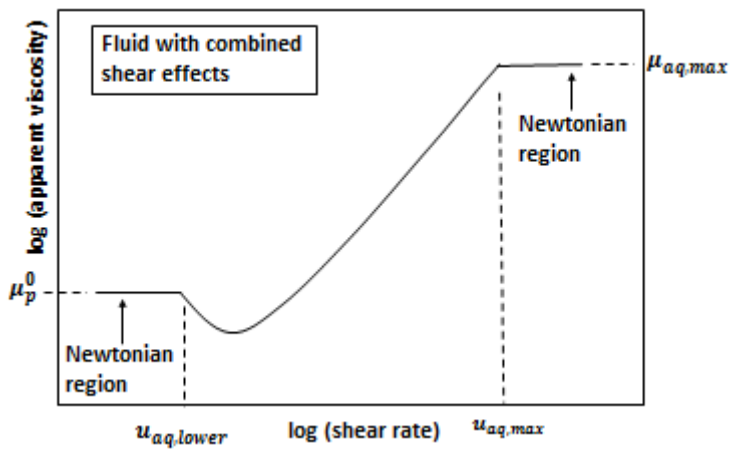


Figure H- 3

Figures H-1, H-2 and H-3 plot changes in apparent viscosity with changes in shear rate. Similar plots can also be obtained in terms of Darcy velocity instead of shear rate. GEM uses the following expression to relate shear rate with Darcy velocity:

$$\dot{\theta} = \frac{\theta_{fac}\left|u_{aq}\right|}{\sqrt{kk_{rw}\phi S_w}} \qquad \text{(H.14)}$$

where,

θ_{fac}	is the shear rate factor which is a user-input (*SHEAR_FAC)
u_{aq}	is the Darcy velocity of the aqueous phase
K	is the absolute permeability
ϕ	is the current porosity
k_{rw}	is the relative permeability to water and
S_w	is the water saturation. Absolute permeability is computed by taking the average of the I-, J- and K-direction permeability values

The shear rate factor is given by the following expression:

$$\theta_{fac} = C\left[\frac{3n+1}{4n}\right]^{\frac{n}{n-1}} \qquad \text{(H.15)}$$

where, C is a constant and n is the power-law exponent.

The following describes the models available in GEM to compute the effect of shear on the aqueous phase viscosity. The description, written for Darcy phase velocity, is equally valid for shear rate.

Shear thinning model (*SHEARTHIN)

For $u_{aq} \le u_{aq,lower}$

$$\mu_p = \mu_p^0 \quad \text{(H.16)}$$

For $u_{aq} > u_{aq,lower}$

$$\mu_p = \mu_p^0 \left[\frac{u_{aq}}{u_{aq,lower}} \right]^{nthin-1} \quad \text{if} \quad \mu_p > \mu_\infty \quad \text{(H.17)}$$

$$\mu_p = \mu_\infty \quad \text{if} \quad \mu_p \le \mu_\infty \quad \text{(H.18)}$$

In the above expressions,

μ_p	is the viscosity of the polymer solution;
μ_p^0	is the viscosity of the polymer solution with zero shear;
μ_∞	is the viscosity of the aqueous phase without the polymer (pure water);
u_{aq}	is the solution velocity; and
$u_{aq,lower}$	$u_{aq,lower}$ is the solution velocity at which its behavior changes from Newtonian to non-Newtonian and
n_{thin}	is the power-law exponent

Shear thickening model (*SHEARTHICK)

For $u_{aq} \ge u_{aq,max}$

$$\mu_p = \mu_{aq,max} \quad \text{(H.19)}$$

For $u_{aq} \ge u_{aq,max}$

$$\mu_p = \mu_p^0 \left[\frac{u_{aq}}{u_{aq,max}} \right]^{nthick-1} \quad \text{if} \quad \mu_p > \mu_p^0 \quad \text{(H.20)}$$

$$\mu_p = \mu_p^0 \quad \text{if} \quad \mu_p \le \mu_p^0 \quad \text{(H.21)}$$

In the above expression, $u_{aq,max}$ is the solution velocity at which its behavior changes from non-Newtonian to Newtonian and nthick is the power-law exponent.

Combined shear model (*SHEARTHIN and *SHEARTHICK)

For $u_{aq} \le u_{aq,lower}$

$$\mu_p = \mu_p^0 \qquad \text{(H.22)}$$

For $u_{aq} \geq u_{aq,max}$

$$\mu_p = \mu_{aq,max} \qquad \text{(H.23)}$$

For intermediate velocities in the combined model:

$$\mu_p = \mu_{thinning} + \mu_{thickening} \quad \text{if} \quad \mu_p < \mu_{aq,max} \qquad \text{(H.24)}$$

$$\mu_p = \mu_{aq,max} \quad \text{if} \quad \mu_p \geq \mu_{aq,max} \qquad \text{(H.25)}$$

Polymer adsorption

GEM provides two modes of data input for polymer adsorption. The first one is a table of polymer adsorption as a function of mole fraction of polymer in the aqueous phase (*ADSTABA). GEM uses this lookup table to interpolate the amount adsorbed for a given mole fraction. In the second mode, Langmuir-type adsorption model is utilized. GEM uses the following extended Langmuir isotherm model:

$$\omega_i = \frac{\omega_{i,max} B_i y_{i_aqu}}{1 + \sum_j B_j y_{j_aqu}} \qquad \text{(H.26)}$$

where

B_i	is the parameter for Langmuir isotherm (user-input)
ω_i	is the amount of adsorbed component
$\omega_{i,max}$	is the maximum amount of adsorbed component i
y_{i_aqu}	is the molar fraction of adsorbed component i in the aqueous phase

Inaccessible pore volume

The inaccessible pore volume (IPV) of a rock for a polymer solution is determined experimentally and applied evenly to all grid blocks irrespective of the relative sizes of the pores and polymer molecules. GEM expects a user input (*PORFT) which is a fraction of the pore volume ***accessible*** to polymer (1 – IPV). The residual of the flow equation for the polymer component for each block is adjusted by multiplying it with the user-input factor.

Permeability reduction

Polymer retention causes a reduction in the effective permeability of the aqueous phase. This reduction is taken into account in GEM for each grid block by dividing the relative permeability of the aqueous phase (k_{rw}) by a permeability reduction factor, F_{kr}:

$$k'_{rw} = k_{rw} / F_{kr} \qquad \text{(H.27)}$$

The minimum value of the factor is 1.0. The maximum reduction corresponds to the adsorptive capacity of the rock ($\omega_{i,max}$). The permeability reduction factor at the current adsorption level

in the grid block (ω_i) is computed by a linear interpolation between the minimum and the maximum values:

$$F_{kr} = 1 + (RRF - 1)\frac{\omega_i}{\omega_{i,max}} \qquad \text{(H.28)}$$

where RRF is the permeability reduction corresponding to $\omega_{i,max}$. RRF is also referred to as "residual resistance factor". In GEM, RRF is a user-input (*RRFT). The dataset must contain the *PEZRM-VS-POR keyword with the *POLYMER sub-keyword.

Residual adsorption level

The extent of reversibility of polymer adsorption can be specified in GEM by the *ADRT keyword. The amount of polymer adsorbed can reduce to this specified level only. If, however, the amount adsorbed was lower than the specified value of ADRT, further reduction will *not* take place.

Polymer degradation

The current version of GEM provides a simplified approach to model polymer degradation. It is useful when information on the products of degradation is not available or when the main interest is in the rate at which the polymer will degrade and in turn affect the aqueous phase properties rather than in the products formed from degradation.

The degradation model assumes an elementary pseudo-reaction in which polymer degrades to form water:

$$\text{Polymer} \rightarrow \left(\frac{MW_p}{MW_{H_2O}}\right) H_2O \qquad \text{(H.29)}$$

where MW_x is molecular weight of x (polymer or H_2O). In this model, the mass balance is maintained although the atom-balance is not for reasons explained above.

The rate expression for the reaction is:

$$r_p = -k_o \,.\, e^{-E/RT} \,.\, C_p \qquad \text{(H.30)}$$

where

r_p	is the rate of polymer degradation
k_o	is the frequency factor of the reaction
E	is the activity coefficient of the reaction
C_p	is the concentration of the polymer in the aqueous phase

The amount of water produced due to this pseudo-reaction will result in a mole balance error. However, the error is expected to be negligible since the concentration of the polymer injected is expected to be very small (typically less than 1200 ppm or 0.12 wt%). In practice, the selected polymer should exhibit low degradation.

GEM offers other keywords to specify a reaction in general. The keywords *POLY-DEGR-FREQFAC and * POLY-DEGR-EACT completely describe the polymer degradation model and no other reactions related keyword is required.

References:

Bondor, P.L., Hirasaki, G.J. and Tham, M.J., "Mathematical Simulation of Polymer Flooding in Complex Reservoirs", SPE Journal, October 1972, pp. 369-382.

Cannella, W.J., Huh, C. and Seright, R.S., "Prediction of Xanthan Rheology in Porous Media", SPE Paper 18089, presented at the 63rd Annual Technical Conference and Exhibition of the Society of Petroleum Engineers held in Houston, Texas, USA, 2-5 October 1988.

Clifford, P.J. and Sorbie, K.S., "The Effects of Chemical Degradation on Polymer Flooding", SPE Paper 13586, presented at the International Symposium on Oilfield and Geothermal Chemistry held in Phoenix, Arizona, USA, 9-11 April 1985.

Green, D.W. and Willhite, G.P., Enhanced Oil Recovery, SPE Textbook Series Vol. 6, Society of Petroleum Engineers, USA, 1998.

Goudarzi, A., Delshad, M. and Sepehrnoori, K., "A Critical Assessment of Several Reservoir Simulators for Modelling Chemical Enhanced Oil Recovery Processes", SPE Paper 163578, presented at the SPE Reservoir Simulation Symposium held in The Woodlands, Texas, USA, 18-20 February 2013.

Pancharoen, M., Thiele, M.R. and Kovscek, A.R., "Inaccessible Pore Volume of Associative Polymer Floods", SPE Paper 129910, presented at the 2010 SPE Improved Oil Recovery Symposium held in Tulsa, Oklahoma, USA, 24-28 April 2010.

Sheng, J.J., Modern Chemical Enhanced Oil Recovery, Elsevier Inc., USA, 2011.

Singleton, M.A., Sorbie, K.S. and Shields, R.A., "Further Development of the Pore Scale Mechanism of Relative Permeability Modification by Partially Hydrolysed Polyacrylamide", SPE Paper 75184, presented at the SPE/DOE Improved Oil Recovery Symposium held in Tulsa, Oklahoma, USA, 13-17 April 2002.

Verma, S., Adibhatla, B., Kaminsky, R., Wattenbarger, C. and Davidson, J., "Modelling Polymer Flood in an Unstructured Grid Simulator", SPE Paper 118985, presented at the 2009 SPE Reservoir Simulation Symposium held in The Woodlands, Texas, USA 2-4 February 2009.

Keyword Index

3

3PTSCAL_KRG 696
3PTSCAL_KROG 696
3PTSCAL_KROW 696
3PTSCAL_KRW 696

A

AC 473
ACTIVATION-ENERGY 561
ACTIVCOEF 568
ACTIVCOEF-SOL 568
ACTIVITY-MODEL 568
ADACSTC 672
ADAMAXC 672
ADGCSTC 665
ADGMAXC 665
ADRT 677
ADS_WATERZONE 549
ADSCST 539
ADSORBTMAX 667
ADSORBTMAXA 674
ADSTAB 667
ADSTABA 674
AI_ALIGN 835
AIM 788
AIMSET 1069
AIMWELL 1071
ALL 168
ALLELEM 161
ALPHAKRG 630
ALTER 1131
ALTER-HIST 1138
ANNIH-MATRIX 564
APPOR-METHOD 1294
AQFILL 568
AQFUNC 380
AQIONS-IEX 574
AQMETHOD 374
AQPROP 374
AQUEOUS-DENSITY 501
AQUEOUS-VISCOSITY 501
AQUIFER 374
ARESOILKRG 630
AUTO_KMP_AFFINITY 828
AUTODRILL 1088
AVISC-AQUEOUS 509

B

BCDOMAIN 1002
BCOEF 893
BG 155, 746
BHPDEFAULT 1118
BHPDEPTH 1210
BHPGRAD 1212
BHPHEADINIT 1144
BIN 477
BINARY_DATA 169
BINS 485
BIOTSCOEF 869
BKRGCL 658
BKRGRL 658
BKROCRW 658
BKROCW 658
BKROGCG 658
BKROGCRG 658
BKRWIRO 658
BKRWRO 658
BLOCKDFL 283
BLOCKGROUP 388
BLOCKS 283
BO 746
BSGCON 650
BSGCRIT 650
BSOIRG 650
BSOIRW 650
BSORG 650
BSORW 650
BSWCON 650
BSWCRIT 650
BVISC-AQUEOUS 509

C

CACHE_ALIGN 833
CALIB_POR 1013
CASEID 196
CCPOR 410
CDEPTH 763

CEC-CALC 663
CEC-IEX 662
CER-REDUCTION 831
CHARGE-AQUEOUS 555
CHECKONLY 144, 197
CHECKRB 807
CHEM-EQUIL-SET 565
CIRREVERS 416
CKNJ 926
CKSJ 926
CKSPACING 926
CLUMPSETTING 1387
CO2-AQU-FACT 570
CO2-MNR-FACT 570
COAL-DIF-COMP 682
COAL-DIF-TIME 682
COHESEXPNT 909
COHESHARD 909
COHESION 869, 886, 893, 902
COHESIONMAP 931
COMMAND-LINE-ARG 178
COMMENT 147
COMPNAME 466
COMPNAME-AQUEOUS 555
COMPNAME-IEX 574
COMPNAME-MINERAL 556
COMPNAME-POLYMER 577
COMPNAME-SAL 501
COMPNAME-SURFACTANT 578
CON 162
CONCENTRATION-ORDER 557
CONVERGE 791
CONVERT-TO-CORNER-POINT 264
CONV-RESONLY 791
COORD 293
CORNERS 295
CORNER-TOL 387
CPOR 339
CP-ROCK 452
CPRPOR 410
CPTPOR 459
CRAKSET 926
CRAKTYPE 926
CRIT 479
CRITDELD 583
CRL_WELLS 1389
CRNCON_NODE 288
CROCK 624
CROCKTAB 411
CROCKTABC 423
CROCKTABD 419
CROCKTABE 417
CROCKTABH 413
CROCKTABR 425
CROCKTABU 421
CROCKTYPE 407
CRVWGT 594
CTPOR 459
CTYPE 435
CUBIC-ROOT-SELECTION 553
CVFANGLE 909
CW 501
CYCLPROD 1085
CYCPRT_END 1373
CYCPRT_START 1373

D

DATE 1065
DATUMDEPTH 766
DBL-WBLOCK-COAL 546
DCPOR 339
DEBUG 251
DENSTR-KEY 753, 1169
DENSTR-MIX 753, 1169
DENW 501
DENWS 501
DEPLETION 433
DEPTH 273
DEPTH-TOP 279
DER-CHEM-EQUIL 560
DERIVATIVEMETHOD 531
DER-REACT-RATE 560
DFRICANGLE 893
DGOC 742
DI 267
DIARY 198
DIARY2 198
DIFCOR-AQU 533
DIFCOR-GAS 533
DIFCOR-OIL 533
DIFFC-AQU 533
DIFFC-GAS 533
DIFFC-OIL 533
DIFFUSION 526
DIFRAC 332
DILANGLE 869
DILFACTOR 926
DIM 189
DIP 281
DISPERARRAY-ISO 680
DISPERARRAY-LNG 680
DISPERARRAY-TRN 680
DISPLACTOL 952
DJ 269
DJFRAC 332
DK 271

DKFRAC 332
DLOADBC 984
DLOADBC2D 984
DLOADBC3D 988
DLOADIJK 988
DPLANES 816
DRILLALTP 1287
DRILLMXW 1281
DRILLQ 1281
DRILLQTALTP 1290
DRILLQTREFP 1290
DRILLT 1281
DRNRADW 1285
DRUCKER 883
DTMAX 784, 1068
DTMIN 784, 1068
DTOP 275
DTRAPN 596
DTRAPW 596
DTWELL 1067
DUALPERM 321
DUALPOR 320
DWGC 742
DWOC 742
DYNAGRID 1394
DYNAGRID-IGN-DIR 1394
DYNAGRID-TINT 1394
DYNAGRID-TSINT 1394
DYNAGRID-WLN-A 1394
DYNAGRID-WLN-V 1394

E

ECOEF 893
EFCAP 608
EFOIL 608
EFSALT 608
EFSURF 608
ELASTMOD 869, 886, 891, 902, 909, 915
END-GRID 458
ENTHALPY-MIXING 582
ENTHCOEF 581
EOSSET 468
EOSTYPE 472
EPCAP 603
EPGCP 603
EPOIL 603
EPOMF 603
EPSALT 603
EPSCUTOFF 909
EPSPC 625
EPSPCG 625
EPSURF 603
EQUALSI 171
EQUILIBRATE 1086
EQUIL-REACT-RATE 565
EXPN1 893
EXPN2 893
EXPPM 441

F

FAULT 372
FAULTARRAY 405
FI_ALIGN 834
FILENAMES 184
FLASH–METHOD 528
FLASH–METHOD–OGW 528
FLOIL 603
FLSALT 603
FLUX_ISECTOR 396
FLUX_SECTORNAMES 396
FMCAP 603
FMCOMP 599
FMGCP 603
FMKRW 611
FMMOB 603
FMOIL 603
FMOMF 603
FMPERM1 603
FMPERM2 603
FMSALT 603
FMSURF 603
FOAM-MODEL 602
FORCETOL 952
FPVOLM 949
FRACANGLE 946
FRACTURE 153
FRACVOL 324
FRANGSOFT 909
FRATIO 893
FREE-DIFFUSION-COAL 546
FRICANGLE 869, 886, 893, 902, 921
FRICANGMAP 931
FRICANGMN 893
FRICANGMX 893
FRICHARDEN 909
FYOUNGAF 893
FZ 159

G

GAMALI 859
GAMALJ 859
GAMALK 859
GAMMA 891

GAPPOR 1292
GASZONE 773
GAUSSPNT 952
GCAPD 886
GCAPLOC 886
GCAPMAT 886
GCAPMOD 886
GCAPMODEL 885
GCAPR 886
GCAPTEN 886
GCAPW 886
GCFACTOR 1009
GCINCRMT 886
GCIOFF 1305
GCION 1305
GCONCYCLE 1371
GCONCYCLE_END 1358
GCONCYCLE_START 1358
GCONCYCR_END 1369
GCONCYCR_START 1369
GCONI 1221
GCONIINCR 1356
GCONIMULT 1356
GCONM 1265
GCONP 1216
GCONPINCR 1354
GCONPMULT 1354
GCOUPLING 1005
GCPOFF 1305
GCPON 1305
GCUPDATE 1011
GDI 848
GDJ 848
GDK 848
GELEVATB 862
GELEVATT 862
GENPLAST 909
GEOCHEM-V2 557
GEODEPTH 848
GEODOMAIN 1000
GEOEXTD 862
GEOGRID 848
GEOM3D 847
GEOMECH 846
GEOMETRY 1180
GEOPOSTFREQ 1018
GEOPOSTINPUT 1023
GEOPOSTPRO 1018, 1021
GEORBLOCK 967
GEOROCK 868
GEOSOLID 879
GEOTYPE 868
GEXDI 862
GEXDJ 862
GEXDK 862
GEXMAXI 862
GEXMAXJ 862
GEXMAXK 862
GEXMINI 862
GEXMINJ 862
GEXMINK 862
GEXPONENTN 891
GFRACBLK 942
GFUELFRAC 1231
GFUELMASK 1233
GFUELMAXR 1235
GLCONTROL 1158
GLIFT 1155
GLOADBC 993
GLOADBC3D 995
GLOPT 1160
GMCREEP 900
GMKCOMP 1237
GMKMAXR 1239
GMKUPTO 1241
GMONETONE 848
GNULL 848
GOC_PC 768
GOUTSRF 855
GPATM 893
GPERMBB 942
GPERMES 933, 1008
GPERMLC 933, 1008
GPERMNEX 939
GPERMNEX_MT 933
GPERMNEY 939
GPERMNEY_MT 933
GPERMNEZ 939
GPERMNEZ_MT 933
GPERMNTX 939
GPERMNTX_MT 933
GPERMNTY 939
GPERMNTY_MT 933
GPERMNTZ 939
GPERMNTZ_MT 933
GPERMTS 933, 1008
GPERMVL 933
GPHYDDEP 1275
GPOLY 848
GPRODGROUP 1243
GPTABLE 1273
GPTOLMUL 1010
GRECYMASK 1247
GRECYMAXR 1249
GRID 256
GRMPRSTAB 877
GROTATEI 848

GROTATEJ 848
GROTATEK 848
GROUP 1076
GROUPALQ 1277
GRPORTAB 873
GRPRESTAB 875
GRSTRESTAB 877
GRTEMTAB 871
GRTHRDS 823
GSALESFRAC 1251
GSALESMASK 1253
GSALESMAXR 1255
GTRANSLI 848
GTRANSLJ 848
GTRANSLK 848
GUIDEI 1302
GUIDEP 1302
GULBULKMOD 891
GULOGINT 933

H

H2O_INCLUDED 554
HARDEN 869
HCFLAG 473
HEADITER 1142
HEAD-METHOD 1140
HEATING_VALUES 482
HEAT-LOSS 453
HENRYC 522
HENRY-CORR-C1 522
HENRY-CORR-CO2 522
HENRY-CORR-H2S 522
HENRY-CORR-N2 522
HLPROP 453
HYS_MODEL 630
HYSKRG 625
HYSKRGM 625
HYSKROG 625
HYSKROM 625
HYSKROW 625

I

IFTMISPT 638
IFTTABLE 600
IJK 163
IJPJ 926
IKPJ 926
IMPLICIT-SHEAR 520
INACTIVE-WELL-SOLVE 830
INCOMP 1105
INCOMPWL 1119
INHERIT-SOLID-OFF 545
INITFANGLE 909
INITIAL 727
INJECTOR 1085
INJECTOR-EOS 757, 1178
INJECTOR-P 757, 1178
INJECTOR-T 757, 1178
INJ-SALIN 1391
INJ-TEMP 1393
INT 176
INTCOMP 594
INTCOMP_VAL 596
INTER_REGION_TM 367
INTERP_ENDS 716
INTERRUPT 192
INTLIN 600
INTLOG 600
INTPOWER 848
INUNIT 201
INVENTORY-CO2 248
ION-SIZE-AQUEOUS 555
IRCONNECT 398
ISECTOR 395
ITERCER 832
ITERMAX 803
ITERMAXG 956
ITUBE1 1318
ITYPE 734
IVAR 165
IWELLBORE 1092

J

JACPAR 820
JDUMPG 956
JFGMAX 658
JFUNC 655
JFW_SHIFT 661
JFWMAX 658
JLOADBAL 818
JLOAD-CHIMP 818
JVAR 166

K

KDIR 263
KLINKENBERG 690
KRCBM 685
KRGAS 646
KRINTRP 596
KROIL 647
KRPERF 1202
KVAR 167

L

LANG-DIFFUSION-COAL 546
LANGMULT 695
LAYERCLUMP 1375
LAYER-CTRL 1316
LAYERGRAD 1207
LAYERIJK 1200
LAYERLUMP_ACTR 1388
LAYERRLUMP 1381
LAYERXYZ 1196
LAYERXYZ-METHOD 1196
LBPRINT 203
LCSPRINT 204
LIST 146
LOG-CHEM-EQUIL-COEFS 561
LOG-CHEM-EQUIL-CONST 561
LOG-SCOEFF-IEX 574
LOG-TST-RATE-CONSTANT 561
LOG-TST-RATE-CONSTANT-ACT IV 561

M

MAGNIFYDI 952
MAGNIFYDJ 952
MAGNIFYDK 952
MANIFOLD 1269
MASSDENSITY-MINERAL 556
MATRIX 152
MAXCHANGE 785
MAXCLK 783
MAXCPU 783
MAXERROR 206
MAXSTEPS 782
MCCMODEL 921
MCETHA 921
MCOCR 921
MCOEF 893
MCOMINDX 921
MCONNG 959
MCPREHD 921
MCRCSL 921
MCSWINDX 921
MDICLU_PG 959
MDPLNRBK 310
MINC 328
MIN-REACT-SURFACE-AREA 561
MIXINGRULE 497
MIXVC 497
MOD 172
MODEL 464
MODELSHUT 1086
MODILU 802
MOHRCOUL 883
MOLALITY-AQUEOUS 777
MOLALITY-AQUEOUS-PRIMARY 777
MOLALITY-GASEOUS 777
MONITOR 1123
MPLNE 959
MRC-RESET 1153
MW 473
MW-AQUEOUS 555
MW-MINERAL 556
MXCNRPT 1122

N

NB 893
NC 465
NC-AQUEOUS 555
NCHECK-CEQ 794
N-CHEM-EQUIL 557
NC-IEX 574
NC-MINERAL 556
NCOUPLING 1015
NDARCYCOR 693
NE 893
NETGROSS 355
NETPAY 353
NEWTONCYC 795
NFZ 159
NINCS 952
NINEPOINT 265
NITERGEO 952
NLINEAR 890
NNODES 283
NOCOUPERM 1005
NODE4 952
NODE8 952
NODE9 952
NODES 283
NODEX 283
NODEY 283
NODEZ 283
NOINTERMF 403
NOLIST 146
NONDARCY 690
NORM 785
NORTH 798
NORTHG 956
NO-WBLOCK-COAL 546
N-RATE-REACT 557
NREGIONS 733
NTB 893
NTE 893
NULL 334

NULL-PERF 1146
NUMERICAL 781
NWHYFS 1103

O

OGW_FLASH 554
OMEGA 476
OMEGAS 485
OMEGB 476
OMEGBS 485
ON-TIME 1307
OPEN 1088
OPENMP_SCHEDULE 825
OPERATE 1107
OPERATE-HIST 1136
ORTHOGG 956
OUTPRN 215
OUTREACT 249
OUTSRF 231

P

PALMAN_ARC_COAL 551
PALMAN_CMG_COAL 551
PALMAN_CMG2_COAL 551
PARTOL-TOL 387
PAYDEPTH 277
PCGMAX 658
PCHOR 473
PCON-MANIFOLD 1271
PCOW_SHIFT 661
PCRIT 473
PCRITS 485
PCWMAX 658
PDEGAA 808
PDEGAB 809
PEAKANGLE 909
PEAKCOHES 909
PERF 1183
PERFV 1183
PERMI 351
PERMJ 351
PERMK 351
PGDILA 906
PGPACT 906
PGPDMAX 906
PGTHRDS 824
PHASEID 489
PINCHOUTARRAY 383
PINCHOUT-TOL 385
PIVOT 801
PIVOTG 956
PLANT-TABLE 759, 1174
PLNR_REFINE 310
PLNRFRAC 314
PLNRFRAC_TEMPLATE 314
PLOADBC 978
PLOADBC3D 982
PLSTRAINY 867
PNPROSL 806
PNTHRDS 822
POISSONMAP 931
POISSR 441
POISSRATIO 869, 886, 891, 893, 902, 909, 915, 921
POLY-DEGR 573
POLY-DEGR-EACT 571
POLY-DEGR-FREQFAC 571
POLY-VISC-SAL 573
POLY-VISC-SHEAR 573
POR 336
PORFT 677
POROSTOL 1015
PORVOL 338
PPATTERN 810
PRECABG 956
PRECC 797
PRECCG 956
PREFCONC 521
PRES 738
PRESCBC 968
PRESCBC3D 974
PRESLN 441
PRESSTOL 1015
PRINTGEO 952
PRIOR-FORM 1298
PRODUCER 1085
PRPOR 339
PSAT 492
PTSCALING 696
PTUBE1 1318
PVC3 478
PVC3S 485
PVCUTFR 382
PVCUTOFF 382
PWELLBORE 1095

Q

QNSS-TO-NR 528

R

RANGE 298
RANGECHECK 205
RCONBK 848
RCONBT 848

RCONFT 848
RCONLF 848
RCONRT 848
RCONTP 848
REACTION-CHEM 557
REACTION-IEX 574
REACTION-RATE 557
REACTIVE-SURFACE-AREA 561
RECYCLE_GEM 1214
REFDEN 489
REFDEPTH 741
REFINE 298
REFPH 522
REFPW 501
REF-TEMP-RATE-CONST 561
REPORTING-GROUP 1080
RESDATE 211
RESDUCOHES 909
RESISTIVITY 533
RESTART 211
RESTIME 211
RESULTFILE 223
REWIND 208
RF_EXPONENT 579
RFCALC 579
RFMAX 539
RG 154
RHOW 501
RHOWS 501
RIGIDNULL 997
RIGIDTOP 999
RM-LAYERCLUMP 1383
RM-LAYERRLUMP 1385
ROCKDEN 679
ROCKFLUID 591
ROCKJOINT 926
RPLTABD 947
RPT 592
RPWTABD 947
RRFT 677
RS 746
RTYPE 636
RUN 1064

S

SA1 915
SALINITY 501
SALINITY-CALC 568
SALINR 777
SATWCUTOFF 554
SBETA 915
SBG 157
SBLOCKGROUP 390
SCALING-STARS 694
SCOEFF-IEX 574
SCONNECT 401
SD_REINF 686
SDBIOTSCF 879
SDCOHES 879
SDEGREE 800
SDEGREEG 956
SDELASTMOD 879
SDFRICANG 879
SDILANG 879
SDPOISSON 879
SDTHEXPCF 879
SECTOR 392
SECTORARRAY 394
SECTORNAMES 395
SEEKPARENT 253
SEPARATOR 747
SETHA1 915
SETPI 1311
SFBET 608
SFCAP 608
SFDRY 608
SFOIL 608
SFSALT 608
SFSURF 608
SG 473
SGAMMA 915
SGT 618
SHAPE 329
SHEAR_FAC 515
SHEAREFFEC 516
SHEARMOD 921
SHEARTAB 519
SHEARTHICK 517
SHEARTHIN 517
SHUTIN 1088
SIGMA 638
SITERPG 956
SKAPA1 915
SKAPA2 915
SLCON 650
SLOIL 608
SLSALT 608
SLT 618
SM 915
SN 915
SNG 915
SOLID_ALPHA 543
SOLID_BETA 543
SOLID_CRITVEL 543
SOLID_GAMMA 543
SOLID_SIGMA 543
SOLID_TREF 537

SOLID-CONV-RATE 541
SOLIDMODEL 537
SOLIDTHERMAL 537
SOLUBILITY 522
SOLVER 805
SOLVERG 956
SOMGAS 709
SOMGASI 709
SOMWAT 709
SOMWATI 709
SORDER 799
SORDERG 956
SORM 637
SPATM 915
SPECGRAV 993, 995
SPEC-REACT-SURFACE-AREA 561
SR2ASCII 224
SR2PREC 225
SRFORMAT 226
SRFTNG 655
SRFTNW 655
SSMODEL 915
STEN 915
STIFFCOM1 951
STIFFCOM2 951
STIFFINIT 951
STIFFTANG 951
STOICH-IEX 574
STOICHIOMETRY 557
STOP 1402
STRESI 960, 962
STRESJ 960, 962
STRESK 960, 962
STRESS 960
STRESS_RETURN 952
STRESS3D 962
STRESSALL 960
STRESSGRAD 960
STRESSGRAD3D 962
STRESSH 960
STRESSHIJ 962
STRESSHIK 962
STRESSHJK 962
STRESSTOL 1015
STRINF 441
SUBDOMAIN 322
SUMMARY 250
SUPER-SATURATION-INDEX 561
SW 738
SWINIT 770
SWNEQ 772
SWOC 745
SWR-H2OVAP 554
SWT 612
SWT3 630

T

TARGET 1133
TB 473
TCMULT 775
TCRIT 473
TCRITS 485
TDMAX 949
TDMIN 949
TEMPER 740
THCONR0 452
THERMAL 581
THEXPCOEF 930
THEXPMAP 931
TIME 1066
TITLE1 193
TITLE2 194
TITLE3 195
TKRGCL 704
TKRGRL 704
TKROCRW 704
TKROCW 704
TKROGCG 704
TKROGCRG 704
TKRWIRO 704
TKRWRO 704
TORNOPS2DIFF 533
TORNOPSDIFF 533
TORPSDIFF 533
TRACE-COMP 552
TRANLI 361
TRANLJ 361
TRANLK 361
TRANS_MULT_REGION 366
TRANSD 326
TRANSF 447
TRANSFER 331
TRANSI 356
TRANSIJ- 359
TRANSIJ+ 359
TRANSIK- 359
TRANSIK+ 359
TRANSJ 356
TRANSK 356
TRANSLATE 148
TRANSMF 363
TRANSMULT 364
TRANVI 369
TRANVJ 369
TRANVK 369
TREFVS 480

TRES 487
TRESCA 883
TRIGGER 1326
TRLIMS 688
TRPOR 459
TSGCON 699
TSGCRIT 699
TSLCON 699
TSOIRG 699
TSOIRW 699
TSORG 699
TSORW 699
TST-ACTIVITY-POWER 561
TST-PARAM1 561
TST-PARAM2 561
TSWCON 699
TSWCRIT 699
TWELLBORE 1100
TWOPTFLUX 804

U

UBA 149
UNDERRELAX 796
UNFIXFDIR 942
UNI-WBLOCK-COAL 546
UNLOADSTR 1017
URBCOEF 893
URECOEF 893
UREXPN1 893
UREXPN2 893
URNB 893
URNE 893
URNTB 893
URNTE 893
USER_INPUT 728
USTRESCUR 933

V

VAMOD 343
VATYPE 343
VCRIT 473
VELDEPRP 641
VERTICAL 728
VINFINITY 522
VISCOEFF 497
VISCOR 493
VISCOSITY 497
VISCTABLE-AQUEOUS 509
VISFLOWR 902
VISINIT 902
VISPARA 902
VISPOWER 902
VISSCHEME 902
VISSTEP 902
VISTIME 902
VISVC 497
VISW 501
VOLMOD 341
VOLUMEFRACTION-MINERAL 777
VONMISES 883
VRI_GROUP 1245
VSHIF1 480
VSHIFT 480
VSHIFTS 485
VSMIXENDP 512
VSMIXFUNC 512
VSSALTCMP 514

W

WCURRCN 1121
WEL_OPN_MLT 449
WEL_OPN_MMN 449
WEL_OPN_MMX 449
WEL_OPN_THR 449
WELGEO 1194
WELL 1082
WELLALQ 1279
WELLINIT 1073
WELPRN 1400
WELSEP 1163
WELSEP-ICPRT 207
WGC_PC 768
WHPCRTB 1101
WIRANGE 1192
WLISTOPEN 1090
WLISTSHUT 1090
WMKMAXR 1257
WMKUPTO 1259
WOC_PC 768
WORKOVER-DATA 1150
WPRN 213
WPRODGROUP 1261
WRADIUS 950
WRECYMAXR 1263
WRST 208
WSLMAX 539
WSRF 228
WTINCR 1350
WTMULT 1350

X

XCORN 291
XFLOW-MODEL 1148

Y

YAQU-RATE-CUTOFF 560
YCORN 291
YLDSTRESS 869, 902
YOUNGM 441
YOUNGMAP 931

Z

ZCORN 289
ZCRIT 473
ZDEPTH 764
ZGAS 735
ZGLOBAL 738
ZGLOBALC 738
ZOIL 735