



IMEX USER GUIDE

THREE-PHASE, BLACK-OIL RESERVOIR SIMULATOR

VERSION 2015

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 © 2015 Computer Modelling Group Ltd.

All rights reserved.

The license management portion of this program is based on:

Reprise License Manager (RLM)

Copyright © 2006-2015, Reprise Software, Inc.

All rights reserved.

IMEX uses Intel(R) Compilers.

IMEX, 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

IMEX is CMG's new generation adaptive implicit-explicit black-oil simulator which includes features such as local grid refinement, comprehensive well management, pseudo-miscible option, volatile oil option, polymer flooding, horizontal wells, dual porosity/permeability, flexible grids, and many more. IMEX was developed to simulate primary depletion, coning, water, gas, solvent, and polymer injection in single and double porosity reservoirs.

This User Guide provides a step-by-step procedure for preparation of an input data set for this program. A tutorial section is provided as well as a set of appendices describing the underlying theory. Use of this User Guide requires a basic knowledge of reservoir engineering and some exposure to reservoir simulation.

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 between IMEX 2015.10 and IMEX 2014.10.....	1
Important Changes between IMEX 2014.10 and IMEX 2013.10.....	4
Important Changes between IMEX 2013.10 and IMEX 2012.10.....	7
Important Changes between IMEX 2012.10 and IMEX 2011.10.....	10
Important Changes between IMEX 2011.10 and IMEX 2010.10.....	14
Important Changes between IMEX 2010.10 and IMEX 2009.10.....	19
Important Changes between IMEX 2009.10 and IMEX 2008.10.....	22
Important Changes Between IMEX 2008.10 and IMEX 2007.10.....	24
Important Changes Between IMEX 2007.10 and IMEX 2006.10.....	26
Important Changes Between IMEX 2006.10 and IMEX 2005.10.....	30
Important Changes Between IMEX 2005.10 and IMEX 2004.10.....	35
Important Changes Between IMEX 2004.10 and IMEX 2003.10.....	37
Important Changes Between IMEX 2003.10 and IMEX 2002.10.....	39
Important Changes Between IMEX 2002.10 and IMEX 2001.10.....	42
Important Changes Between IMEX 2001.10 and IMEX 2000.10.....	46
Important Changes Between IMEX 2000.10 and IMEX 1999.10.....	49
Important Changes Between IMEX 1999.10 and IMEX 98.05.....	54
Important Changes Between IMEX 98.05 and IMEX 98.00.....	57
Important Changes Between IMEX 98.00 and IMEX 97.00.....	58
Important Changes Between IMEX 97.00 and IMEX 96.00.....	63
Important Changes Between IMEX 96.00 and IMEX 95.00.....	68
Important Changes Between IMEX 95.00 and IMEX 94.00.....	72
Important Changes Between IMEX 93.00 and 92.00.....	76
Introduction to IMEX	80
Tutorial	85
Introduction.....	85
Data Groups in the Keyword Input System	86
How to Document Your Data Set	87
How to Do a Restart.....	88
Controlling Contents of the Output File	90
Controlling Contents of the Graphics File (SR2).....	91
Describing Your Grid System.....	92

Describing Refined Grid	94
Using Dual Porosity/Dual Permeability	95
Input of Null Blocks	99
Using the Aquifer Option (see Appendix G).....	101
Using the Pseudo-Miscible Option.....	103
Using the Rock Compaction/Dilation Model Option	104
Using the Polymer Option.....	112
Problems with Small Timesteps or Long Execution Times	113
Using the API Tracking Option.....	116
Problems with Solver Convergence	117
Simulating a Gas-Water or Gas Reservoir	118
Simulating a Gas-Water with Condensate Reservoir	119
Simulating a Tilted Water-Oil/Water-Gas Contact	121
Horizontal Wells.....	125
Vertical Equilibrium Calculation	126
Defining Multiple PVT Regions/Initialization Regions	130
Defining Wells	135
Defining the Well Type	136
How to Shut In a Well and Reopen It.....	137
Voidage Replacement.....	139
Using the Wellbore Model	140
Operating and Monitoring Constraints	142
Definition of Well Indices within IMEX.....	144
Input of Well Indices	146
Stopping a Simulation Run.....	148
Using Wildcards in Well Lists	149
Guidelines for Setting Up Well Data.....	150
Recurrent Data from Other Sections	153
Subsidence Output.....	155
Parallel IMEX.....	157
Using the Seawater Injection Option.....	159
Using the Scale Buildup Option	160
Simulating a Volatile Oil Reservoir	162
Using the Subdomain-Dual Permeability-Reinfiltration Model.....	163
Modelling Non Darcy Flow in Hydraulic Fractures Accurately Using a Grid Based Approach	165
Using the Fracture Clean-up Model	171
Using the Oil Wet Option.....	175
Special Keywords for Advanced Users (SHIFT)	182
Modelling Non Darcy Flow in Fractured Shale Gas Reservoirs	183
The WATER_FIX Keyword	196
Introducing a Hydraulic Fracture in Recurrent Data	197
Relative Permeability and Capillary Pressure Hysteresis in Oil-Water Systems	198
Two and Three Point Scaling of Relative Permeability Values	205

Keyword Data Entry System 209

Introduction to Keyword System.....	209
Comments (Optional)	215
Blank Lines (Optional)	216
Scan Mode for Checking Errors	217
Include Files (Optional)	218
Controlling Data File Listing (Optional)	219
Controlling Data File Listing Using NOLISTLIM (Optional)	220
Changing the Comment Indicator (Optional)	221
Changing the Keywords by Using Translate Rules (Optional).....	222
Input of Grid Property Arrays.....	223
Entering Matrix Grid Properties	225
Entering Fracture Grid Properties.....	226
Entering Refined Grid Properties.....	227
Assign Data to Block Groups	228
J and K Direction Data from I Direction	230
Constant Value Arrays.....	231
Array Input in IJK Notation.....	232
Array Input of Values that Vary in the I Direction	234
Array Input of Values that Vary in the J Direction.....	235
Array Input of Values that Vary in the K Direction	236
Values that Vary for Most or All Grid Blocks.....	237
Assign Property to Fracture Zone (Conditional).....	238
Modifying Array Data (Conditional)	240
Interpolating Table Data (Optional).....	243

Input/Output Control 245

Command Line Input/Output Control.....	245
Values Stored in Binary Form	249
ASCII Characters.....	251
Input/Output File Names (Optional).....	252
Important Notes (Run Time Dimensioning)	256
Problem Type Where IMEX Fails to Allocate Adequate/ Appropriate Storage	257
Run Time Dimensioning (Optional)	258
Background (Run Time Dimensioning).....	263
Advantages of Run Time Dimensioned IMEX.....	264
Project Main Title (Optional).....	265
Project Second Title (Optional)	266
Project Third Title (Optional)	267
Case Identification (Optional).....	268
Check Only (Optional).....	269
Trap UNIX Signal 2 Interrupt (Optional)	270
ASCII Form of SR2 (Optional).....	272

Precision of SR2 (Optional)	273
Controlling Data File Listing Using NOLISTLIM (Optional)	274
Input Data Units (Optional)	275
Output Data Units (Optional)	278
Data Range Checking (Optional)	279
Maximum Number of Error Messages (Optional)	280
Restart Timestep (Optional)	281
Separate or Combined Restart/Graphics (Optional)	283
Restart Record Writing Frequency (Optional)	284
Output Printing Frequency (Optional)	286
Items in Output Print File (Optional)	289
Simulation Results File Writing Frequency (Optional)	297
Items in Simulation Results File (Optional)	299
Items in Diary File (Optional)	306
Production Split Output (Optional)	308
Debug Output (Optional)	309

Reservoir Description 313

Fundamental Grid Definition (Required)	313
K Direction Index (Optional)	320
Block Dimensions for the I Direction (Conditional)	322
Block Dimensions for the J Direction (Conditional)	324
Block Dimensions for the K Direction (Conditional)	326
Depth to the Centre of a Grid Block (Conditional)	328
Depth to the Tops of Grid Blocks (Conditional)	330
Depths to Centre of Pay (Conditional)	332
Depths to Top of Block (Conditional)	334
Grid Tilt Angles (Conditional)	336
Corner Point Depths for Corner Point Grids (Conditional)	338
Lateral Corner Point Locations for Corner Point Grids (Conditional)	340
Line-Based Corner Point Locations for Corner Point Grids (Conditional)	342
Complete Corner Point Locations for Corner Point Grids (Conditional)	344
Corner-point Node Locations (Conditional)	347
Convert Corner Point Grid to Node Based Grid (Conditional)	352
Block Groups (Optional)	353
Local Refined Grid (Optional)	355
Planar Fracture Template	367
Dual Porosity (Optional)	371
Dual Permeability (Optional)	372
Dual Porosity Subdomain Method (Optional)	373
Assigning Fractional Volumes to Subdomain blocks (Optional)	375
Subdomain to Subdomain Flow Transmissibility Multipliers (Optional)	377
Dual Porosity MINC Method (Optional)	379

Shape Factor Calculation (Conditional).....	380
Matrix-Fracture Transfer Calculation (Conditional).....	382
Allowing Addition Matrix-Fracture Connections (Optional)	384
Fracture Spacing (Conditional).....	385
Null Block Indicator (Optional).....	386
Porosity (Required).....	388
Rock Compressibility (Required)	389
Pore Volume Modifiers (Optional).....	391
Block Geometry Modifiers (Optional).....	393
Permeabilities (Required)	400
Netpay (Optional)	402
Netgross (Optional).....	404
Transmissibility Multipliers (Optional)	405
Transmissibility Multipliers for Lower Indexed Block Faces (Optional).....	408
Transmissibility Multiplier for Matrix-Fracture Flow (Optional)	410
Transmissibility Multiplier Action (Optional).....	411
Inter Region Transmissibility Multiplier (Optional).....	413
Transmissibility Multiplier Regions (Optional).....	415
Pinch Out Array (Optional).....	416
Pore Volume Cut-Off Threshold (Optional).....	418
Pinchout Tolerance (Optional).....	419
Corner Point Tolerance (Optional).....	421
Faults (Optional)	422
Aquifers (Optional) see Appendix G	424
Pressure Influence Function (Conditional) see Appendix G	431
Sectors (Optional)	433
Sector Array (Optional)	435
Sector Assignment via Names and an Array (Optional)	436
Flux Sector (Optional)	438
Lease Planes (Optional)	440
Irregular Connection (Optional)	442
Special Connections (Optional)	445
Fault Array (Optional)	447
Compaction/Dilation Rock Type (Optional).....	449
Compaction/Dilation Rock Compressibility (Optional)	451
Compaction Rock Table (Optional).....	452
Compaction Hysteresis Rock Table (Optional)	455
Treat *CCPOR/*CROCKTAB/*CROCKTABH as Depletion Parameters/ Tables (Optional)	458
Compaction Irreversibility Flag (Optional)	460
Dilation Rock Table in Elastic Zone (Optional)	461
Dilation Rock Table in the Plastic Zone (Optional)	463
Dilation Rock Table in Unloading Zone (Optional)	465
Dilation Rock Table in Recompacting Zone (Optional).....	468

Dilation Rock Table in Reloading Zone (Optional)	470
Compaction/Dilation Rock Region (Optional).....	478
Dispersion Coefficients (Conditional).....	485
Fault Transmissibilities (Optional).....	487
Component Properties	489
Fluid Model (Required).....	489
Reservoir Temperature (Optional)	493
Oil and Gas PVT Table (Optional).....	494
Oil and Gas Differential Liberation Table (Optional)	499
Gas PVT Table for API Model (Optional)	505
Oil PVT Tables for API Model (Optional)	508
Gas PVT Table for GASWATER Model (Optional)	513
Condensate - Oil and Gas PVT Table (Optional).....	516
Volatile - Oil and Gas PVT Table (Optional)	521
Oil Compressibility as a Function of Pressure (Optional).....	527
Bo Above the Bubble Point as a Function of Pressure (Optional)	530
Oil Viscosity Above the Bubble Point as a Function of Pressure (Optional).....	532
Bo above the Bubble Point as a Function of Pressure and STC Oil Density for API Tracking (Optional)	534
Condensate Model - Undersaturated Eg, Bg, Zg Table.....	539
Condensate Model - Undersaturated Gas Viscosity Table	543
Oil Viscosity Above the Bubble Point as a Function of Pressure and STC Oil Density for API Tracking (Optional)	546
Reference Pressure for Gas-Oil Capillary Pressure (Optional)	551
Solvent PVT Table (Conditional).....	552
Extended Oil-Solvent Mixing Parameter Table (Optional).....	556
Densities (Required).....	559
Under-Saturated Oil Compressibility (Optional)	561
Oil Phase Viscosity Pressure Dependence (Optional).....	562
Water Formation Volume Factor (Required)	563
Water Phase Viscosity (Required)	565
PVT Type (Optional).....	567
Maximum Rate of Increase of Solution Gas Ratio or Bubble Point Pressure (Optional).....	568
Maximum Rate of Increase of Oil Content or Dew Point Pressure (Optional)	570
Initial Gas-Oil Ratio (Conditional).....	572
Gas and Solvent Mixing Parameter (Conditional)	573
Minimum Solvent Saturation (Conditional).....	574
Polymer Adsorption Table (Conditional)	576
Polymer Viscosity Mixing (Conditional)	578
Reference Polymer Viscosity (Conditional).....	581
Reference Polymer Concentration (Conditional)	582

Polymer Permeability Table (Conditional)	583
Reference Seawater Viscosity (Conditional)	584
Gas Adsorption for Water Filled Blocks (Optional).....	585
Rock-Fluid Data	587
Notes on Rock-Fluid Data	587
Start of Rock-Fluid Property Input (Required)	590
Relative Permeability Curves by Table (Required)	591
Water-Oil Relative Permeability Table (Conditional)	594
Water-Oil Imbibition Relative Permeability Table.....	605
Liquid-Gas Relative Permeability Table (Conditional)	607
Capillary Pressure Hysteresis Parameters (Optional)	618
Oil-Water Relative Permeability Hysteresis Parameters (Optional).....	621
Gas Relative Permeability Hysteresis Parameter (Optional)	624
Reduced Vertical Extent of P_{cow} Cycle for Trapped Oil Hysteresis (Conditional)	626
Oil-Trapping Relationship (Optional).....	627
Method for Evaluating 3-Phase Kro (Optional).....	629
Rock Type (Optional)	632
Saturation Endpoints for each Grid Block (Optional)	633
Endpoint Relative Permeability Value for Each Grid Block (Optional).....	641
Endpoint Capillary Pressure Values for Each Grid Block (Optional)	644
Three Point Saturation Scaling Option (Optional).....	647
Two-Point Capillary Pressure Scaling (Optional)	650
Rock-Fluid Table-based Three-point Relative Permeability Scaling (Optional).....	652
Extrapolation of K_{rw} outside of the *SWT table to 1.0 (Optional).....	653
Water Oil-Capillary Pressure (J Function) Shift for Each Grid Block (Optional)	654
Non-Darcy Flow in Reservoir (Optional)	655
Non-Darcy Coefficient Correction Factor (Conditional)	658
Fracture Velocity Width Correction for Polymer–Water Mixture Viscosity Conditional)	659
Pressure Gradient Threshold Arrays.....	660
Leverett J Function Option	662
J Function Surface Tension Arrays.....	665
Scale Deposition Table (Optional).....	666
Scale Damage Table (Optional).....	667
Subdomain Reinfiltration (Optional)	668
Adsorption Parameters for each Gridblock (Optional)	670
Phase Segregation Model Modifier (Optional).....	672
Initial Conditions	675
Notes on Initial Conditions	675
Initial Conditions Identifier (Required)	678
User Input or Vertical Equilibrium Selection (Required).....	679

Initial Oil Phase Reservoir Pressure (Conditional).....	684
Initial Pressure vs. Depth (Conditional)	685
Number of Initialization Regions (Optional)	687
Specification of Initialization Regions (Optional)	688
Initial Bubble Point Pressure (Conditional)	689
Initial Bubble Point Pressure vs. Depth (Conditional)	690
Initial Solution Gas Ratio vs. Depth (Conditional)	692
Initial Dew Point Pressure (Conditional)	694
Initial Dew Point Pressure vs. Depth (Conditional)	695
Initial Oil Content (R _v) vs. Depth (Conditional).....	697
Initial Solvent Bubble Point Pressure (Conditional)	699
Initial Oil Saturation (Conditional)	700
Initial Water Saturation (Conditional).....	701
Initial Polymer Concentration (Conditional).....	702
Initial Seawater Volume Fraction (Conditional)	703
Initial Light Oil Volume Fraction (Conditional)	704
Initial Light Oil API Volume Fraction vs. Depth (Conditional)	705
Reference Depth and Reference Pressure (Conditional)	707
Depth to Water-Oil Contact (Conditional)	709
Datum Depth Specification (Optional).....	712
Specification of Capillary Pressure Values at Contacts (Optional).....	715
Specification of Method for Computing Oil Saturations in Gas Zone	718
Specification of Water Saturation Above GOC or Below WOC (Optional).....	719
Specification of Minimum Initial Oil Saturation for API Tracking (Optional).....	721
Water Saturation Input Used with Gravity-Capillary Initialization (Conditional).....	722
Non Equilibrium Water and Oil Saturation Used with Gravity-Capillary Equilibrium Initialization (Conditional).....	724
Specification of Initialization Behavior when Contacts Lie Outside PBT Table Depths.....	726

Numerical Methods Control	729
Numerical Methods Control Identifier (Optional).....	729
Maximum Timesteps (Optional)	730
Maximum CPU Seconds (Optional).....	731
Maximum Timestep Size (Optional).....	732
Minimum Timestep Size (Optional).....	733
Normal Variation in Variables per Timestep (Optional).....	734
Relaxation Options (Optional)	736
Adaptive Implicit Switching (Optional).....	738
Convergence of Newton's Method (Optional).....	740
Maximum Newtonian Cycles (Optional)	743
Minimum Number of Newtonian Cycles (Optional).....	744
Formulation Modification for Blocks with no Hydrocarbon.....	745

Force Maximum Change Checks (Optional)	747
Number of Cuts (Optional)	748
Convergence Tolerance for Linear Solver (Optional)	749
Convergence Tolerance for Linear Solver (Optional)	750
Orthogonalization (Optional)	751
Solver Equation Ordering (Optional)	752
Solver Factorization Degree (Optional)	753
Pivot Stabilization (Optional)	754
Maximum Iterations (Optional)	755
AIMSOL/PARASOL Switch (Optional)	756
Red-Black Ordering Check for Parasol (Optional)	757
Factorization Degree within PARASOL Classes (Optional)	758
Factorization Degree between PARASOL Classes (Optional)	759
Increase PDEGAA and PDEGAB for Blocks on Class Boundaries (Optional)	760
Parasol Class Partitioning Pattern (Optional)	761
Target number of Planes per Jacobian Domain (Optional)	766
Number of Threads to be Used (Optional)	767
Number of Threads for Parallel Grid-Processing (Optional)	768
Complete Storage Grid Array of Jacobian Domain Numbers (Optional)	769
Target Number of Jacobian Domains (Optional)	770
Target Number of Jacobian Domains based on Parasol Classes (Optional)	772
Specify Behavior When Solver Stalls (Optional)	774
Sub-matrix Inversion Algorithms (Optional)	775
Well and Recurrent Data	777
Notes on Well and Recurrent Data	777
Well Management and Group Control	779
Specifying the Well and Group Control Hierarchy	781
Introducing Group Control after Start of Simulation	783
Limitations of the Well Management and Group Control Module	784
Recurrent Data from Other Sections	785
Well and Recurrent Data Identifier (Required)	786
Well Change Date (Conditional)	787
Well Change Time (Conditional)	788
Well Change First Time-Step Size (Optional)	789
Maximum and Minimum Time-Step Size (Optional)	790
Setting Grid Blocks to Implicit or Explicit (Optional)	791
Setting Well Blocks and Neighbours to Implicit (Optional)	792
Set Frequency of Initialization of Bottom-Hole Pressure (Optional)	794
PDRAIN Method (Optional)	796
Group Identification (Optional)	799
Well Identification (Required)	803
Well Backflow Model (Optional)	806

Well Head Method (Optional).....	808
Well Head Iteration (Optional).....	810
Simultaneous BHP and Well Head Iteration (Optional)	812
Perforations in Inactive Blocks (Optional).....	814
Well Type Definition (Required)	816
Shut in Wells above Formation (Optional)	818
Well Status Definition (Optional)	820
Shut and Reopen a List of Wells (Optional)	822
Tubing Data for Injectors (Conditional).....	824
Tubing Data for Producers (Conditional).....	827
Wellbore Temperature (Optional)	832
Composition of Injected Water, Gas, and Oil Phases (Conditional).....	833
Well Operating Constraints (Required).....	835
Maximum Number of Continue-Repeat (Optional)	845
Monitored Well Constraints (Optional).....	846
Alter Primary Well Operating Constraint Value (Optional)	853
Reset Injection Composition (Optional).....	855
Well Current Constraint (Optional).....	857
Alter Well Constraint Value (Optional)	858
Operating Constraints in History Matching Mode (Optional).....	860
Alter Observed Flow Rates for History-Matching (Optional).....	862
Sets/Alters a Well's Phase Productivity (Optional)	864
User-Specified Reference Depth for Well BHP (Optional)	868
User-Specified Pressure Gradient For Reference Depth for Well BHP (Optional)	870
Pressure Gradients for Calculation of Pressure Differences Between Completions (Conditional).....	872
Alter Polymer Concentration (Optional)	875
Alter Seawater Injection Volume Fraction (Optional)	877
Well Scale Table Assignment (Optional).....	879
Layer Scale Table Assignment (Optional)	880
Well Scale Removal (Optional)	882
Layer Scale Removal (Optional).....	883
Data for Workover Action for Wells (Optional)	884
Resetting Well Operating Constraint after Value Change (Optional).....	887
Gas Lift Option (Optional)	889
Gas Lift Control (Optional)	892
Gas Lift Optimization (Optional)	894
Well Element Geometry (Optional)	897
Location of Well Completions (Conditional).....	900
Auto Thresholding for Well indices (Optional)	909
Well Geometry Parameters (Optional)	911
Geometric Data for Deviated Well Completions (Conditional).....	914
Simplified Geometric Data for Deviated Well Completions (Conditional)	918

Data for Hydraulic Fracture Proxy (HFP) Model (Conditional).....	920
Relative Permeability Data for Well Completions (Optional).....	925
Group Production Constraints (Optional).....	930
Group Injection Constraints (Optional)	936
Gas Make-up Target for Group Gas Recycling (Optional).....	945
Maximum Make-up Gas Rate for Group Gas Recycling (Optional)	947
Gas Producing Group for Group Recycling/Voidage Replacement (Optional)	949
Injection Group for Group Recycling/Voidage Replacement (Optional)	951
Maximum Re-injection Rate for Group Gas Recycling (Optional).....	953
Fuel Consumption Fraction for Group Gas Recycling (Optional).....	955
Maximum Fuel Consumption Rate for Group Gas Recycling (Optional)	957
Sales Rate Fraction for Group Gas Recycling (Optional).....	959
Maximum Sales Rate for Group Gas Recycling (Optional)	961
Water Make-up Target for Group Water Recycling (Optional).....	963
Maximum Water Make-up Rate for Group Water Recycling (Optional)	965
Water Producing Group for Group Recycling (Optional)	967
Maximum Re-injection Rate for Group Water Recycling (Optional)	969
Solvent Make-up Target for Group Solvent Recycling (Optional).....	971
Maximum Make-up Solvent Rate for Group Solvent Recycling (Optional)	973
Solvent Producing Group for Group Recycling (Optional)	975
Maximum Re-injection Rate for Group Solvent Recycling (Optional)	977
Monitored Group Constraints (Optional).....	979
Allow a Cycling Group to be Defined (Optional).....	982
Allow Cycling Group Data to be Revised (Optional).....	992
Allow Cycling Group Data to be Revised (Optional).....	994
Select Cycling Group Cycle Part to Start and End Cycling Group Control (Optional).....	996
Defining Group Production or Injection as Going Through a Manifold (Optional).....	998
Pressure Constraint Translation for Manifolds (Optional)	1000
Specification of Hydraulics Tables for Calculating Pressure Difference Between Manifold and Surface (Optional)	1002
Manifold Depth for Calculating Pressure Difference Between Manifold and Surface (Optional).....	1004
Group Artificial Lift Quantity Value (Optional).....	1006
Well Artificial Lift Quantity Value (Optional)	1008
Priority List for Automatic Drilling of Wells (Optional).....	1009
Group Apportionment Options (Optional).....	1011
Apportionment Method for Meeting Group Targets (Optional).....	1013
Priority Formulae for Apportionment (Conditional)	1017
Guide Rates for Groups or Wells.....	1021
Flag for Specifying Groups or Wells under or not under Group Apportionment (Optional)	1024

Allows for a Set of Keywords to be Processed When a Specified Condition (Trigger) is Satisfied (Optional)	1026
Alter Well Constraint Value via a Multiplier or an Increment (Optional)	1045
Group Production Constraint Multiplier or Increment (Optional)	1048
Group Injection Constraint Multiplier or Increment (Optional).....	1050
Define Reporting Group (Optional)	1052
Well/Group On-time Fraction (Optional).....	1054
Hydraulic Pressure Table (Conditional).....	1058
Set Number of Points for WHP Root Search (Optional).....	1066
Control Layer Lumping Option (Optional)	1068
Report Layer Lumping Option (Optional)	1071
Control Layer Lumping Option (Optional)	1073
Control Layer Lumping Option (Optional)	1074
Control Layer Lumping Option (Optional)	1075
Selection of Instantaneous Versus Actual Rates (Optional).....	1076
Dynamic Grid Amalgamation Control (Optional)	1077
Terminate Simulation (Required).....	1080

Appendix A 1081

Theoretical Outline.....	1081
Four Component Black-Oil Model with Dual Porosity	1081
Dual Permeability.....	1085
Subdomain Model	1086
Matrix-Fracture Transfer Coefficient, sigma	1086
Independent and Dependent Variables.....	1087

Appendix B 1091

Well Model.....	1091
Total Mobility Weighted Injectors	1091
Unweighted Injectors	1091
Well Injectivity.....	1091
Producers.....	1092
Wellhead Pressure Wellbore Model.....	1092

Appendix C 1095

Calculation of Well Parameters.....	1095
-------------------------------------	------

Appendix D 1099

Figure D-1a: Numbering of the Grid (Non-Corner Point)	1099
Figure D-1b: Numbering of the Grid (Corner Point)	1100
Figure D-2: Cylindrical Reservoir.....	1101
Figure D-3: Calculation of Gravity Components for a Tilted Grid System	1102

Figure D-4: Radial (Cylindrical) Coordinates	1103
Figure D-5a: Depth Measurement	1104
Figure D-5b: 3-Dimensional Variable Thickness Grid System (8x3x2)	1105
Figure D-6: Fault Throw Definition	1106
Figure D-7: Typical Data Curves.....	1107
Figure D-8: Typical Relative Permeability Curves.....	1108
Appendix E	1109
Polymer Option.....	1109
Factors and Mechanisms Affecting Polymer Floods.....	1109
Polymer Types	1115
Mass Conservation Equations.....	1116
Guidelines for Polymer Application.....	1117
References for Polymer Option	1118
Pseudo-Miscible Option	1119
Mass Conservation Equations.....	1119
Pseudo-Miscible Considerations.....	1120
References for Pseudo-Miscible Option	1125
Appendix F	1127
The Use of IMEX in Well Testing Applications	1127
Appendix G	1135
Analytical Aquifers.....	1135
Mathematical Background.....	1136
Van-Everdingen and Hurst Method	1136
Infinite Aquifer with Constant Rate Solution	1136
Limited Aquifers with Constant Rate Solution.....	1137
Carter-Tracy Method	1137
Fetkovich Method	1138
Practical Aspects of Using Analytical Aquifers	1139
a) Selecting Type of Aquifer	1139
b) Selecting Method of Solution.....	1139
Aquifer Parameters	1140
Model Defaults	1141
Appendix H	1149
Running IMEX Coupled to the GAP Surface Network Model	1149
Remote Simulation Job Submission via SSH	1160
IMEX Results Accessible from RESOLVE Script	1164
Operation on Well Layers via RESOLVE Script.....	1170
Pseudo-Solvent Production / Injection.....	1172

Appendix I **1173**

Coupling IMEX to FORGAS	1173
General Considerations and Guidelines	1173
Preparation of the IMEX Data Set	1174
Recurrent Data.....	1174
Multiple Reservoirs	1177
Remote Simulation Job Submission via SSH.....	1177
Coupling with FORGAS	1180
Getting Started Linking IMEX and FORGAS	1182

Keyword Index **1191**

Introduction

Important Changes between IMEX 2015.10 and IMEX 2014.10

SIMULATOR CHANGES

Solvent Run with GAP

Allow IMEX/GAP to operate in solvent models. With the command-line argument –ssolvent, the solvent rate is added to the hydrocarbon gas rate for producers. The total gas gravity is thus the volume-weighted gravity of the gas mixture at the bottom-hole pressure. For injectors, the gas rate is either the rate of HC gas or solvent depending upon the injector types.

Table-based 3pt Endpoint Relative Permeability Scaling

A new keyword *TBLBAS_3PTKR is introduced to switch between the block-based and table-based three-point endpoint relative permeability scaling. This option is required to match the performances among CMG's simulators.

Two-point Capillary Pressure Scaling

Use keyword *PCSCALE_2PT to activate the two end point capillary pressure scaling for both oil-water and oil-gas capillary pressure for reservoir blocks. The default capillary pressure scaling is the maximum factor scaling. The two point scaling requires the input of the minimum factor, through keywords *PCWMIN, *PCWIMIN, *PCGMIN, *PCGIMIN, *JFWMIN, *JFGMIN.

Oil-Trapping Relationship

A new keyword *OILTRAP is implemented to assign the oil-trapping relationship that is used to determine the residual oil saturation of imbibition from the historical-maximum-attained oil saturation. Different methods (the Land's correlation, linear relationship and user-supplied table) will affect the oil recovery from the reservoir transition zone.

Mixed-Wet Hysteresis

New keywords to allow the input of water-oil imbibition relative permeability tables (*SWTI); assign hysteresis methods of oil-water and gas relative permeabilities and signal the entry of associated parameters (HYSKRO, *HYSKRW, *HYSKRG, *KILLOUGH_MOD_KRWCO). For the hysteresis effect on the oil-water relative permeability, CARLSON, KILLOUGH and KILLOUGH_MOD methods are implemented, in addition to the original Kro hysteresis which may not be necessarily coupled with the Pc hysteresis.

Solver Technology

New matrix inversion cofactor based algorithms introduced for 2×2 and also for 4×4 matrices. These cofactor techniques tend to be a bit more accurate, no pivoting is required for 2×2 , 3×3 and 4×4 matrices and are faster than traditional LU methods. Cofactor techniques were already used for 3×3 block diagonal matrices in previous versions of IMEX. The corresponding keyword is *MATINV. Sub-keyword *COFACTOR uses cofactor methods for these block diagonal matrices. The old defaults may be obtained using *MATINV *OLD. For IMEX, *MATINV *COFACTOR is now the default. IMEX fluid models affected by this change are OILWATER and GASWATER (2×2) and MISCG, MISNCG, POLY, API-INT and BLACKOIL_SEAWATER (4×4).

Template Data Set Changes

Added mxsmo069.dat to demonstrate the hydraulic fracture with planer template

Added mxsmo070.dat for mixed wet water-oil hysteresis test

Added mxsmo071.dat for Pc trapped oil hysteresis test

Added mxsmo072.dat for verification of Pc scaling

Added mxsgw021.dat to demonstrate the *PUMP2 option for iSegWell

Added mxsgw022.dat to demonstrate the *TABLE for FCD in iSegWell

Added mxsgw023.dat to demonstrate the scale removal (*SGPFSCLRMV) in iSegWell

Added mxwwm066.dat to demonstrate the general form of *PERF

Added mxwwm066.dat to demonstrate WI regulation (*WIRANGE)

Added mxwwm067.dat to demonstrate rate instability monitoring (*RATSTAB)

Added mxwwm069.dat to demonstrate hydraulic radius in head calculation (*HYRADIUS)

Added mxwwm070.dat to demonstrate WHT calculation (*TWELLBORE) from PTUBE1

Data Incompatibilities with Previous Versions of IMEX

None.

Keyword Default Changes from Previous Versions of IMEX

The target number of Jacobian domains n via command-line argument –jacdoms n was defaulted to 2 if n is absent. It now takes from the target number of threads m if command-line argument –parasol m is specified.

WELL MANAGEMENT CHANGES

Keyword *PERF

A new tabulated list-driven format to allow layer-varying input of all the parameters (wi , dir , ff , kh , re , rw , $skin$, di) involved in well index calculations. This format is very flexible and backward compatible.

Sub-keyword *RATSTAB

Newly added wellbore rate stability monitor (*MONITOR) based on the operating BHP-RATE curves as a complement to the existing *WHYSTAB monitor. A real well should

operate only above a certain rate at which the BHP remains in the stable region of the tubing-intake or tubing-outflow curve with WHP and phase ratios being held constant.

Keyword *WIRANGE

A new keyword 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.

Sub-keyword *HYRADIUS

Newly added well hydraulic radius input for gravitational-frictional head calculations (*HEAD-METHOD *GRAV-FRIC). The well hydraulic radius can be different from the default wellbore radius from *GEOMETRY.

Keyword *KRPERF

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

Enhanced Hydraulic Tables for Well-head Temperature Calculation

Keyword *PUTBE1 has been enhanced to read and process the table body of well-head temperature (*WHT). This along with new keyword *TWELLBORE allows computation of well-head temperature of producers using tabulated wellbore data.

GRID CHANGES

Hydraulic Fracture with Planar Template

With new grid property array qualifiers (*FZ and *NFZ) to assign data to fracture or non-fracture zone, 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

Important Changes between IMEX 2014.10 and IMEX 2013.10

SIMULATOR CHANGES

Multi-Segmented Well Module

IMEX now incorporates a multi-segmented well module. Outboard documentation is provided.

Extended Oil Solvent Mixing Parameter Tables

The *OMEGA_OST table allows the input of an extended Omega (oil-solvent mixing parameter) table for solvent models. This table allows the user to define two separate types of Omega, one to be used in the oil-solvent density mixing rule (omega_den) and one to be used in the oil-solvent viscosity mixing rule (omega_vis). Sorm may also be altered in this table using the sormadd column. Finally, it is possible to use the *OMEGA_OST table to define tables that are functions of both pressure, P, and bubble point pressure, Pb.

'Three Point' Critical Saturation Relative Permeability Value Scaling Arrays Added

It is now possible to scale the relative permeability table values of Krwro, Krocrw, Krgrl and Krogcrg at the critical saturations of oil, water, liquid and gas respectively in addition to the values of Krwi, Krocw, Krgcl and Krogcg, which are endpoint values at irreducible /connate saturations. See keyword *KRWRO and the tutorial chapter 'Two Point and Three Point Scaling of Relative Permeability Values' for more information.

Solver Technology

Improved Parallel Performance

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

Other New Keywords

*Loss of Miscibility Smoothing at Sorm (*MINSS *SOTHRESH soval)*

'*MINSS *SOTHRESH soval' allows the user to control the range over which loss of miscibility occurs at Sorm, soval is entered and loss of miscibility occurs over the range So = SORM+soval and So = SORM-soval.

*Phase Segregation Model Modifier for *TRANSFER 1 Option*

The use of *PHS_SEG_MODL *MODIFIED allows the user to enter DKFRAC > DK without having DKFRAC internally limited to DK for the capillary continuity calculation when using the *TRANSFER 1 phase segregation model.

Explicit Specification of Initialization Behavior when Contacts Lie Outside PBT Table Depths

The CONTACT_PBT_BEHAVIOR keyword can be used to change how the simulator reacts when initialization region phase contacts lie outside the range of the initialization region's PBT table. *The LEGACY option, the previous default, would issue a warning and then change the contacts. The default is now changed to *ERROR which prevents the run from proceeding until the table or contacts are altered. The user can choose to ignore the problem by specifying the *WARNING option which warns but does not reset the contacts.

Template Data Set Changes

Added mxsmo062.dat to demonstrate *PHYS_SEG_MODL
Added mxsmo063.dat to demonstrate *CONTACT_PBT_BEHAVIOR
Added mxsmo064.dat to demonstrate *AQUIFER *RESBND
Added mxsmo065.dat to demonstrate *AQUIFER *RESBND
Added mxsmo066.dat to demonstrate 3Pt Scaling of Kr Value
Added mxsmo067.dat to demonstrate 3Pt Scaling of Kr Value
Added mxsmo068.dat to demonstrate use of BLOCKGROUPS
Added mxspr010.dat to demonstrate *OMEGA_OST
Added mxspr011.dat to demonstrate *OMEGA_OST
Added mxwwm062.dat to demonstrate *WELGEO and *INCOMPWL
Added mxwwm063.dat to demonstrate *WCURRCN
Added mxwwm064.dat to demonstrate *DTWAUTO
Added mxwwm065.dat to demonstrate new trigger feature (polymer injection)

Data Incompatibilities with Previous Versions of IMEX

None.

Keyword Default Changes from Previous Versions of IMEX

CONTACT_PBT_BEHAVIOR – Default of *LEGACY changed to *ERROR

WELL MANAGEMENT CHANGES

***Triggers on Solvent/Polymer Rates and Solvent/Polymer Cumulatives**

The *ON_WELL and *ON_GROUP based triggers can now be used to monitor solvent volumetric injection/production rates and cumulative volumes as well as polymer mass rates and cumulative mass produced/injected.

Keyword *WELGEO

*WELGEO specifies the geometric characteristics for listed wells to be used by the simulator to calculate the well index internally.

Keyword *INCOMPWL

*INCOMPWL sets the global compositions of injected lift-gas for producers or injected streams (gas or solvent) for injectors used in the wellbore calculations.

Keyword *WCURRCN

*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 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.

Keyword *DTWAUTO

Allows the user to use DTWELL after a well management induced well change.

Subkeyword *ALL will force the simulator to use the timestep size specified on *DTWELL if a well action is deemed significant. This could include well auto-drilling to meet group targets, auto-well/auto-layer constraints, or changes in operating constraints. Sub-keyword *AUTODRL will force the simulator to use the timestep size specified on *DTWELL strictly due to the auto-drilling of a well.

Keywords Retired

*PERFVHY is no longer supported, the *PERF keyword can be used instead.

Keyword Default Changes

*APPOR-METHOD – new default is *INGUIDE, changed from *IP.

All runs using group control will be affected. To reproduce earlier results, *APPOR_METHOD *IP must be specified for each group.

Keyword Name Changes

“*GCONI *TARGET” is changed to “*GCONI *GTARGET” and “*GCOMP *TARGET” is changed to “*GCOMP *GTARGET” to avoid possible reading issues when a well *TARGET is also defined.

GRID CHANGES

***AQUIFER *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.

BLOCKGROUPS

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 System”. In addition, a block group can be applied to certain block-based data-input keywords (e.g., *REFINE).

Important Changes between IMEX 2013.10 and IMEX 2012.10

SIMULATOR CHANGES

Initialization Regions Separate from PVT Regions

Initialization regions may be specified using the *NREGIONS and *ITYPE keywords. Previous versions of IMEX linked PVT regions to initialization regions. Thus one PVT region was required for each initialization region. This is no longer required. Older data sets are unaffected.

Rock Compaction Table Reassignment in Recurrent Data

The *CTYPE value (compaction table number) of gridblocks can now be reassigned in recurrent data. This allows the user to realistically model processes where permeability modification vs. pressure behavior is added later in a run. Hydraulically fracturing and introducing fracture conductivity vs. pressure behavior in a new well partway through a simulation is an example of this. Please see the manual as some restrictions apply to the use of this feature in recurrent data.

Solver Technology

Improved Parallel Performance

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

Separate Graphics/Restart SR2 files (*RESTART_SR2 or Command Line -restart_sr2)

We have changed the default behavior when writing out SR2 graphics and restart data and have added a keyword and a command-line option to control if graphics and restart information in the sr2 files are combined or separated, for both input and output.

The new default ‘*RESTART_SR2 *SEPARATE’ (-restart_sr2 separate) separates graphics and restart data. ‘*RESTART_SR2 *MAIN’ (-restart_sr2 main) combines graphics and restart data.

Combined or separated SR2 files must be kept consistent between initial and restart runs. The new IMEX default is to use the *SEPARATE option as this option requires less disk space and take less CPU time for output due to writing the graphics data in single precision while writing restart data in double precision. The *MAIN option needs to write both graphics and restart data in double precision to maintain restarting accuracy. Please see the *RESTART_SR2 keyword in the IO Control Section for details on file naming conventions.

Other New Keywords

*Pressure vs. Depth Table (*PREST)*

During non-equilibrium initialization, it is possible to assign block pressures using a table of reservoir pressures vs. depth for each initialization region. Previous versions of IMEX required the user to enter the entire pressure array. This option simplifies non-equilibrium initialization.

*Output of Well PI Based on an Approximation of the Well’s Drainage Pressure (*OUTSRF *WELL *PDRAIN, *PDRAIN-METHOD)*

Specification of *OUTSRF *WELL *PDRAIN initiates a more involved calculation of a well’s productivity index based on a flexible approximation of well drainage area/pressure. *PDRAIN-METHOD controls how the approximate drainage pressure is calculated.

*Change the Minimum Oil Saturation Initially Placed in Each Block when Using API-Tracking Option (*APIMINSO)*

Using the API-Tracking Option, it was possible for the model not to initialize blocks with a small amount of oil (to prevent formulation issues) when using the *DEPTH_AVE option. This issue has been eliminated in this release. In addition the initial minimum oil is now controllable using the *APIMINSO keyword in the Initialization Section. *APIMINSO is set to 1.00e-04 by default.

*New Restarting Features (*RESTART *RESDATE/*RESTART *RESTIME)*

It is now possible to specify a restart at a specific date or time in addition to a timestep number. The *RESDATE “date” and *RESTIME “time” keyword can be used in addition to *RESTART “num”

Template Data Set Changes

Added mxflu006.dat to demonstrate *PREST table input

Added mxsмо060.dat to demonstrate initialization regions

Added mxsмо061.dat to demonstrate *CTYPE in recurrent data

Added mxwwm055.dat to demonstrate *DRILLT

Added mxwwm056.dat to demonstrate gas lift triggers

Added mxwwm057.dat to demonstrate new Sector triggers

Added mxwwm058.dat to demonstrate *AVRGTIME in triggers

Added mxwwm059.dat to demonstrate *OPERATE-HIST

Added mxwwm060.dat to demonstrate *WCUTBACK in *OPERATE

Added mxwwm061.dat to demonstrate *PDRAIN well PI output option

Data Incompatibilities with Previous Versions of IMEX

None.

WELL MANAGEMENT CHANGES

***OPERATE-HIST, *ALTER-HIST**

The *OPERATE-HIST keyword signals that a well’s operating constraints entered using the *OPERATE keyword are historically observed rates and allows the well to operate under history matching mode. Under history matching mode, it is possible to specify the reservoir fluid rate using the sub-keyword *RSV. This converts the STC fluid rates entered on the *OPERATE keywords to reservoir fluid volumes. The reservoir volume (based on current reservoir conditions) is produced. *ALTER-HIST allows the modification of the historically observed flow rates.

***OPERATE *WCUTBACK**

A new type of well operating constraint has been introduced. The *WCUTBACK sub-keyword of *OPERATE specifies that a well will run on a reduced rate target of a specified control phase whenever it violates a well ratio constraint (example GOR, WOR etc.) or maximum/minimum well block pressure constraint.

***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.

***AVRGTIME in Triggers**

The *AVRGTIME sub-keyword of *TRIGGER can be used to specify the time period over which a moving average of a quantity (such as rate) is calculated.

Gas lift Triggers

The *ON_WELL and *ON_GROUP triggers now include gas lift based triggers.

***Sector Trigger Enhancements**

The *ON_SECTOR keyword allows test conditions to be applied to a sector. Minimum and maximum sector pressure and sector saturations have been added to *ON_SECTOR test quantities

***PDRAIN-METHOD**

The *PDRAIN-METHOD keyword allows the user to choose how the approximate drainage area is to be calculated when *OUTSRF *WELL *PDRAIN output is selected.

GRID CHANGES

***TRANSMULT**

The *TRANSMULT keyword allows the specification of the action to take when encountering successive transmissibility multipliers. *REPLACE (the default) allows the current multiplier to be replaced by the new multiplier. *ACCUMULATE allows the current multiplier to be multiplied by the value just read in. *TRANSMULT may be altered in recurrent data to change the default *REPLACE option to *ACCUMULATE.

Inter Region Transmissibility Multipliers

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

***CRNCON_NODE**

A NODE based corner-point grid option has been implemented in CMG simulators. Builder can be used to convert existing corner point grids to NODE based grids. Alternatively the keyword *CRNCON_NODE or the command line option '-crncon_node' allows the user to have the simulator internally convert an existing corner point grid to its NODE based counterpart. This option may speed up the grid generation part of a simulation considerably at the cost of extra memory. NODE based grids have been found to be very efficient for grids with a large number of refined cells (for example: hydraulically fractured wells modeled using the LS-LR-DK technique).

Important Changes between IMEX 2012.10 and IMEX 2011.10

SIMULATOR CHANGES

Gas Adsorption

The new keywords *ADGMAXV, *ADGCSTV, *ROCKDEN and *ADGPCRIT (Rock-Fluid Section) and *ADS_WATERZONE (Component Properties section) allow the user to specify the Langmuir Isotherm parameters used to model the adsorption/desorption of gas to/from rock. This feature can be used to model simple adsorption effects in shale gas and CBM problems.

Multiple Solvent PVT Tables

IMEX now optionally allows the PVTS table, the *OMEGASG keyword and the “*DENSITY *SOLVENT” keywords to be entered for each PVT region. This allows the solvent properties and mixing parameters between solvent and oil and solvent and gas to vary from PVT region to PVT region. This option is backwards compatible; no data modification is required for earlier solvent models to run.

Solver Technology

Improved Cache Alignment

For both fully implicit and adaptive implicit models this will improve parallel efficiency.

Automatic Parallel Partitioning in 2 Dimensions (Any Two of the x, y, and z Directions)

Two dimensional partitioning now can be set up automatically using *PPATTERN *AUTOP2D

Improved Parallel Performance

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

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.

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.

Other New Keywords

*Three Point Endpoint Scanning (*3PTSCALING)*

When *3PTSCALING is enabled an interior scaling point is added to the saturation endpoint 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}	Scaled between S_{wcrit} and $(1-S_{orw})$, if block S_w is less than or equal to $(1-S_{orw})$, or between $(1-S_{orw})$ and $(1-S_{oirw})$, if S_w is greater than $(1-S_{orw})$.
K_{row}	Scaled between S_{wcon} and S_{wcrit} , if block S_w is less than or equal to S_{wcrit} , or between S_{wcrit} and $(1-S_{orw})$, if S_w is greater than S_{wcrit} .
K_{rg}	Scaled between S_{lcon} and S_{lrg} , if block S_l is less than or equal to S_{lrg} , or between S_{lrg} and $(1-S_{gcrit})$, if S_l is greater than S_{lrg} .
K_{rog}	Scaled between S_{lrg} and $(1-S_{gcrit})$, if block S_l is less than or equal to $(1-S_{gcrit})$ or between $(1-S_{gcrit})$ and $(1-S_{gcon})$, if S_l is greater than $(1-S_{gcrit})$

*Extrapolate K_{rw} from $K_{rw}(S_{orw})$ to $K_{rw}=1.0$ at $S_w = 1.0$ (*KRWEXTRAP)*

This keyword enables the linear extrapolation of water relative permeability from the last water saturation in the oil-water table to a saturation of 1.0 and a value of K_{rw} of 1.0.

Enhancements to Existing Keywords

**WATER_FIX “Option 4”*

The WATER_FIX keyword has been used in the past to improve performance in water filled blocks when small amounts of oil or gas re-enter the block. Without this option, numerical issues could occur.

The *WATER_FIX 4 option has been implemented to improve its performance. *WATER_FIX 4 is recommended for models where the complete disappearance of water leads to numerical issues. Water_fix 4 includes all of the enhancements included in water_fix 2 and 3.

Template Data Set Changes

Added mxgro024.dat to demonstrate the *IRCONNECT keyword

Added mxspr009.dat to demonstrate multiple solvent PVT regions

Added mxsmo054.dat to demonstrate *PPARTITION *AUTOP2D (two dimensional partitioning)

Added mxsmo055.dat to demonstrate a shale gas model using desorption

Added mxsmo056.dat to demonstrate a coal bed methane model using gas desorption

Added mxsmo057.dat to demonstrate three point scaling (*3PTSCALING)

Added mxsmo058.dat to demonstrate K_{rw} extrapolation (*KRWEXTRAP)

Added mxwwm054.dat to demonstrate on-time average layerlump rate output (*LAYERLUMP_ACTR)

Added mxwwm055.dat to demonstrate the specification of the minimum time interval between the opening of auto-drill wells in a dill queue (*DRILLT) *DRILLT not in 2012 manual, outboard documentation only

Data Incompatibilities with Previous Versions of IMEX

Keywords *SECTOR, *SECTORARRAY and *SECTORNAMES may not be used together.

The *LAYERXYZ calculation method has been improved, producing different results, in some cases, from the previous calculation method.

Use of variable *PB and *PDW and *API initialization arrays may result in slightly different initialization. A problem, which occurred when inconsistent data was entered, has been corrected. This does not affect the *PBT, *PDWT or *APIT table vs. depth initialization options.

WELL MANAGEMENT CHANGES

***LAYERXYZ-METHOD *DIRECT**

The original method for calculating well bore radius and well permeability for directional wells has been improved. The New default method, directional weighting (*DIRECT) replaces the previous method (*OLD) which used cardinal weighting. Please see the keyword *LAYERXYZ-METHOD for a detailed description of the differences.

***MONITOR *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).

***MONITOR *AUTOLAYER1**

Similar to *AUTOLAYER but corresponding to *SHUTLAYER1.

***GCONP *TARGET *NONE**

Remove any production target specified previously (including *VREP, *RECYCLE and *PMAINT). Existing maximum / minimum constraints are not affected.

***GCONI “stream” *TARGET *NONE**

Remove any previously specified injection target (including *VREF, *VREFP, *RECYCLE and *PMAINT) for the indicated stream (gas, water or solvent). Existing maximum constraints are not affected.

***LAYERLUMP_ACTR**

Use on-time averaged rates for calculation of rates and rate based quantities when using LAYERLUMPS. The default is to use instantaneous rates.

***DRILLT**

*DRILLT is a keyword to allow the specification of minimum time intervals between the auto-drilling of injectors and producers (keyword not in 2012 manual, outboard documentation only).

GRID CHANGES

***IRCONNECT Keyword**

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.

***SECTOR, *SECTORARRAY and *SECTORNAMES**

Keywords *SECTOR, *SECTORARRAY and *SECTORNAMES may not be used together. In a single data set all sectors need to be defined using the same method. BUILDER can be used in the conversion of one format to another.

**MOD *ARRAY Input option Added*

The *MOD *ARRAY input option has been implemented for Grid Module variables (see Simulator Changes).

Corner Point Node Location Entry

Specify the location and usage of each corner point node of a fundamental grid. The keywords *NNODES, *NODEX, *NODEY, *NODEZ, *NODES, and *BLOCKS are used. This option reduces the size of the grid data and speeds up grid reading.

LGR Grid Definition Enhancement (Ratio Method)

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.

Important Changes between IMEX 2011.10 and IMEX 2010.10

SIMULATOR CHANGES

Static Grid Amalgamation

IMEX can now use the DYNAGRID Option to statically amalgamate, re-refine and de-refine grid blocks. The keywords *AMALGAMATE, *INTO, *DEREFINE, *REREFINE and *DEAMALGAMATE are supported. Please see the *DYNAGRID Keyword in the Recurrent Data section for details

Solver Technology

Improved Manual Parasol Partitioning

The manual Parasol partitioning feature can be used to optimize Parasol partitioning when required, this feature, along with the Command Line Option “–info_partition” allows the user to optimize parallel solver performance.

Improved Parallel Performance

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

Improved Crossflow Handling for API Tracking, Solvent and Polymer Models

The fully mixed crossflow model previously had difficulties properly distributing fluids in the wellbore for API tracking, solvent, and polymer/seawater models. This issue has been addressed in the 2011.10 version of IMEX.

Improved API Tracking Formulation

Prior to this release of IMEX, the API tracking formulation in IMEX could fail to converge adequately enough to ensure a consistent solution. This has been remedied in the 2011.10 release.

Improvements to Condensate Model STC Density

Under some circumstances it was possible that the condensate STC density was not calculated properly, this has been corrected in the 2011.10 release. This was noted as an issue when the well module *VREFP voidage replacement option was introduced.

Other New Keywords

*Undersaturated Bo and Oil Viscosity with API Tracking (*BOTAPI, *VOTAPI)*

Prior to this release, the use of the *BOT and *VOT undersaturated tables was not supported when the API tracking option was used. Instead simpler options were required (see *APIGRAD and *CVO). This release introduces the *VOTAPI and *BOTAPI tables for API tracking models. The two tables are entirely analogous to the *BOT and *VOT tables for non API tracking models. Please see keywords *BOTAPI, *VOTAPI and *APIGRAD for more details.

*Limit Maximum Rate of Increase of Oil Content (*DRVDT)*

Analogous to *DRSDT for solution gas ratio, *DRVDT controls the rate of increase in condensate oil content with time. Please see the *DRVDT and *MINCYC keywords for more details.

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

On the face of it, IMEX and STARS endpoint scaling seem very similar; unfortunately the devil is in the details.

Often it was necessary to use additional endpoint array keywords to ensure STARS and IMEX would use the same scaling. The differences were due to the fact that IMEX considered S_{LCON} in the gas-liquid table to be the independent variable (changing S_{OIRG}) when S_{WCON} was altered on the oil-water relative permeability table and STARS considered that S_{OIRG} was the independent variable (changing S_{LCON}).

In addition, STARS has the additional feature (which IMEX 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 IMEX uses the *SLCON array) will turn on *SCALING-STARS and enable IMEX to treat these two aspects of endpoint scaling as STARS does.

*Rs and Rv vs. Depth Tables Added (*RST and *RVT)*

Previous versions of IMEX allowed bubble point and dew point pressure versus depth tables to be entered in the initialization section. The new version of IMEX also allows the entry of Rs vs. Depth and Rv vs. Depth tables. These tables are converted into either Pb or Pdew vs. depth tables immediately after input. Please see keywords *RST and *RVT for a complete description. This option is incompatible with API tracking models.

**FORCE_MAXCHK Keyword*

Normally the maximum change test (entered using the *MAXCHANGE keyword) is not applied on the first three Newton iterations of a timestep, and convergence is allowed even if the maximum change test is violated.

When *FORCE_MAXCHK is active (*ON), convergence will not be allowed in the first three Newton iterations if the maximum change test is violated. However the test itself is not applied to repeat the timestep. If it is violated, the simulator will continue iterating until either the maximum change test is not violated within the first three Newton iterations or until the 4th Newton iteration is reached and the maximum change test is applied.

The default for *FORCE_MAXCHK is *OFF for all models except the API tracking models, where the default is *ON. Please see the *FORCE_MAXCHK keyword for a complete description.

Enhancements to Existing Keywords

**WATER_FIX Formulation Option Extended to Non Black Oil Models*

The WATER_FIX keyword has been used in the past to improve performance in water filled blocks when small amounts of oil or gas re-enter the block. Without this option, numerical issues could occur.

The use of this option has been extended to the non Black Oil models, including API tracking, condensate and gas-water with condensate models. In addition, a new version of the keyword, *WATER_FIX 3, has been implemented to improve its performance. *WATER_FIX 3 is recommended for API tracking models. Please see the *WATER_FIX keyword for more details.

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

The useful *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).

*MINC-DK Option (*MINC and *TRANSD > 0)*

Similar to SD-DK (see *TRANSD), the MINC-DK option allows the use of the *MINC dual continua option with the *TRANSD keyword. When *MINC and *TRANSD > 0 are both input, in the vertical direction, all matrix blocks in a MINC block will be connected to the corresponding matrix MINC block directly above and below the current block. This allows the user to obtain the better areal transient response of MINC and have the same type of gravity drainage seen in dual permeability models. When this option is used, it is necessary to set all DKFRAC = 0.0 to avoid overestimating matrix-fracture and matrix-matrix flow within a MINC block. Please see keywords *TRANSD and *MINC for more details.

Template Data Set Changes

Added mxdrm008.dat to demonstrate grid amalgamation

Added mxdrm009.dat to demonstrate grid derefinement and rerefinement

Added mxfrr021.dat to demonstrate MINC-DK

Added mxsмо049.dat to demonstrate *DRVDT Keyword

Added mxsмо050.dat to demonstrate the Use of *BOTAPI and *VOTAPI tables in an API tracking Model

Added mxsмо051.dat to demonstrate the usage of *SWINIT with the *DEPTH_AVE initialization option and the *RST option (Rs vs. depth table)

Added mxsмо052.dat to demonstrate the *SCALING-STARS option

Added mxsмо053.dat to demonstrate the manual selection of Parasol partitions

Added mxwwm053.dat to demonstrate the use of the logical “OR” condition in triggers

Data Incompatibilities with Previous Versions of IMEX

*SCALING-OLD relative permeability endpoint scaling is no longer supported as it is incompatible with the *SCALING-STARS option

WELL MANAGEMENT CHANGES

***FRAC Added to *WELL Keyword**

Well fractions can now be defined per well and is useful for defining fractional ($\frac{1}{2}$, $\frac{1}{4}$, 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 $\frac{1}{4}$ 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.

GRID CHANGES

Improve Corner-Point Efficiency

These changes were made to the SR2 file format to improve efficiency and capacity for large corner-point grids. Results and IMEX can read both the new and previous SR2 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).

***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).

**3DMF Added to SUBDOMAIN Option*

Normally when calculating the matrix-fracture transmissibility term for the subdomain dual continua model, the vertical flow into the subdomain is ignored. In certain circumstances this may significantly reduce the overall matrix-fracture transmissibility, the use of the option *3DMF introduces the M-F transmissibility terms from the vertical direction into the top and bottom subdomain cells. Please see the *SUBDOMAIN keyword in the Reservoir Description Section.

**TRANSD Support Added to *MINC to Produce MINC-DK Option*

Support for simulator change.

Important Changes between IMEX 2010.10 and IMEX 2009.10

NOTES FOR IMEX 2010.10

Well Module

1. Completion Lumping Option
2. *WHYSTAB *AUTOWELL Monitor
3. *MXCNRPT Maximum number of Continue Repeats
4. *KRPERF *SORMAX Entry for Trapped Oil Hysteresis

Simulator

1. Improved Parallel Performance
2. Trapped Oil Hysteresis Option
3. IMEX-GAP Interface Improvements:
Handles Crossflow during shut in (like *MODELSHUT)
4. Flux Sector Reporting Option
5. Entry of Permeability and Non Darcy Correction Factor in Recurrent Data

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.

*CLUMPSSETTING 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).

***KPERF *SORMAX Entry for Trapped Oil Hysteresis**

SORMAX on the *KPERF keyword allow the user to change this saturation endpoint in recurrent data. Sormax is the oil saturation in the two phase oil-water system at which oil first becomes immobile during the boundary/primary imbibition process (the imbibition starting from oil saturation equal to 1-Swcon). It only has an effect when the Krow hysteresis or trapped oil hysteresis is modeled for the rock type of the completion.

Improved Parallel Performance

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

Trapped Oil Hysteresis Option

Trapped oil hysteresis is now available for P_{cow} and K_{row} curves. To invoke the trapped oil hysteresis option, it is only necessary to define the imbibition P_{cow} in an SWT table using constant P_{cow} for water saturation values equal to and great than 1-Sormax. Once activated, trapped oil hysteresis couples P_{cow} and K_{row} hysteresis. Keyword *HYSKRO is not necessary and will be ignored. The endpoint array *SORMAX has been added to allow the user to enter values of Sormax on a block by block basis. The keyword *DAMP-PCOW-TROIL has been added to reduce the capillary pressure difference between imbibition and drainage curve which might result in a pressure oscillation. Please see the Tutorial section: The Trapped Oil Hysteresis Option for P_{cow} and K_{row} (Oil-Water System).

IMEX-GAP Interface Improvements IMEX-GAP Interface Improvements:

*Handles Crossflow during Shut in (like *MODELSHUT)*

When wells are shut in by GAP but experience crossflow during the shut in, the resulting pressure profiles around the well may change drastically if the productivity of well layers vary drastically and crossflow is occurring. This feature allows IMEX to shut in a GAP /RESOLVE controlled well using a *MODELSHUT type option, which allows crossflow in the shut in well. GAP/RESOLVE controls whether the well is *MODELSHUT or simply shut in. The user does not need to specify this IMEX keyword.

Flux Sector Reporting Option

*FLUX_SECTORNAMES and *FLUX_ISECTOR keywords introduce special sector definitions which are used to calculate reservoir flow between these “flux sectors”. Flow into these sectors and between these sectors can be reported to the output print file and to the simulation results file as time series data. Flow can be further broken down into I-direction, J-direction and K-direction flow components.

*FLUX_SECTORNAMES defines a list of flux sector names and their corresponding flux sector numbers. *FLUX_ISECTOR assigns these flux sector numbers to cells using standard array input options. Both keywords can only appear once in the data set. Please also see the *WPRN *SECTOR, *WSRF *SECTOR, *OUTPRN *FLUX_SECTOR, and *OUTSRF *FLUX_SECTOR keywords for output options.

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.

Important Changes between IMEX 2009.10 and IMEX 2008.10

NOTES FOR IMEX 2009.10

Well Module

1. *ITUBE1/*PTUBE1
2. *GCONI ‘Stream’ *PMAINT *PMMAXR d1 d2
3. *GCONPMULT *RECYCLE
4. *LAYERHFP
5. *PERFRG keyword removed

Simulator

1. Improved Parallel Performance
2. *DRSDT
3. Oil wet Initialization Option
4. *RTYPE Subkeyword of Relative Permeability Scaling Arrays
5. IMEX-GAP Interface Improvements:
 - Layer Information Available for Scripting
 - Improved IPR Curve Generation
6. PTUBE1/*ITUBE1 Table Generation in Builder

***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 (m^3/day | bbl/day | cm^3/min) for water and (m^3/day | scf/day | cm^3/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.

***LAYERHFP**

*LAYERHFP allows the user to supply geometric and non Darcy information specifying productivity indices which act as proxies for vertical fractures (assuming pseudo radial flow) on a layer by layer basis.

***PERFRG keyword removed**

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

Improved Parallel Performance

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

***DRSDT**

This option allows the input of the maximum rate of increase of solution gas ratio.

Oil Wet Initialization Option

An oil wet reservoir when initialized using the depth average initialization option now produces realistic saturation profiles. This assumes that the reservoir was initially water wet and over time changed to an oil wet condition.

***RTYPE Subkeyword of Relative Permeability Scaling Arrays**

The relative permeability scaling arrays can now be made to operate on blocks which are members of specified rocktypes (RTYPES).

IMEX-GAP Interface Improvements

The IMEX-GAP interface now is able to output well-layer level information to Resolve, this enables the user to script in Resolve using well-layer information.

The IPR curve generation now optimally assigns pressure points in the IPR table to try to capture curvature more accurately.

***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.

Important Changes Between IMEX 2008.10 and IMEX 2007.10

NOTES FOR IMEX 2008.10

Well Module

1. Cycling Groups
2. *GAPPOR *AUTODRILL “stream identifier”

Simulator

1. Improved Parallel performance
2. Improvements to the IMEX-GAP interface
3. NONDARCY GENERAL option
4. Fracture Cleanup Model

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. Examples include water alternating gas injection (WAG) or cyclic steam 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.

Improved Parallel Performance

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

Improvements to the IMEX-GAP interface

LSF and SSH network job submission is now supported. The GAP down-time factor is now passed to IMEX in order to calculate the IMEX on-time factor. Petroleum Experts is now

supporting the IMEX-GAP DLL so the advanced IPR calculations in Resolve are now available in this DLL.

Non Darcy General Option

The Non Darcy model now has a more general input method to input parameters for the β factor, in addition to the existing techniques. This input also allows non Darcy flow to be modeled in all phases, not only the gas phase. In addition, input parameters are made available to improve convergence of the Forchheimer number iteration (see Tutorial Section “Modeling Non Darcy Flow in Hydraulic Fractures Accurately Using a Grid Based Approach”).

Fracture Cleanup Model

The fracture cleanup model is a group of options which allows the user to model the production and clean-up of fracturing fluids used to create/extend a fracture followed by production from the fracture.

Velocity dependent polymer-water mixture viscosity is modeled

(Keyword: PMIX VELTABLE)

A fracture width velocity correction term is added (Keyword: FRWIDTHCOR)

Block and direction dependent pressure gradient thresholds are modeled

(Keyword: PTHRESHI/PTHRESHJ/PTHRESHK)

Non equilibrium initial saturation overrides have been added (Keyword: SWNEQ/SONEQ)

The pressure gradient threshold option can be initiated at a restart

See the Tutorial Section “Using the Fracture Cleanup Model” and review the keywords mentioned above for more details.

Important Changes Between IMEX 2007.10 and IMEX 2006.10

NOTES FOR IMEX 2007.10

Well Module

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, Water or Solvent *GMKUP, *WMKUP, *SMKUP
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

Simulator

1. Improved Parallel performance
2. Improvements to the IMEX-GAP interface
3. Volatile Oil Model
4. Subdomain-Dual Permeability Model
5. Water – Polymer Mixture Viscosity Table
6. Depletion Option
7. Improved Aquifer Printout

Shut in Wells above Formation

*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.

Two New Drawdown Constraint Types

*DWA defines the maximum drawdown within all open layers,

$$\Delta P_d = \max_{l, \text{open}} \pm (P_{\text{block}_l} - P_{\text{well}_l})$$

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

*DWB specifies the average drawdown for all open layers:

$$\Delta P_d = \pm \sum_{l, \text{open}} PI_l (P_{\text{block}_l} - P_{\text{well}_l}) / \sum_{l, \text{open}} PI_l$$

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). 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 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, *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, Water or Solvent (GCONI *VREP *GMKUP/*WMKUP/*SMKUP)

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

Sector Pressure Maintenance Injection Target

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.

Improved Parallel Performance

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

Improvements to the IMEX-GAP interface

The Voidage replacement interface in RESOLVE is now supported. Scripting using IMEX variables is supported and IMEX variables are now available for output in RESOLVE. TRIGGERS may now be used in IMEX-GAP runs. Selected Simulator well control is now enabled in IMEX-GAP runs. Please See Appendix H for details.

Volatile Oil Model

The Volatile oil model uses an extension of the *BLACKOIL option. Gas can exist in solution in the oil phase and oil can exist in the gas phase. This option is similar to the *GASWATER_WITH_CONDENSATE. It is more flexible in that the gas phase need not always exist. This allows the model to be used in situations where oil can be above its bubble point. The volatile oil model uses many of the same keywords as the *BLACKOIL model and GASWATER_WITH_CONDENSATE model. In addition, it introduces a new PVT table option *PVTVO, which includes both wet and dry gas properties. Please See the Tutorial section for more details in using the volatile oil model.

Subdomain-Dual Permeability Model

The SD-DK model is an extension of the Subdomain model which allows the Simulator to model gravity drainage and reinfiltration processes. Please see the *SUBDOMAIN, *FRACVOL, *TRANSD and *SD_REINF keywords for details. Also see the Tutorial section for a discussion on how to use the SD-DK model.

Water – Polymer Mixture Viscosity Table

Previously, water viscosity in polymer model runs were analytical functions of pure water viscosity and polymer concentration. Polymer–water mixture viscosity can now be defined using tables of relative polymer concentration versus water viscosity (through a water viscosity multiplying factor). Please see the *PMIX *TABLE keyword for more details.

Depletion Option

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. See the *DEPLETION keyword in the Reservoir Description section for more details.

Improved Aquifer Printout

The normal printout of initial aquifer properties has been extended to include the ratio of aquifer volume to reservoir volume. Aquifer volume includes both the volume associated with the analytical aquifer and the volume associated with entirely water filled blocks in the numerical model. Please see the *AQUIFER keyword in the Reservoir Description section for details.

Important Changes Between IMEX 2006.10 and IMEX 2005.10

NOTES FOR IMEX 2006.10

1. Seawater/Scale Buildup Option
2. TRIGGER Action Option
3. Improved Parallelization Performance using *JACDOMS
4. Well Definition Order Relaxed/Improved
5. *WLISTSHUT/*WLISSTOPEN Option
6. *WTMULT/*GCONIMULT/*GCONPMULT Options
7. Improvements to Tubing Head Pressure Calculations
8. *WOC_SW available with *DEPTH_AVE Initialization.
9. Reinfiltration Modeling for Fractured Reservoirs (Beta)
10. Binary File Reading Option
11. DPCONNECT

Seawater Injection/Scale Buildup Option

A Seawater Injection/Scale buildup option has been incorporated in IMEX. This feature is actually made up of two independent options: 1) a Seawater Injection Tracking Option and 2) a Scale Buildup/Well Damage model. The two models can be coupled to make well damage a function of both water rate and the fraction of seawater in the produced water.

The Seawater Injection option adds a seawater flow equation to the normal Black Oil or Oil Water Models in IMEX. This equation is solved simultaneously with the oil, water and gas flow equations. It is an active model, and the injected water may have a viscosity different from formation water. Thus it is possible to model seawater injection altering fluid flow patterns within the reservoir. This model and the Scale Buildup/Well Damage model are coupled through the *SCLDPS table (Scale Buildup/Well Damage Option) which relates scale deposition to the volume fraction of seawater in the produced water. The Seawater Option is controlled by the following new keywords:

- | | |
|-------------------------------------|---------------------------|
| 1. *MODEL *BLACKOIL_SEAWATER | Component Properties |
| 2. *MODEL *OILWATER_SEAWATER | Component Properties |
| 3. *OUTPRN/*OUTSRF *GRID *SEAWATFRC | Input/Output Control |
| 4. *CONVERGE *MAXRES *SEAWATER | Numerical Methods Control |
| 5. *INCOMP SEAWATER | Well and Recurrent Data |
| 6. *ALTERFS | Well and Recurrent Data |

The Scale Buildup/Well Damage Model allows the user to model well (layer) productivity damage based on the amount of water produced through each layer of a well. Optionally the damage can be made a function of the fraction of seawater produced with formation water.

This Option is controlled by the following new keywords:

- | | |
|------------------|-------------------------|
| 1. *SCLDPS | Rock Fluid Data |
| 2. *SCLDMG | Rock Fluid Data |
| 3. *SCLTBL-WELL | Well and Recurrent Data |
| 4. *SCLTBL-LAYER | Well and Recurrent Data |
| 5. *SCLRMV-WELL | Well and Recurrent Data |
| 6. *SCLRMV-LAYER | Well and Recurrent Data |

The Scale Buildup/Damage model can be used with any Property model available in IMEX (for example: API Tracking, Solvent, and Gas-water). It is only with the *BLACKOIL_SEAWATER and *OILWATER_SEAWATER models that the active seawater equation can be used to make scale formation a function of seawater fraction produced.

Trigger Action Option (Well and Recurrent Data)

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.

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.

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). Please see the *TRIGGER Keyword in the Well and Recurrent Data Section for a complete description of the option.

***JACDOMS Parallelization Option (Numerical Methods Control)**

*JACDOMS (-jacdoms on the command line) enables a new technique to control the Jacobian domains used for parallelization. This technique is generally more efficient than those previously used to control Jacobian Domains (controlled using the *DPLANES or *DTYPE keywords) and is recommended.

More Flexible Well Definition Ordering (Well and Recurrent Data)

It is now 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. Previously it was necessary to define a well’s type (injector or producer) immediately after its initial definition.

***WLISHUT, *WLISOPEN (Well and Recurrent Data)**

*WLISHUT and *WLISOPEN provide a means to temporarily shut a large list of wells and later re-open them without disturbing the pattern of shut and autodrillable wells that existed when the *WLISHUT keyword was read.

***WTMULT (Well and Recurrent Data)**

*WTMULT allows modification of any previously specified well constraint value for well(s) listed by well_numbers or 'well_names' by applying a multiplying factor.

***GCONIMULT, *GCONPMULT (Well and Recurrent Data)**

*GCONIMULT is used to specify multipliers which modify existing group injection target controls. *GCONPMULT is used to modify existing group production target controls with the use of multipliers.

Mechanistic/Momentum Wellbore Pressure Drop Model Improvements

The Mechanistic Wellbore pressure drop calculation in IMEX (*HEAD-METHOD ...

*MODEL-MOMENTUM) has been made more rigorous. The updated Momentum option is currently a full implementation of the Petalas-Aziz mechanistic model.

***WOC_SW in the *DEPTH_AVE Vertical Equilibrium Initialization Option**

*WOC_SW allows the user to specify non-zero oil saturation below the water oil contact.

Small non-zero water zone oil saturation is often useful in avoiding convergence difficulties if oil attempts to move into the water zone. Previously, when this option was required it was necessary to use *BLOCK_CENTER initialization.

Reinfiltration Model for Fractured Reservoirs (Beta - no Builder Support or Manual Entries)

The Subdomain-Dual Permeability-Reinfiltration (SD-DK) model is controlled by the following Keywords,

- *SUBDOMAIN turns on subdomain dual porosity model,
- *FRACVOL controls subdomain block thickness,
- *TRANSD controls SD-DK inter-subdomain connections, and
- *SD_REINF determines if SD-DK inter-subdomain connections are used to represent reinfiltration or gravity drainage.

Modeling capabilities include:

1. Standard Subdomain Model

If the *TRANSD keyword is not included in the model, *SD_REINF is not required and the model reverts to the standard *SUBDOMAIN model. It is still an excellent idea to use the new *FRACVOL keyword to minimize the error in the capillary holdup.

2. Gravity Drainage between Subdomains

Using the *TRANSD keyword to set non-zero transmissibility multipliers between domains allows the fluid in the bottom of a matrix subdomain to flow into the top of the matrix subdomain directly below it without having to flow into the fracture first. In essence this is a hybrid method which combines the Subdomain model and the Dual Permeability model and gives similar results. As there is no capillary holdup in the matrix, using *FRACVOL does not significantly improve the models results. As *TRANSD is grid block dependent, gravity drainage between subdomains can be modeled in user specified regions of the reservoir.

3. Reinfiltration between Subdomains.

As in 2) the non-zero *TRANSD keyword enables inter-subdomain flow. The use of *FRACVOL and *SD_REINF together enables the use of the bottom-most subdomain block as a proxy for a horizontal fracture between the subdomains. *FRACVOL is used to reduce the thickness of the bottom division of the subdomain which reduces the amount of fluid stored in the fracture proxy block and minimizes the error in capillary holdup.

*SD_REINF = 1.0 sets the bottom subdomain division's capillary pressure to 0.0 (the fracture's value) and introduces a real capillary discontinuity in the flow between the two subdomains. *SD_REINF is grid block dependent.

4. Gravity Drainage and Reinfiltration between Subdomains

Actual drainage from Subdomain blocks is a combination of gravity drainage and reinfiltration. This may be approximately accounted for by scaling back the bottom subdomain division's capillary pressure rather than setting it explicitly to zero.

This can be accomplished by setting *SD_REINF to a value between 0.0 and 1.0. The fracture proxy block's capillary pressure is scaled by $1.0 - *SD_REINF$.

See the document "SD-DK with Reinf.doc" in the DOC directory of your IMEX 2006.10 install directory for more details.

Binary File Reading Option

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 data set 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.

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.

*DPCONNECT Option

The DPCONNECT keyword allows the grid module to produce extra connection between blocks in a dual porosity/permeability system when either the matrix or the fracture is missing from one of the two blocks being connected.

When enabled, a dual porosity or dual permeability block with a missing fracture will connect the matrix to both the matrix and fracture of surrounding blocks. In addition a dual porosity or dual permeability block with a missing matrix will connect the fracture to both the matrix and fracture of surrounding blocks.

Important Changes Between IMEX 2005.10 and IMEX 2004.10

NOTES FOR IMEX 2005.10

1. Improvements have been made in frictional pressure drop calculations. Two new template data sets have been added to demonstrate some of these features: mxwwm028.dat and mxwwm029.dat
2. Manifold group options have been added. Four new templates have been added to demonstrate some of these features: mxwwm030.dat, mxwwm031.dat, mxwwm032.dat and mxwwm033.dat
3. Beta version of coupling to Petroleum Expert's Gap surface facility model is now available. Please see the installed 'Doc' directory for more information.
4. The *TRANSF keyword for modelling fault transmissibilities has been added.
5. A beta 64 bit Linux version is now available for Red Hat Linux Enterprise 3 for Intel EM64T enabled Xeon processors.
6. A beta 64 bit Windows version for Windows x64 for Intel EM64T enabled Xeon processors is available.
7. For parallel IMEX, options are now available to use up to 64 processors for a single run. Improvements have been made in load balancing with changed defaults for Parasol. More non-solver portions of the code have been made parallel.
8. The "maxsteps n" command line option has been added.

In the Reservoir Description Section, the keyword *TRANSF has been added. This keyword may only appear in this section and not in recurrent data. It allows the user to specify fault transmissibility multipliers which are independent of the multipliers specified in the *TRANSI, *TRANSJ and *TRANK keywords.

The use of the *MOD keyword has been altered. It is no longer permitted to have multiple *MOD keywords following a keyword. However a single *MOD keyword may contain as many mod definitions as necessary. For example:

```
*PERMI *CON 1000.00
*MOD 1:1 1:2 1:1 * 0.5
*MOD 2:2 1:2 1:2 * 0.25
```

is not permitted, while:

```
*PERMI *CON 1000.00
*MOD 1:1 1:2 1:1 * 0.5
2:2 1:2 1:2 * 0.25
```

is permitted.

Two major options have been added to IMEX's handling of groups and wells. The first is the frictional pressure drop option; the second is the manifold group control option.

Keywords associated with frictional pressure drop.

***HEAD-METHOD** well_list *GRAV-FRIC (*MODEL-AG |MODEL-HOMOGENEOUS| *MODEL-MOMENTUM | *MODEL-DRIFTFLUX) (*HEADRROUGH rrough) (*HEADSURFT surft)

This keyword describes which frictional pressure drop calculation method is used for wells in the well list.

***HEADITER** well_list (*ONE-STEP | *ITERATIVE)

This keyword specifies the calculation procedure for calculating the frictional pressure drop. The *ITERATIVE method allows the model to compute heads which are consistent with completion pressures. The *ONE-STEP method does not iterate to calculate consistent heads and very often results in unphysical results.

***BHPHEADINIT** well_list (*SEPARATE | *COMBINED)

This keyword when used determines whether well bottomhole pressures as well as layer heads are made consistent with completion pressures. This option is not normally required for good results as the HEADITER *ITERATIVE option is quite robust on its own. Using the *COMBINED option can result in significantly increased run times.

New keywords associated with the Manifold Model.

***MANIFOLD** (*PROD | *GASI | *WATI | *SOLI) group_list (*ON | *OFF)

Determines which groups are manifold groups.

***PCON-MANIFOLD** (*PROD | *GASI | *WATI | *SOLI) group_list (*ON | *OFF)

Determines whether rate constraints are translated into manifold pressure constraints.

***GPHYDDEP** (*PROD | *GASI | *WATI | *SOLI) group_list depths

This keyword adjusts the depth used to calculate the pressure drop between a surface compressor and a subsea manifold.

***GPTABLE** (*PROD | *GASI | *WATI | *SOLI) group_list

This keyword determines which table is used to calculate the pressure drop between a surface compressor and a subsea manifold.

***GROUPALQ** (*PROD | *GASI | *WATI | *SOLI) group_list

This keyword specifies the group ALQ property which is used in standard VFPPROD and VFPINJ tables which are associated with flow from manifold to compressor.

***WELLALQ** well_list

This keyword specifies the well ALQ property which is used in standard well VFPPROD and VFPINJ tables.

Other keywords which are modified to account for manifold groups are:

*GCONI, *GCONP, *INCOMP and *IWELLBORE

Important Changes Between IMEX 2004.10 and IMEX 2003.10

NOTES FOR IMEX 2004.10

1. A 32 bit Linux version is now available.
2. A technique for modeling tilted water oil contacts has been developed using PCOW_SHIFT (see tutorial section).
3. Multiple datum depths are supported
4. For parallel IMEX, options are now available to use up to 32 processors for a single run.
5. The “checkonly” option has been improved to read and process all of the well data
6. The 3 phase oil relative permeability when using Stone’s Second Model (the default) is checked for consistency at low oil saturations. Specifically IMEX now warns if non-zero K_{ro} is observed when oil saturation equals zero.
7. Warning messages are written for inconsistent capillary pressure curves.
8. PVT consistency checks have been extended.
9. Warning messages are written when PBT table minimum and maximum depths alter the WOC or GOC of a PVT region.
10. Two new well group monitor constraints have been added.
11. The maximum number of characters in a well name has been increased from 16 to 40. The number of characters in group names, sector names and leasesline names have not changed and remain at 16.
12. The maximum number of characters per line in the data file has been increased from 130 to 512.

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 2004.10

Input/Output Control

***OUTPRN/OUTSRF *RES *PCWSHF *JFWSHIF**

These keywords enable the output of capillary pressure and J function shift/offsets.

Rock-Fluid Data

***PCOW_SHIFT/*JFW_SHIFT**

This keyword enables the input of water capillary pressure or water J function shifts. This allows the user to model tilted water contacts.

Initial Conditions

***DATUMDEPTH**

The DATUMDEPTH keyword can now be applied separately to each PVT region. Within each PVT region different calculation options may be used.

Well and Recurrent Data

***GCONM *MAXGAS**

These keywords identify a maximum surface gas rate monitor for group production. The *STOP, *SHUTALL, *SHUTMOW, *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are valid for this monitor. 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.

***GCONM *MAXSTW**

These keywords identify a maximum water rate monitor for group production. The *STOP, *SHUTALL, *SHUTMOW, *SHUTMOL, *SHUTMOLDOWN, and *SHUTMOLUP actions are valid for this monitor. The “most offending” well or layer for this constraint is deemed to be the one with the highest WCUT rather than the one with the highest water rate.

Important Changes Between IMEX 2003.10 and IMEX 2002.10

NOTES FOR IMEX 2003.10

1. Peak storage requirements have been drastically reduced. For large 32-bit and large 64-bit versions the peak storage requirements are approximately one-half of what they were for 2002.10.
2. Support for very large 64-bit problems has been added and tested on problems up to 112,000,000 grid blocks.
3. For parallel IMEX, options are now available to use up to 16 processors for a single run.
4. The sector output, both to the ascii file and the SR2, now includes average P/Z values weighted according to both block pore volume and block hydrocarbon pore volume. In order to use this feature, reservoir temperature must be defined (see *TRES)
5. Streamline output is enabled for display in results.
6. Gas condensate modeling capability has been added.
7. Scaling of relative permeability, capillary pressure and J function maximums (i.e. Krw at irreducible oil, K_{row} at connate water, etc) has been included.

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 2003.10

Input/Output Control

***OUTSRF GRID *STRMLN**

This keyword enables the output of interblock velocities to allow Results3D to display Streamlines. See the results manual for more information on Streamline display. Due to the large size of the SR2 output, streamline display is not enabled when the *ALL output option is used with OUTSRF GRID. Streamline output must be explicitly requested.

***OUTPRN/OUTSRF *RES *KRW SCL *KROWSCL *PCWSCL *KRG SCL *KROGSCL
*PCGSCL**

These keywords enable the output of initial relative permeability, capillary pressure and J function maximums.

***OUTPRN/OUTSRF *GRID *KRW SCL *KROWSCL *PCWSCL *KRG SCL *KROGSCL
*PCGSCL *RS *BO *EG *RV *DPP**

These keywords enable the output of relative permeability, capillary pressure and J function maximums and the output of Rs, Bo, Eg, oil content Rv (for condensates) and dew point pressures (for condensates) in recurrent output. The relative permeability table maximums are only output at times other than the initial time if they are explicitly requested on the OUTPRN GRID or OUTSRF GRID keywords, the *ALL output option will not enable their output.

***PSPLIT *ON**

This keyword enables the output of surface production splits for black oil and condensate models. In a black oil model the gas production is split into gas at surface from solution gas in the reservoir and gas at surface from free gas in the reservoir.

In a condensate model, in addition to the splitting of the surface gas production, oil production is also split. Oil is split into oil from gas in the reservoir and oil from mobile oil in the reservoir.

Reservoir Description

***ISECTOR**

***SECTORNAMES**

These keywords enable an alternate method of entering sector information. Use of these keywords may make sector data conversion to IMEX from other simulators more straightforward.

Once read into MODELBUILDER, these keywords are translated into SECTORARRAY keywords.

Component Properties

***MODEL *GASWATER_WITH_CONDENSATE**

This keyword enables the gas water with condensate model.

***PVTCOND**

This keyword signals the input of the main condensate PVT table. Saturated gas and condensate properties are entered in this table.

***BOT**

This keyword enables the input of under saturated Bo. This allows the user to directly input Bo as a function of pressure and bubble point pressure.

***VOT**

This keyword enables the input of under saturated oil viscosity. This allows the user to directly input oil viscosity as a function of pressure and bubble point pressure.

***EGUST/*BGUST/*ZGUST**

These keywords enable the input of Gas Eg/Bg/Zg for pressures above the dew point when using the GASWATER_WITH_CONDENSATE option.

***VGUST**

This keyword enables the input of Gas viscosity for pressures above the dew point when using the GASWATER_WITH_CONDENSATE option.

Rock-Fluid Data

***KRWIRO *KROCW *PCWMAX *JFWMAX *KRGCL *KROGCG *PCGMAX
*JFGMAX**

These array keywords enable the input of relative permeability table maximums on a block by block basis in initial and recurrent data.

The user can now enter the values of K_{rw} at irreducible oil, K_{row} at connate water, P_{cow} (or water J Function) at connate water, K_{rg} at connate liquid, K_{rog} at connate gas and P_{cog} (or gas J function) at connate liquid for each block. This can be done both in the rock-fluid data section and in the recurrent data section.

Initial Conditions

***PDEW**

***PDEWT**

These keywords enable the input of initial block dew point pressures or initial dew point pressure versus depth tables for each PVT region. When using the GASWATER_WITH_CONDENSATE option initial dew point pressures must be defined.

Numerical

***NORM *PDW**

***MAXCHANGE *PDW**

These keywords enable the input of timestepping control based on the changes in dew point pressure during the previous timestep. These keywords can only be used with the GASWATER_WITH_CONDENSATE option.

***CONVERGE *MAXRES (*OIL|*WATER|*GAS|*OTHER|*SOLVENT|*POLYMER|
*LIGHTOIL)**

These keywords enable the control of maximum residual based convergence tolerances by individual phase or component. This allows more precise control of a run. In addition, when a higher level of output information is selected (see *WPRN *ITER) each phase's maximum residual error is displayed.

Well and Recurrent Data

***PTUBE *CONDENSATE**

This keyword enables the use of condensate tubing head pressure tables. These tables can be used with the GASWATER_WITH_CONDENSATE option.

Important Changes Between IMEX 2002.10 and IMEX 2001.10

NOTES FOR IMEX 2002.10

The recommended procedure for running on a dual processor Pentium III or Pentium 4 is to use the two command-line arguments “–doms –parasol” described in “Numerical” below.

The “–doms” switch enables parallelization of the Jacobian building and some other non-solver portions of the code.

The “–parasol” switch enables the use of the parallel linear solver, Parasol, instead of the usual non-parallel solver, AIMSOL.

The use of either of these parallel IMEX options requires that you have a license with “Parallel IMEX” enabled.

Parasol will in general give slightly different answers than AIMSOL, as it is a different iterative solver.

There are a number of additional options related to Parasol and the parallel Jacobian building. These are described below and in the Numerical Methods Control section of this manual.

The “-aimsol” command line switch overrides PARASOL related keywords and forces the use of AIMSOL.

This version has a correction for the *SCONNECT Keyword when used with dual porosity and dual permeability models.

The use of *MODSI units has been corrected in the aquifer model.

The *MAXRES convergence tolerance has been made more consistent when used with the Polymer option. This will correct problems encountered in convergence when dispersion coefficients are used.

The 2002.10 release changes the time unit of *DISPI, *DISPJ and *DISPK from seconds to the unit defined by the users choice of *INUNIT. The user must change data accordingly.

A problem with WHP controlled wells getting unphysically large pressure derivatives has been corrected.

The use of the *ONTIME feature with *WHP controlled wells has been corrected.

If the head is changed for a temporarily shut-in well, then, if it is reopened during the same timestep it is reset to its original value.

If a well has its constraint type changed during apportionment, its bottomhole pressure is re-initialized. This produces more consistent results.

For pseudo-miscible runs, there may be a slight drift in answers. Smoothing for So-Sorms has been made more consistent.

Handling of oil phase disappearance has been made more consistent.

For non-physical pressures (less than zero), if they occur in a block which contains a well, the corresponding well is temporarily shut in.

The handling of non-physical pressures during convergence has been improved.

The crossflow handling of injectors has been improved to correctly account for crossflowing solution gas.

Special release 64 bit versions of IMEX have been created for Itanium WinXP 64-bit and Power 3, 4 AIX 5.1 IBM RS6000's.

The *MODEL *LIGHTOIL Option is undergoing major revision and is not currently available. Please contact CMG Support for more information.

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 2002.10

Numerical

Command line arguments:

–doms

which is equivalent to

*DPLANES

and overrides the DTYPEDPLANES keywords in the data set, and

–parasol

which is equivalent to the

*SOLVER *PARASOL

*PPATTERN 2

keywords

–aimsol

which overrides any parasol keywords in the Numerical Methods Control section and forces the use of AIMSOL.

Non-solver parallelization of IMEX:

***DTYPE**

Complete storage grid array input of Jacobian domain numbers.

***DPLANES imxdom**

imxdom target number of planes per domain – Planes are chosen in the dimension with the largest number of non-trivial planes. imxdom is the number of corresponding non-trivial planes in this direction per domain. If imxdom not specified and DPLANES is, then the default is 4.

***PNTHRDS**

Number of threads to be used - if parallel Jacobian building and Parasol are not specified, then defaulted to one. If parallel Jacobian building or Parasol is specified, then it is defaulted to the min of (the number of processors in current machine, 2).

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.

Keywords related to PARASOL:

***SOLVER (*AIMSOL | *PARASOL)**

Choose which solver to use, AIMSOL or PARASOL.

Default if no *SOLVER keyword is *SOLVER *AIMSOL.

***PNSUBD nclas2**

Choose the number of PARASOL classes used for vector operations in the GMRES iteration. Default is 2.

***CHECKRB (*ON | *OFF)**

When ON, red-black ordering is abandoned for a PARASOL class in which the fraction of red blocks is too small.

Default is OFF.

***PDEGAA idegaa**

Factorization degree to be used within PARASOL classes. Default is the value of *SDEGREE (1 for red-black, rcmrb; 2 for natural, rcm).

***PDEGAB idegab**

Factorization degree to be used between PARASOL classes. Defaults to idegaa + 1.

***PPATTERN (ipatr | *PARTITION | *PPARTITION | *GPARTITION | *APARTITION)**

PPATTERN sets the basic partitioning of the reservoir into non-connected regions and separators that makes possible the parallelization of the linear solution. Default is *PPATTERN 2.

Please see the Numerical Methods Control section for a complete description of the *PPATTERN keyword.

Well and Recurrent Data

There are two major additions to the well data for this release: The introduction of reporting groups (the *REPORTING-GROUP keyword) and the addition of a bottomhole fluid (BHF) group production rate target and monitor.

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.

***REPORTING-GROUP ‘reporting_group_name’ well_list ‘weight_list’**

*REPORTING-GROUP allows the user to define a set of wells with differing membership weights. No Group Control can be specified for a reporting group, but there are no restrictions placed on well membership in reporting groups. The weight_list allows the reporting group to handle element of symmetry output.

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.

```
*GCONP      'group_name_1' ... 'group_name_n'  
          *MAX    *BHF    value    action  
*GCONP      'group_name_1' ... 'group_name_n'  
          *TARGET   *BHF    value    action  
*GCONP ... *MAX *BHF and *GCONP ... *TARGET *BHF : allow  
the user to control bottomhole fluid group production rates.  
*GCONM      'group_name_1' ... 'group_name_n'  
          *MINBHF  value   (*STOP | *SHUTALL)  
*GCONM ... *MINBHF: allows the user to monitor group bottomhole  
fluid group production rates.  
*GUIDEP     *BHF    'group_or_well_list'    'guide_rates'  
*GUIDEP *BHF: allows the user to specify guide rates proportional to  
bottomhole fluid rates.
```

There is also a minor keyword addition in this section. The keyword *NWFHYS 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.

```
*NWHYFS  
      well_list  
      Integer_list
```

Important Changes Between IMEX 2001.10 and IMEX 2000.10

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 2001.10

Input/Output Control

***SBDZ**

IMEX has added subsidence to the grid_list for *OUTPRN *GRID.

***SBDZ**

IMEX has added subsidence to the grid_list for *OUTSRF *GRID.

See the Tutorial Section for more information

The user should notice that the display of well rates has been sped up greatly in ResultsGraph, this is due to a ResultsGraph optimization which makes use of extra information the simulator writes to the SR2 binary files.

The *LIST/*NOLIST options for displaying echoed data has been altered. The default under the *LIST option is to display the first 20 lines of data, the remainder of the data is not echoed. This default action can be overridden by placing the keyword *NOLISTLIM in the Input/Output control section.

When output of reservoir condition well variables is enabled, the voidage replacement ratio defined by the group control option is output to the text output file and to the SR2 binary file for display using Results Graph.

Reservoir Description

The grid module routines have been sped up significantly in this release. The improvements will be most evident in large corner-point models where initialization can now take place using approximately one-third the CPU time.

The grid model can now automatically handle the accidental user refinement of null blocks. Previous versions of the Grid Module would signal an error if this occurred.

The handling of pinchouts has been made consistent between corner point and non-corner point grids. “Pinchout out” connections that may have been previously missed will now be correctly made.

Component Properties

Water properties can now be defined per PVT region. Oil viscosity change with pressure (*CVO) can now be defined per PVT region. The input structure is both backwards compatible and more flexible than previous versions. The tutorial section has been altered to reflect these changes, please see “Defining Multiple PVT Regions” in the Tutorial Section.

All oil and water properties per PVT region are echoed in the simulator output file (.out).

In addition, it is now possible to define an oil compressibility for each PVT region using *CO. It is no longer necessary to include a *COT table for each PVT region or include oil compressibility as a function of bubble point pressure in each PVT table to accomplish this.

Rock-Fluid Model

Capillary pressure hysteresis in the P_{cog} table and Krow hysteresis has been added to the simulator. See the keywords below and the *SGT/*SLT keywords for a detailed description of the options.

***HYSKRO**

K_{row} hysteresis using Carlson's model (non-wetting phase).

***EPSPCG**

P_{cog} hysteresis using Killough's model.

***RPT *IMBIBITION (*PCOW *PCOG *BOTH)**

The user can specify which curve (imbibition or drainage) to begin the simulation on. The specification is done independently for P_{cow} and P_{cog} .

The *SWTKRTHR and *SLTKRTHR keywords are no longer sub-keywords of the *SMOOTH keyword (for *SWT and *SLT/*SGT). They are now sub-keywords of *SWT and *SLT/*SGT. This allows the user to change the threshold saturation for endpoint determination without using the smoothing options.

Initialization

The initialization input has been made more flexible when using multiple PVT regions. Now multiple PVT region Contacts (such as WOC or GOC or WGC) can be placed on a single *DWOC, *DGOC or *DWGC keyword. In addition, multiple PVT region reference depths and pressures can be placed on a single *REFDEPTH or *REFPRES keyword. Please see "Defining Multiple PVT Regions" in the Tutorial Section for more details.

Contacts and reference pressures are now echoed in the simulation output file (.out)

Numerical

*AIM *STAB *AND-THRESH is now the default adaptive implicit method used. This was changed from *AIM *STAB.

Neighbours of implicit blocks have their IMPES stability checked once per timestep, however; neighbours of blocks that have just switched from IMPES to implicit on that timestep are no longer checked. The *AND_THRESH sub-keyword ensures that all IMPES blocks have their maximum saturation and pressure changes checked every iteration for IMPES to implicit switching.

The result of this change is generally a more stable simulation.

Well and Recurrent Data

***GEOA and *KHA**

The anisotropic Peaceman well model has been added to the simulator. This model accounts automatically for permeability anisotropy and grid block aspect ratio. It is available on the well 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 as it is calculated internally.

This feature is especially useful when used with the *LAYERXYZ and *LAYERIJK deviated well options.

*LAYERIJK

The new *LAYERIJK keyword allows the user to override, on a layer basis, the well direction input on the *GEOMETRY keyword.

The *LAYERXYZ allows the user to enter trajectory information into the simulator through Builder. In addition, *LAYERXYZ information can be synthesized from Builder without requiring trajectory data.

However, for simple well trajectories, the *LAYERIJK keyword allows the user to model some aspects of well deviation without using builder or requiring trajectory files.

*GMKUPTO *WMKUPTO *SMKUPTO

*GMKMAXR *WMKMAXR *SMKMAXR

*GRECYMAXR *WRECYMAXR *SRECYMAXR

*GPRODGROUP*WPRODGROUP

*SPRODGROUP *GSALESFRAC *GSALESMAXR

*GFUELFRAC *GFUELMAXR

Enhancements to the group recycling options have been made in this version of IMEX. In addition to reducing the amount of gas injected due to fuel and sales gas, options to separately specify maximum recycling rates and total injection rates has been implemented for gas, water, and solvent recycling.

See the *GCONI keyword in this manual as it describes in detail the recycling calculation. In addition, see the manual entries describing the recycling options listed above.

*BHPDEPTH *BHPGRAD *LAYERGRAD

These new keywords allow the user to specify a depth to which a well's bottomhole pressure is referred to.

The use of quotes in well names is now supported in a consistent manner. To use a double quote in a well name, use a single quote as a well string delimiter. To use a single quote in a well name, use a double quote as a well string delimiter.

Important Changes Between IMEX 2000.10 and IMEX 1999.10

FORGAS Integration

Previously, if any well had any change (e.g. ALTER), even a non-FORGAS well, then all FORGAS wells were reset for the following timestep to the current IMEX specification, over-writing the FORGAS values. Now, even if a FORGAS well is ALTER'd, except for the time where it is initially defined, the *ALTER will be ignored and the value from FORGAS will be used.

The IMEX calculated values for a and b for the IPR calculation for FORGAS are now based, for wells perf'd in more than one grid block, on the surface mobility weighted average delta P's (or delta P squared for quadratic pressure) rather than the value at the reference layer. Also, the reservoir pressure passed to FORGAS was formerly the reservoir pressure in the reference layer. Now it is the surface mobility weighted average of (reservoir pressure - head). This new approach, in some cases, significantly reduces oscillations in rates.

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 2000.10

Input/Output Control

Sectors are now viewable in RESULTS. When probing a block in RESULTS, the sectors the block is in will be displayed. In addition, It is possible to select viewable regions by sector in both 2D and 3D views.

Command Line Options:

IMEX has the new command line option

-log Directs screen input to a log file which 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. In addition, the user *INTERRUPT *INTERACTIVE choice is also entered in the ".in" file when the -log option is used. A carriage return in the ".in" file where the simulator is expecting to read the interrupt option defaults to *RESTART-STOP.

*NDBETCR, *WSRFTN, *GSRFTN, *CRSETN, *SWCON, *SWCRIT, *SOIRW, *SORW, *SGCON, *SGCRIT, *SLCON, *SOIRG, *SORG

IMEX has added the following variables to the res_list for both *OUTPRN *RES and *OUTSRF *RES. *NDBETCR displays the non-Darcy correction factor, *WSRFTN and *GSRFTN display the water-oil and gas-liquid surface tension arrays respectively. *CRSETN displays the compaction rock type of each block. The remaining keywords display water-oil and gas-liquid relative permeability table end point arrays.

*PCOW, *PCOG, *RFG, *SWCON, *SWCRIT, *SOIRW, *SORW, *SGCON, *SGCRIT, *SLCON, *SOIRG, *SORG

IMEX has added the following variables to the grid_list for both *OUTPRN *GRID and OUTSRF *GRID. *PCOW and PCOG display the capillary pressures of each block, *RFG displays the non-Darcy flow reduction factor. The remaining keywords display water-oil and gas-liquid relative permeability table end point arrays.

*WELL-SECTOR

This keyword specifies that well in sector information will be printed out.

Well in sector information consists of two tables. The first is ordered by sectors (sector table) and lists all active wells within a sector (an active well has at least one perforation open). The second table is ordered by well (well table) and lists all sectors associated with each active perforation of every well.

Wells which have all perforations closed or have not had any perforations defined will not be listed in either table. Individual perforations that are no longer active will not be listed in the well table. Perforations which have had their production/injection zeroed due to use of the *XFLOW-MODEL *ZERO-FLOW keyword will not be listed in the well table for as long as their production/injection is zeroed. Active perforations of wells that are “shut in” will be listed in the well table, but “shut in” wells are not listed in the sector table.

The table information is printed out every timestep to a separate file with extension “sct”, the file will be created if when scanning the data IMEX finds the *OUTPRN *WELL-SECTOR keywords. The amount of printout can be controlled by using the *NONE sub-keyword in recurrent data.

Reservoir Description

Sector calculations in some instances were taking excessive amount of run-time. This was due to an unfortunate choice of algorithm for sector lookup during sector calculations. This algorithm was revised, resulting in remarkable improvements in run-time. For example, on a field problem with 159 sectors, total run-time was reduced by a factor of 2.56 with an increase in storage allocated of only 0.4%.

The compaction option has been revised to remove several limitations. Compaction can be used with dual porosity and dual permeability models. In addition the following changes were made:

1. When values for the third column in a rock table are blank or negative, horizontal permeability multipliers are defaulted to one.
2. When values of the fourth column in a rock table are blank or negative, vertical permeability multipliers are defaulted to be the same as the corresponding horizontal permeability multipliers.
3. When a value of a permeability multiplier is greater than or equal to zero and less than 1.0E-04, it is replaced by a value of 1.0E-04.

4. When a value of a porosity multiplier is less than 1.0E-02, it will be replaced by a value of 1.0E-02. Small values of porosity may result in poor performance of the simulator.
5. If there is only one rock type and there is no keyword *CTYPE used, all the grid blocks are assigned to have that rock type.
6. There is no longer a limit to the number of compaction rock types which may be used in a data set.
7. There is no longer a limit to the number of rows in a table permitted under the *CROCKTAB or *CROCKTABH keywords.
8. There is no limitation in the number of hysteresis curves for a given rock type. However, the first row of each sub-table under the keyword *CROCKTABH must be identical to the corresponding row on the main path from which the hysteresis path starts as discussed in the user manual.
9. The *CROCKTYPE region definitions may exist for region numbers not specified for any given block using the *CTYPE keyword. However, if a block is set to a particular region type number using *CTYPE, then there must exist a corresponding *CROCKTYPE definition.
10. The compaction rock region types (input using *CTYPE) may now be output using the sub keyword *CRSETN with *OUTPRN *RES and/or *OUTSRF *RES.

A dilation option was added, see the Reservoir Description section and the Tutorial section for further information.

A pseudo-capillary pressure option was added to the dual porosity and dual permeability option. See the keyword *TRANSFER for a complete description of the option.

The *PINCHOUTARRAY keyword was added to the reservoir description section. This allows for an alternative method of pinchout definition.

Component Properties

The use of surface tension input on *PVT tables has been extended to work with multiple PVT regions. The surface tension is used to correct P_{cog} with respect to pressure. A single reference pressure is input for the surface tension correction, but this reference pressure *REFPST is used to calculate a separate reference surface tension for each table.

The use of oil compressibility (C_o) tables for oil above the bubble point at any pressure has been supported in IMEX for some time. However, the conversion of B_o 's above the bubble point to C_o 's is not trivial as IMEX replaces the original C_o 's with integrated averages. We have added the ability of automatically converting B_o 's to C_o 's to ModelBuilder. The user may now use ModelBuilder and enter either B_o 's or C_o 's above the bubble point.

Rock-Fluid Model

*SWCON, *SWCRIT, *SOIRW, *SORW, *SGCON, *SGCRIT, *SLCON, *SORG

The end point scaling option had been completely rewritten and redesigned. Instead of the four scaling points: 1) connate water (*SWCON previously *SWC), 2) critical gas (*SGCRIT previously *SGC), 3) residual oil in the oil-water system (*SORW), and 4) residual oil in gas-liquid system (*SORG). We now scale with respect to:

Connate Water Saturation	(*SWCON)
Critical Water Saturation	(*SWCRIT)
Connate Gas Saturation	(*SGCON)
Critical Gas Saturation	(*SGCRIT)
Irreducible Oil Saturation (oil-water system)	(*SOIRW)
Residual Oil Saturation (oil-water system)	(*SORW)
Connate Liquid Saturation (gas-liquid system)	(*SLCON)
Residual Oil Saturation (gas-liquid system)	(*SORG)

There is a complete description of the new option in the Rock-Fluid section under the keyword *SWCON.

Use of the new scaling produces more consistent scaling of the relative permeability tables and initialization.

The use of the new end point scaling option may alter previous results. For the sake of compatibility, we have allowed the older four point end point scaling method to be used (see *RPT *SCALING-OLD).

A Leverett J Function option has been implemented into IMEX. 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. In addition the effect of pressure on Pcog can be accounted for by entering surface tension as a function of pressure on PVT tables. See the keyword *JFUNC, *SRFTNW, *SRFTNG, *PVT, *REFPST for details on the use of this option.

Non-Darcy gas flow in the reservoir can now be modeled in addition to non-Darcy flow into the well (see *TURB on the *PERF card). The non-Darcy reservoir gas flow model uses the Forchheimer equation with the Forchheimer number calculated using any of 1) Geertsma's 1974 correlation (*NONDARCY *GEERTSMA), 2) Frederick and Graves first correlation (*NONDARCY *FG1) and 3) Frederick and Graves second correlation (*NONDARCY *FG2). The Forchheimer number for gas can be further modified by the scaling parameter defined on the *NDARCYCOR keyword.

See *NONDARCY and *NDARCYCOR for more details.

Initialization

The Initialization section has no obvious changes, but the new end point scaling option and the Leverett J Function option both can alter initialization results.

Well and Recurrent Data

IMEX now allows the user to perforate null blocks. Previous versions of IMEX stopped and reported errors when perforations went through null blocks. This new feature allows perforations in null blocks, the perforations are closed automatically (*NULL-PREF). This feature is on by default.

IMEX allows the user to enter the layer X, Y and Z coordinates of the entrance and exit location of a well perforation running through a grid block. With this information IMEX will calculate an accurate deviated well productivity/injectivity index (*AYERXYZ).

The *KPERF keyword allows the user more flexible control of perforation relative permeability arrays.

The *GCOMP and *GCONM keywords now allow the user to shut all wells in a group when group constraints are violated.

Well reference layers can be defined anywhere in a branching well.

The *PERF keyword now can handle all refined grid situations including hybrid grids. The *PERFRG, *PERFVHY and *PERFHY keywords are valid but no longer required.

The *ON-TIME information of each well is now written to the SR2 file system. This now allows RESULTS to display instantaneous rates as well as average rates.

The *FULLY-MIXED wellbore crossflow model is now the default for all wells.

Important Changes Between IMEX 1999.10 and IMEX 98.05

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 1999.10

Input/Output Control

User block addresses (UBA) are used throughout the output file. No longer are block addresses printed out in packed storage or complete storage format. UBA: I, J, K format is used consistently throughout the printout.

Command Line Options:

IMEX has several new command line options.

'-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' Data Directory

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' Working Directory

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.

***PVTSETN**

IMEX has added the following variable to the res_list for both *OUTPRN *RES and *OUTSRF *RES.

***FLUXSC, *FLUXRC, *VELOCSC, *VELOCRC,**

IMEX has added the following variables to the grid_list for OUTSRF *GRID. The above keywords allow RESULTS to plot velocity and flow vectors on top of displayed properties. The above can also be displayed for each direction, but their primary purpose is for the generation of flux/velocity vectors.

***POROS, *PERM, *KRSETN**

IMEX has added the following variables to the grid_list both *OUTPRN *GRID and OUTSRF *GRID. *PERM displays permeability at each desired output time, permeability is now a function of pressure and so varies with time.

***WSRF *SECTOR**

Lease line output to the SR2 system has been added. It is controlled by the *SECTOR frequency of SR2 output.

Reservoir Description

Note: Grid Module keywords appearing in the Well or Recurrent Data section between two *DATE or *TIME cards must be grouped together. Failure to do so could allow the Grid Module to read only part of the data. The cards in question are *TRANSI, *TRANSJ, *TRANSK, *TRANLI, *TRANLJ, *TRANLK, and *REFINE (RANGE).

The Grid Module has generally been sped up for most cases and now has improved tolerance of odd corner point grids. Refined corner point grids are now allowed.

In addition, the algorithms for handling the connections made by sloping faults has been rewritten and improved.

The method for handling transmissibility multiplier inheritance to refined grids has been improved.

The handling of reservoir porosity versus pressure has been greatly extended in the IMEX simulator. Multiple rock regions have been added as well as the use of pressure tables for both porosity and permeability (vertical and horizontal) multipliers.

In addition the modelling of porosity and permeability hysteresis and irreversibility vs. pressure allows IMEX to model compaction. See the keywords *CROCKTYPE, *CROCKTAB, *CROCKTABH, *IRREVERS, *CCPOR, *CPRPOR and *CTYPE for a complete explanation.

Component Properties

IMEX includes significant additions to the Component Property section.

A gas-water (2-equation subset) has been added to IMEX. This allows a simpler data input format to model gas-water problems as well as reduced execution times to run gas-water problems. It is no longer necessary to enter fictitious oil properties to model gas-water systems. Only gas and water properties need be entered. Gas-water component properties keywords are listed below:

***MODEL *GASWATER**

This keyword signals IMEX to use the 2-equation gas-water model subset of the black-oil equations.

***PVTG**

This keyword is used to input a gas PVT table (for use with GASWATER option). Gas formation volume factor (B_g , E_g , or Z_g) and gas viscosity are tabulated against pressure.

Rock-Fluid Model

***SWT, *SLT, *SGT**

The gas-water option triggers slightly different input in each of the above tables. The tables no longer require input of either K_{row} or K_{rog} . P_{cow} is replaced by P_{cgw} . P_{cog} is not entered.

The gas relative permeability hysteresis option and the P_{cow} hysteresis options have been reviewed and improved. MXSMO024.dat has been added to the template directory to highlight the P_{cow} hysteresis option. This new template duplicates the radial test example in

J.E. Killough's classic paper "Reservoir Simulation with History-Dependent Saturation Functions", SPEJ, Feb. 1976, pp. 37.

Initialization

The Initialization section has no obvious changes, but the gas-water option allows the user to use the option:

*VERTICAL *BLOCK_CENTER *WATER_GAS

when nonzero Pcgw is entered in the *SWT table. Previous versions of IMEX forced the user to use:

*VERTICAL *DEPTH_AVE *WATER_GAS *TRANZONE *ON

to model gas-water capillary pressure. It is still required to use the *TRANZONE *ON option with the *GASWATER option when *DEPTH_AVE initialization is required.

Well and Recurrent Data

IMEX includes a significant modification to the well module simulator interface. This modification allows the well module better access to simulator information. This in turn allows the well module to more accurately calculate well and group operating constraints.

As a consequence of the above, the *CONT *REPEAT subkeywords on operating and monitor constraints are now operational. In addition, the well gravitational head calculation has been moved to a position within the code which ensures the well module and the simulator use identical heads.

Improvements have been made to how the well module extrapolates tubing head pressure tables; more consistent results will now be obtained.

The well index printout (available when using *WPRN *WELL *ALL) has been greatly enhanced. In addition to well index, the actual layer and total well fluid productivity/injectivity is displayed.

As well as being able to display well productivity the user is now able to directly input fluid phase or total well productivity/injectivity or productivity/injectivity multipliers using the *SETPI keyword listed below.

*SETPI

SETPI can be used to alter the well productivity to measured values or by a multiplicative factor correcting the last or the fundamental productivity.

*SETPI type *well_list values*

type = *OIL | *WAT | *LIQ | *GAS | *TOT | *MULT | *MULTO

Each time a *SETPI keyword is read, wells on the well list have their fundamental productivity corrected to match the user input value. The correction is only applied when the *SETPI keyword is read (i.e. when the test was done). Well productivity will normally change as fluids move in the vicinity of the well. However, productivities can be corrected at any later time by using the *SETPI keyword again.

Important Changes Between IMEX 98.05 and IMEX 98.00

For the 98.05 release the interface between the simulator and well management module has been improved so that the action of repeating a timestep in response to a well constraint violation works more smoothly than in past releases. This is now the strategy of choice if, for example, it is extremely important that a certain well's bottomhole pressure never fall below a certain value.

Since certain constraint checks and constraint actions may be handled slightly differently in the new version of IMEX, some data sets may run differently than with earlier versions. The user should examine output carefully when comparing runs made with earlier versions of IMEX.

The default mode of the WHP well constraint is now *IMPLICIT instead of the former default of *INITIALIZE. IMEX users have for the most part preferred the *IMPLICIT mode. Please see the documentation for the *OPERATE keyword for more information.

Two new well-related keywords have been introduced:

***WORKOVER-DATA:**

This keyword is intended to allow the user more direct control over the parameters determining the workover action which can be imposed in response to certain well monitor violations. It is intended to replace the entry of these parameters after the *WORKOVER action subkeyword of the *MONITOR keyword, which in the past was the only way to enter these parameters; the old way, however, is still supported and documented. Please see the manual pages for the *MONITOR and the *WORKOVER-DATA keywords for more information.

***MRC-RESET:**

This keyword allows the user to specify on a well- by-well basis whether a well should be tested for and set to operate upon its most restrictive constraint after a constraint value has been changed, or whether it should continue with its current operating constraint and switch only if a constraint is violated at the end of a timestep. The default is to check and switch to the most restrictive constraint whenever a constraint value is changed through data (through the *OPERATE, *ALTER, or *TARGET keywords), which can save the simulator the sometimes difficult task of trying to cause a timestep to converge with an unrealistic constraint; however the option of not resetting is allowed in case that the user wishes to enforce operation upon a certain constraint for the timestep. Please see the manual page for the *MRC-RESET keyword for more information.

Important Changes Between IMEX 98.00 and IMEX 97.00

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 98.00

Input/Output Control

Command Line Options:

IMEX 98.00 has several new command-line options.

- `-f input_data' Defines the input data file.
- `-r input_restart' Defines the restart irf file.
- `-checkonly' Runs the model in check only mode. An IMEX license is not required for a check only run specified from the command line.
- `-dimsum' Prints detailed information on model dimensioning at run time.
- `-onestep' Runs the model for one timestep only.

***MAX_GRIDS**

The IMEX 98.00 model now has a *DIM *MAX_GRIDS subkeyword to change the default number of refined grids.

***SR2PREC**

The SR2 binary file is by default written out in double precision mode. Single precision output is also possible. Single precision halves the size of most SR2 files. Restart information is slightly effected by the loss of precision.

***OUTPRN *GRID *APIGRAV**

The API tracking model allows the printout of light oil volume fraction for each block using the *OUTPRN *GRID *APIGRAV print option.

***LSOLVER**

*DEBUG *LSOLVER *ON turns on the debug output of the number of solver failures.

Reservoir Description

Both multilevel refinements and refinements defined in recurrent data have been added to IMEX 98.00. This represents a major improvement in how the user may use IMEX to model complex reservoirs.

A more robust handling of faults has been included as well as a more general implementation of the corner point grid option.

The analytical aquifer model has been reworked, introducing a new option (Fetkovitch aquifer) and improved Carter-Tracy influence function extrapolation and more realistic defaults.

A feature to input corner point tolerances has been included to allow the user to tune whether blocks are in contact with each other. In addition, this feature can be used to define a distance within which a blocks corner is assumed to be pinched out.

Sectors can now be input as an array to further simplify input.

The user can now explicitly define 'special connections' between pairs of blocks along with a transmissibility for the connection.

*REFINE

The *REFINE keyword may also occur as part of recurrent data. Refined blocks may appear later in a simulation. *REFINE must be used to set up refinements defined in recurrent data prior to using the *PERFRG keyword which refers to them.

*RANGE

The *RANGE keyword can now refer to single multilevel refinement blocks as well as to groups of fundamental blocks.

*AQMETHOD

The Fetkovitch aquifer calculation has been added to the analytical aquifer calculation (*AQMETHOD *FETKOVITCH).

The Carter-Tracy aquifer calculation has been modified to 1) use more realistic default values and 2) allow more realistic extrapolation of the aquifer influence functions (*AQMETHOD *CARTER-TRACY).

A compatibility option *AQMETHOD *OLD has been included in this release to allow the users to use the same *AQPROP defaulting as IMEX 97.00 and to use the same influence function extrapolation as IMEX 97.00.

*SECTORARRAY

It is now possible, in array input format, to define block which are in sectors. This allows a streamlined input of sector information.

*SCONNECT

The *SCONNECT keyword can set up a flow connection between any two reservoir blocks. Transmissibility is defined explicitly on the *SCONNECT card.

*CORNER-TOL

The tolerance setting how close to each other corner point blocks have to be before they are considered to be in contact is now user defined.

Component Properties

IMEX 98.00 includes significant additions to the Component Property section.

The phase appearance/disappearance algorithms have been rewritten from scratch to properly account for the disappearance of multiple phases within a grid block (e.g. water and gas within a solvent filled block).

An API tracking option was developed (jointly with Petrobras) to model reservoirs with simple compositional gradients.

A gas saturation threshold was added to the minimum solvent saturation to control onset/loss of miscibility in the four component miscible model.

Both the solvent model's gas saturation threshold and minimum solvent saturation have had smoothing applied to the cutoff values, this causes a smoother transition from miscible to immiscible conditions (and the reverse) as well as improved numerical performance.

Previous restrictions which limited the *OILWATER option to use a single PVT region have been removed. *GORINT is no longer valid when used with the *OILWATER option.

*MINSS min_sol_sat (*SGTHRESH *sgval*)
 (*SMOOTHEND (*ON | OFF))

This keyword controls onset/loss of miscibility at low solvent saturations (min_sol_sat) and low gas saturations (sgthresh). The discontinuity in fluid properties this causes, by default, is smoothed to improve convergence.

*API-INT or *API-INTOW

These keywords signal IMEX to use the API tracking model with free gas (*API-INT) or without free gas (*API-INTOW).

*PVTAPI

This keyword is used to input a gas PVT table (for use with the API tracking option).

*APIGRAD

This keyword is used to input an oil PVT table for each oil density (gravity) specified within a PVT region (for use with the API tracking option).

*GORINT

*GORINT is no longer used with the *MODEL *OILWATER option.
*GORINT continues to be used with the *MISCG model (miscible model with chase gas injection).

Numerical Methods

*NORM *PBUB *MAXCHANGE *PBUB

The *NORM *PB and *MAXCHANGE *PB have been changed to *NORM *PBUB and *MAXCHANGE PBUB for compatibility with CMG's GRIDBUILDER. Please alter data to reflect this change.

Initialization

The Initialization section has changes which reflect the new API tracking option and the *SWINIT option.

*API *APIT

The API tracking option allows for two methods to specify the initial light oil volume fraction in the reservoir. API can either be defined as a grid array or as a function of depth for each PVT region.

*SWINIT

The *SWINIT option allows the user to specify a water saturation and still go through the *BLOCK_CENTER gravity capillary equilibrium option.

*SWINIT can be different from the connate saturation. Equilibrium is achieved by scaling the individual blocks P_{cow} curve to achieve equilibrium at a specified water saturation.

*GOC_SW *WOC_SW

These keywords allow the user to explicitly define the water saturation above the GOC and below the WOC for each PVT region. These values override the normal defaults assigned when using the *BLOCK_CENTER gravity capillary initialization options and the values assigned using *SWINIT.

Well and Recurrent Data

*RANGE and *REFINE

The capability to define new refined grids in the Recurrent Data section has been added to IMEX. Thus the keywords *RANGE and *REFINE may appear in the well (recurrent) data. Please refer to the Reservoir Description section for information on the usage of these cards.

*PERFRG

The ability to perforate in multilevel refined blocks has been added to the *PERFRG keyword.

The *PERFRG card now can handle perfs which exist at in layers at different levels of refinement. For example the following is now possible.

	Fundamental			Refine1 (5x5x1)			Refine2 (3x3x1)			Well Index
	i	j	k	i1	j1	k1	i2	j2	k2	
*PERFRG	2	2	2							1.0
	2	2	3	3	3	1				1.0
	2	2	4	3	3	1	2	2	1	1.0

In addition, wells that are accidentally perforated in pinched out or nulled out blocks are now flagged in a more informative manner. A list of valid perfs (with i,j,k locations) is presented to the user, along with a separate list for invalid perfs. It is now a simple process to produce a valid perf list.

*WELLINIT

The *WELLINIT keyword can now also be defined on a well by well basis. In addition to the previous usage,

*WELLINIT (*ITER | *CHANGE | *Timestep)

the format,

*WELLINIT *well_list* (*ITER | *CHANGE | *Timestep)
can be used.

*ALTER, *TARGET, *GUIDEI, *GUIDEP, *GLIFT, *ON-TIME, *ALTERCP

These keywords listed allow the user a simpler method for assigning all values in a well list to one value. Previous versions of IMEX allowed the user to employ well lists with wildcarding, but values (rates, etc.) assigned to each well had to be entered explicitly (even if one value is used for all wells).

Example

```
*ALTER 'PRODNW'*  
14*1000.0           <----- Input List (14 entries)
```

Thus the number of wells in the wildcard list had to be known, making input more time consuming.

If the user now specifies a single value in the list input IMEX will now set all wells in the list to this value. This applies to all types of well lists (not just those using wildcards).

```
*ALTER 'PRODNW'*  
1000.0           <----- single entry, all wells  
                   in list alter primary  
                   constraint to 1000.0
```

It is not necessary to know the number of wells in the well list to apply the single value to the list.

OLD OPTIONS NO LONGER SUPPORTED BY IMEX 98.00

Input/Output Control

*DIM

The IMEX 98.00 model no longer supports the *DIM *MEMORY_MODEL keyword. The run time dimensioning now makes this backward compatibility feature unnecessary.

Important Changes Between IMEX 97.00 and IMEX 96.00

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 97.00

Input/Output Control

***DIM**

The IMEX 97.00 black-oil model is run time dimensioned. That is, the simulator allocates memory that is required at run time. The multiple copies of simulators for the users specific RAM requirements are no longer necessary. In addition special versions are now not required, as any number of wells and blocks are handled.

In most cases, the action taken by run time dimensioning will be transparent to the user and no extra dimensioning control will be required. However, in the case of complex models with pinched out layers, the run time dimensioner may require additional user guidance in order to minimize storage allocated.

Due to this need we have introduced the *DIM keyword which enables the user to override automatic settings and calculations and to explicitly specify model dimensions.

***INTERRUPT, *WRST_COLD, *STOP_COLD**

The interrupt keyword now has two extra options. The first allows the user to write an unscheduled restart record and continue on with the run. The second is only for PC workstations and immediately terminates the run.

***TNEXT**

The *WRST keyword and the *WPRN (*WELL *GRID *SECTOR) keywords all now allow the use of the *TNEXT keyword. The *TNEXT keyword specifies that the simulator output action (e.g. write a restart) will occur on the date (time) specified on the next *DATE (*TIME) card only.

***OUTPRN, *GRID, *SWC, *SORW, *SGC, *SORG**

The critical/connate/residual saturations can now be respecified in the Recurrent Data section. These variables have been added to the grid output arrays and the SR2 output via the *OUTSRF *GRID card.

***OUTPRN, *RES, *POR**

The porosity printout in the initial condition output no longer includes the effects of net-to-gross ratio. The porosity printed out is the gross porosity.

***FORGAS**

The *FORGAS card is no longer required to use the IMEX/FORGAS coupled model. FORGAS or IMEX-only specification is made interactively at the beginning of a FORGAS/IMEX run.

***DEBUG, *XFLOW, *PARAMS**

IMEX 97.00 incorporates a fully coupled wellbore crossflow model. THE *DEBUG *XFLOW keyword enables the user to obtain detailed information on how the crossflow model is operating. The *DEBUG *PARAMS keyword enables the user to view the integers which determine the 'Virtual' common block dimensions of the run time dimensioned IMEX.

Reservoir Description

The Grid Module within IMEX has been significantly improved this year. Of particular interest to the users of refined grids is a large reduction in CPU time required to initialize large refined regions. The initialization of large refined regions of blocks has been sped up by as much as ten times.

***TRANLI, *TRANSLJ, *TRANSLK**

IMEX 97.00 now allows the user to specify interblock transmissibility on all faces (not just a single face in each direction). This feature allows the user to more easily specify interblock flow between refined and unrefined regions.

Earlier versions of IMEX did not properly handle the use of aquifers connected to specified regions. This has been corrected in IMEX 97.00.

Component Properties

A positive total hydrocarbon compressibility check has been added to the PVT table reader. This check ensures physically meaningful results when gas comes out of solution.

Therefore $B_g \cdot dR_s / dP - dB_o / dP > 0.0$ must be true, or in terms of table entries i and $i+1$:

$$B_g(i+1) \cdot (R_s(i+1) - R_s(i)) / (P(i+1) - P(i)) > (B_o(i+1) - B_o(i)) / (P(i+1) - P(i))$$

A warning is printed if this check is violated. However the run will continue.

Rock-Fluid Properties

New checks on S_{wc} , S_{orw} , S_{gc} and S_{org} have been included to minimize possible user error.

***NOSWC**

The *NOSWC option has been completely rewritten to more accurately account for fluid relative permeabilities measured without connate water present.

***KROIL, *SEGREGATED**

A fourth three-phase oil relative permeability option has been included.

***RTYPE, *SWC, *SGC, *SORW, SORG**

The rock type (*RTYPE) and connate/critical/residual saturations can now be entered in the Recurrent Data section. This feature can be used to model hysteresis effects.

Numerical Methods

The AIMSOL solver has been improved to handle more efficiently options which employ higher order factorizations (*SDEGREE 2). The CPU time penalty for using a more accurate factorization has been substantially reduced.

*NORM *PB, *MAXCHANGE *PB

It is now possible to enter separate desired changes (*NORM) and maximum changes (*MAXCHANGE) for bubble point pressure. If not input the bubble point pressure change will default to the pressure change.

*AIM *STAB *ALL-BLOCKS, *AIM *STAB *AND-THRESH

Two new *AIM *STAB subkeywords have been added to allow the user more control over the stability based explicit-implicit block switching algorithm. Normally only neighbours of implicit blocks are checked for stability (C.F.L.) violations.

The *ALL-BLOCKS option checks every explicit block. This requires a significant amount of extra CPU time and may increase run times by as much as ten percent.

The *AND-THRESH option adds nothing to the CPU time and in addition to the stability checking performs threshold checking on all blocks. This increases the rate at which explicit blocks turn implicit but also may improve run timestepping behavior.

Well and Recurrent Data

MULTILATERAL WELL OPTION

*FLOW-FROM, *FLOW-TO, *CLOSED, *REFLAYER

It is now possible to specify well trajectories for multilateral wells under the *PERF or *PERFRG keywords. For each perforation, following the location, relative permeability and status information, the optional keyword *FLOW-TO for producers or *FLOW-FROM for injectors can be used to assign "parent-child" relationships between perforation layers. This information serves to define the well trajectory. In addition to open or automatic status designation for a perforation, a third choice of closed is introduced with the keyword *CLOSED. Unlike the two previous choices a "perforation" with a closed status has no flow associated. Finally the keyword *REFLAYER may be used to designate the reference layer for bottomhole calculations. Please see the pages for *PERF and *PERFRG for more information.

WILDCARDS IN WELL LISTS

A significant new feature has been added to all keywords which acts upon well lists (e.g. *ALTER 'well1' 'well2'). Wild cards have been incorporated in the following manner:

* replaces any number of characters at the end of a well name or can be used on its own to represent all wells (e.g. *ALTER '*' or *ALTER 'wel*').

? replaces any single character anywhere in the well name (e.g. *ALTER '?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.

E.g. if the wells in the model are 'well1' 'well2' and 'prod1', then, '*' would list all 3, 'p*' would list 'prod1'. '?ell?' would list 'well1' and 'well2'.

CROSSFLOW MODEL

*XFLOW-MODEL 'well list' *SOURCE-SINK
 *ZERO-FLOW
 *FULLY-MIXED

A fully featured implicit crossflow model has been added into IMEX 97.00. This model when used in *FULLY-MIXED mode allows the user to model crossflow in a well. *ZERO-FLOW allows the user to shut in backflowing layers (the default), while *SOURCE-SINK reverts the model to the IMEX 96.00 default which does not properly handle crossflow or backflow.

MISC OPTIONS

*HEAD-METHOD 'well list' *GRAVITY
 *ZERO-HEAD

In flowing wells it is appropriate to calculate the density of the fluids in the wellbore based on mobilities of fluids in the well blocks. However for pumping wells, except for a small amount of liquid collected at the bottom of the well, the head in the well is effectively zero (the head of gas). In this case we offer the user the *ZERO-HEAD option. The default *GRAVITY option calculates wellhead as was previously done in IMEX 96.00.

*OPERATE *PENALTY *WGR
 *NODAMP
 *DAMP
 *PHASE2

The ability to use water-gas ratio as a penalty has been added along with different methods for calculating the penalty. The *PHASE2 most accurately honors the penalty rate but the *NODAMP and *DAMP methods may more accurately mimic field operations.

MOBILITY WEIGHTED WELL PRESSURES

IMEX will print out mobility weighted well pressure, mobility weighted datum pressure and mobility weighted drawdown to aid in history matching. In order to obtain this report, the user must request well output at reservoir conditions. See *OUTPRN *WELL *RESERVOIR'.

The mobility weighted datum pressure is written to the SR2 file system, if detailed layer output or block pressure is specified for output. See '*OUTSRF *WELL'

AUTODRILLING OF WELLS

In the 96.00 release the automatic opening of wells possessing the *AUTODRILL status was triggered as an action taken when a group minimum oil or minimum gas constraint was violated. This had the unfortunate effect of preempting the use, in these situations, of minimum oil or gas monitors to terminate the simulation. In the 97.00 version, the opening of wells with *AUTODRILL status is triggered when *AUTODRILL *ON is specified for a group or groups using the *GAPPOR (group apportionment) keyword and a group listed fails to meet a target during the target apportionment. The *AUTO-OPEN action for the *GCONM keyword is no longer supported; this keyword structure must be converted by the user to the *GAPPOR form. Please see the discussion of the *GAPPOR, *AUTODRILL, and *DRILLQ keywords for more information.

GUIDE RATE APPORTIONMENT

In earlier releases of IMEX when guide rates were specified to determine target rate apportionment, production targets other than oil rates were not properly handled. This has been corrected in version 97.00; guide rate apportionment is done correctly for all allowed target rates.

TRANSFER OF CUMULATIVES IN HIERARCHY REDEFINITION

In earlier releases if a group structure was introduced after a model had been run for some time with no group structure defined, the FIELD cumulative was reset to zero when the FIELD became part of the group hierarchy. In version 97.00, when the group structure is redefined, the newly defined top-level group acquires the cumulative production of the old top-level group.

IDENTIFICATION OF REFERENCE LAYER AFTER REPERFORATION

In earlier releases when a multilayer was reperforated in such a way that some layers were reperforated, some eliminated, and some new layers were added, the reference layer (the one in which the well's bottomhole pressure is defined) was chosen as the layer first perforated historically, regardless of its position in the current set of perforated layers. For example, if a well were first perforated in layers 1, 3, and 4, in that order, and later reperforated in layers 2, 3, and 4, earlier releases would pick layer 3 as the reference layer. Version 97.00 always picks the first layer perforated in the most recent set of perforation lines as the reference layer; in the example, version 97.00 would pick layer 2 as the reference layer. Please see the discussion of the *PERF keyword for more information.

OLD OPTIONS NO LONGER SUPPORTED BY IMEX 97.00

*MONITOR *BACKFLOW *AUTOLAYER for injectors is no longer valid. Use the default *XFLOW-MODEL *ZERO-FLOW to accomplish this function.

In addition, we recommend that *MONITOR *BACKFLOW *AUTOLAYER not be used for producers, instead the XFLOW-MODEL should be used.

The *AUTO-OPEN action for the *MINOIL and *MINGAS monitors under the *GCONM keyword is no longer supported; this keyword structure must be converted by the user to the *GAPPOR form. Please see the discussion of the *GAPPOR, *AUTODRILL, and *DRILLQ keywords for more information.

Important Changes Between IMEX 96.00 and IMEX 95.00

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 96.00

Input/Output Control

IMEX 96.00 has several enhancements that make the input output controls more flexible and user friendly. Display of the screen diary file can be changed to include additional well information. IMEX can now be run with the name of the data set specified through a command line argument (i.e. mx064.exe -f data1.dat). The root names of the output file, input .irf and .mrf files will automatically be the root name of the data file.

***OUTSRF, *WELL, *DOWNHOLE, *LAYER, *BLOCKP**

Well layer records are no longer written to the SR2 files *WELL at surface and reservoir conditions. In order to reduce the size of .mrf files, only well records at surface conditions are written. The user can still write layer information at either reservoir and/or surface conditions using the keywords shown here.

***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 SR2 is in XDR format, then the keyword "XDR" will appear near the top of the index file (IRF). XDR is a nonstandard option, and the source code is shipped with XDR disabled. Most executables shipped by CMG will have it enabled.

***OUTDIARY, *WELLSTATUS**

These keywords specify that the number of producing wells, injecting wells, and shut in wells and the maximum changes of saturations and pressure will be written to the output-diary-file. The oil/gas/water production/injection rates will be written with up to 6 significant figures. The water-cut will be written with up to 3 significant figures.

***INTERRUPT, *RESTART-STOP, *STOP, *INTERACTIVE**

These keywords indicate that when the user interrupts the simulation run by simultaneously pressing the "CTRL-C" keys or by typing "kill -2" followed by the process identification number (PID), one of the following user specified actions can take place:

1. Continue the simulation run.
2. Finish the current timestep, write a restart record and stop the simulation.
3. Flush buffers, close files and stop the simulation run immediately.

Use of this keyword prevents corruption of the .irf and .mrf files when aborting a simulation run and optionally writes a restart record before stopping.

Reservoir Description

*PVCUTOFF

This option allows the user to adjust the pore volume cut off value and controls the level at which a block will be set null due to a small pore volume. This option ensures that blocks with small pore volumes can be systematically removed from the simulation. Such small pore volume blocks can hinder convergence and should not remain in a simulation.

*NETGROSS

This option allows the input of net-to-gross ratios 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.

Rock-Fluid Properties

A number of options for smoothing both user input relative permeability and capillary pressure data. In addition, these data can also be generated using standard power law correlations.

*SMOOTHEND, *QUAD, *CUBIC, *POWERQ, *POWERC

This optional keyword indicates the type of smoothing that is to be used at the end points for the interval between $k_r = 0$ and the first nonzero value of k_r for both k_{rw} and k_{row} (k_{rg} and k_{rog}). The keyword may be followed by subkeywords that specify the type of smoothing to use.

For the subkeyword *QUAD, a quadratic smoothing is used. For *CUBIC, a cubic smoothing is used. If no smoothing has been specified, then linear interpolation is used. For *POWERQ and *POWERC end point smoothing is done using power law if the exponent is between 1.5 and 4.0. Otherwise, for *POWERQ, quadratic is used and for *POWERC, cubic is used for end point smoothing.

Please see the *SWT or *SGT (*SLT) entry in this manual for more details.

*SMOOTH, *ALL, *ALLINT, *ALLIN, *ALLPL

This option smoothes the entire relative permeability table and generates a new table consisting of smoothed, equally spaced values. One or all saturation dependent properties in the table may be smoothed based on user controlled specifications. The data may be smoothed using a power law correlation, linear interpolation or a combination of the two. Please see the *SWT or *SGT (*SLT) entry in this manual for more details.

Numerical Methods Control

*RELAX, *ACC, *GAS

The *RELAX *GAS enables the gas-saturation under-relaxation option. This option can lead to faster convergence for simulations where the overall pressure in the reservoir is declining and gas is being liberated due to the pressure falling below the bubble point. This option prevents the gas from

disappearing in a grid block for three Newtonian iterations. If IMEX predicts that gas in a grid block will disappear for more than three iterations, only then will the gas be allowed to disappear.

The *RELAX *ACC option performs over-relaxation of the accumulation term. This normally improves model performance.

Well Management

Well management has several improvements and bug fixes. Some new options have also been added.

*ON-TIME

This new option allows the input of fractions for all wells or selected wells which specify the fraction of the simulated time during which the well is on-line and actually in operation. If a well is assigned an on-time fraction less than one, the rates reported (the "averaged" rates) are less than the rates which apply when the well is actually in operation (the "instantaneous" rates). The user should enter well constraints (such as BHP, WHP and rate constraints) which correspond to INSTANTANEOUS rates; reported BHP and WHP values will correspond to the instantaneous rates. Reported rates and cumulative volumes will, however, be the averaged values. If the *ON-TIME keyword does not appear in the data set, the default value of one is applied for the on-time fraction and the simulation is unaffected by the introduction of the option.

*MONITOR, *MIN or *MAX, *WHP or *BHP

It is now possible to enter maximum BHP and WHP values as monitored constraints for producing wells; please see the entry for the *MONITOR keyword in this manual for more details.

It is also now possible to specify maximum lift gas rates for wells under the *GLIFT *OPT option; please see the *GLIFT entry for details.

When entering wellbore hydraulics tables under the *PTUBE keyword, it is now possible to direct the simulator to supply missing values by interpolation or extrapolation from other values in the same row of the table by entering *EXTP in the place which the missing number would normally occupy; please see the *PTUBE entry in this manual for more details.

*MINOIL, *MINGAS, *AUTO-OPEN

An *AUTO-OPEN action has been added to the *MINGAS and *MINOIL constraints under the *GCONM group monitoring keyword. This action allows wells in a particular group which have *AUTODRILL status to be opened automatically when the group's gas or oil production rate falls below the specified value. Please see the *GCONM entry in this manual for more details.

***TURB, *QUAD**

This option is used for calculating the drawdown pressure for high gas rate wells. These keywords specify the use of quadratic inflow performance and a rate dependent turbulent skin factor. Please see the *PERF entry in manual for more details.

***WELLINIT, *CHANGE, *ITER**

This option allows the user to specify whether the bottomhole pressure value for all wells running on a rate constraint, drawdown constraint or an implicitly imposed wellhead pressure constraint should be reinitialized 1) only after well changes or 2) after each Newtonian iteration. This keyword also determines whether wells operating under a *WHP *IMPLICIT constraint are checked for shutting in at the Newtonian or timestep level.

Use of the *ITER option, especially for high permeability cases, usually results in rapid convergence of the well equations by reducing the total number of Newtonian iterations. For cases with low permeability and/or high compressibility, this option may reduce the total number of Newtonian iterations and reduce the CPU time.

OLD OPTIONS NO LONGER SUPPORTED BY IMEX 95.00

The *UNDER-RELAX *GAS option is no longer supported by IMEX 95.00. Use the *RELAX *GAS option instead.

Important Changes Between IMEX 95.00 and IMEX 94.00

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 95.00

Reservoir Description

IMEX 95.00 now allows corner point input. Several different methods for entering corner point data are allowed.

***GRID, *CORNER**

This option allows the entry of corner point grids. Corner point grids are made up of blocks defined by their eight corner points. Each corner point is described by giving its x, y and z coordinate values, which gives its location in the reservoir.

***COORD**

This options causes the reading of information defining the "x"- and "y"- coordinate locations for all corner points defining a *CORNER grid. Since the corner points must lie on vertical lines, there being exactly $(n_i+1) \times (n_j+1)$ such lines, and since definition of a line requires the specification of two points, each requiring the specification of three coordinates, *COORD expects to read $2 \times 3 \times (n_i+1) \times (n_j+1)$ values.

***XCORN**

This option will cause the reading of all the $(n_i+1) \times (n_j+1) \times (n_k+1)$ "x"- coordinate values required to define the lateral locations of all points defining a *CORNER grid.

***YCORN**

This option will cause the reading of all the $(n_i+1) \times (n_j+1) \times (n_k+1)$ "y"- coordinate values required to define the lateral locations of all points defining a *CORNER grid.

***ZCORN**

This option causes the reading of all depths ("z"-coordinates) of all the corner points required to define the grid.

***CORNERS**

This options causes the processing of $3 \times (8 \times n_i \times n_j \times n_k)$ values, with the first group of $8 \times n_i \times n_j \times n_k$ values giving all the "x"-coordinates of all corner points, the second group giving all the "y"-coordinates, and the third group giving all the "z"-coordinates.

Rock-Fluid Properties

***KROIL**

This option allows the specification of different methods for the evaluation and determination of K_{ro} , the three-phase oil relative permeability. These methods are Stone's first model; Stone's second model as modified by Aziz and Settari; and the linear isoperm method.

***STONE1**

This specifies the use of Stone's first model for computing K_{ro} . The K_{row} and K_{rog} values may be looked up as functions of either S_w and S_g or as functions of $1-S_o$ and $1-S_{wc}-S_o$.

***STONE2**

This specifies the use of Stone's second model for computing K_{ro} . The K_{row} and K_{rog} values may be looked up as functions of either S_w and S_g or as functions of $1-S_o$ and $1-S_{wc}-S_o$.

***LINEAR_ISOPERM**

This specifies the use of the LINEAR ISOPERM model as proposed by L.E. Baker. The method allows the specification of straight-line segments along which K_{ro} assumes a constant value.

Well Management

***GLIFT, *RATE, *GLR, *OPT**

Gas lift rates can now be specified in three ways:

1. Input lift gas rates directly using the *GLIFT *OPT *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 time-step's production data. However, the lift gas rates will not be used for the calculation of production rates until the next timestep.
$$\text{Lift_gas_inj} = \text{GLR_input} \times \text{Total_liquid_est} - \text{GOR_est} \times \text{Oil_rate_est}$$
3. Gas injection rates will be calculated automatically using the values specified in the *GLOPT keyword.

These rates can be modified at specified dates and times in the Well and Recurrent Data section.

***GLCONTROL**

This option allows the specification of gas lift constraints for a given set of production wells. The specified operating constraints are checked at every timestep. If one of the constraint is violated, then the well or group of production wells will be put on gas lift. The operating constraints that can be specified are: wellhead pressure, bottomhole pressure or stock tank oil rate and maximum water-cut. A hydraulics table can also be specified, which is used for wellbore hydraulics calculations when the well or wells are switched

to gas lift. The operating constraints can be changed at specified times or dates in the Well and Recurrent Data section.

***GLOPT, *MAXGAS, *RECYCLE, *GLOPT-TIME**

This option will automatically optimize gas lift injection rates to provide maximum oil production rates for the specified wells. The total lift gas available can either be input directly using the *MAXGAS keyword or indirectly from the field's gas production rate through the *RECYCLE keyword. In either case, a portion of the available lift gas can be specified for compressor consumption. This consumption rate is proportional to the total lift gas injected. An iterative procedure is used to determine the optimum lift gas allocations to each well. A minimum gas-cost factor, below which gas lift injection uneconomic, can also be specified. This can be used to ensure that large volumes of lift gas are not allocated for small incremental oil volumes. The values specified with this option can be modified at dates and times specified in the Well and Recurrent Data section.

OLD OPTIONS NO LONGER SUPPORTED BY IMEX 95.00

The *WELLDPS option and the *DP_PSEUDO option are not supported by IMEX 95.00.

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 94.00

Fluid Properties

***DIFLIB**

This option allows for the input of differential-liberation tables. These tables are converted by IMEX to PVT tables (corrected for separator conditions).

***LIGHTOIL**

Use a black-oil model, modelling flow of oil, water, and gas. Separate B_o , R_s and two parameters alpha and beta are used to convert stock tank rates to reservoir rates. Alpha is the gas shrinkage factor and beta is the liquid drop out from the gas at stock-tank conditions. This option is useful for those fluids which behave like an oil at reservoir conditions, but are so light that at surface conditions liquids drop out of the gas. This option uses the method outlined in Rovere, A., Sabathier, J.C. and Codreanu, D.B., "Modification of a Black-Oil Model for Simulation of Volatile Oil Reservoirs".

***PVTLO**

This keyword indicates PVT input for the light oil option.

Initial Conditions

The keywords for initial conditions specifications have been completely re-written. However IMEX will accept all the old keywords as before. A depth averaged (*DEPTH_AVE) initialization option has also been added. A description of all the initialization options can be found in Section 8 under the heading Notes on Initial Conditions.

Well Management

The group-well structure can now be changed at any time. Wells can now be moved from one group to another.

Dimensioner

The IMEX 94.00 dimensioner has been improved to dimension the common block tightly.

Important Changes Between IMEX 93.00 and 92.00

WELL GROUP CONTROL MODULE ADDED TO IMEX 93.00

IMEX 93.00 has the ability to specify group control for wells in a fairly comprehensive manner. The following features are now available in IMEX.

1. Up to three group levels can be specified.
2. Groups can have target rates. GOR and water-cuts can be monitored and appropriate actions can be taken.
3. Wells can be opened (drilled) automatically to meet target constraints. This drilling can be done automatically based on instantaneous production/injection potentials or can be done in a user specified sequence.
4. Well layers can be shut in and opened automatically depending on the current GOR and water-cut.
5. Wells can be shut in and opened automatically depending on the current GOR and water-cut.
6. In previous versions of IMEX, voidage replacement was based on surface rates of producing wells. It is now based on production at reservoir conditions and thus gives much more accurate results.
7. In addition to surface rates, bottomhole rates can now be specified as well constraints.
8. *GLIFT -- This keyword allows for the specification of lift gas rates for producing wells.

LIMITATIONS OF WELL GROUP CONTROL

This version of well management has the following limitations:

1. A maximum of three group levels are allowed.
2. The top most group level ('FIELD') cannot have wells attached to it. Only other groups.
3. Groups to which wells are attached cannot have other groups attached to it.
4. New wells may be attached to a group at any time, however a well cannot be attached to more than one group. Furthermore a well cannot be moved from one group to another.
5. Group controlled injection fluids are limited to gas, water or solvent only. Oil injection cannot be controlled by groups.
6. 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/close layers. Thus if the well is perforated horizontally, the behavior of this option may be unpredictable.

DATA INCOMPATIBILITIES BETWEEN 93.00 AND PREVIOUS VERSIONS OF IMEX

Although most of the well data are compatible with previous versions of IMEX, there are some incompatibilities.

1. For the pseudo-miscible option (solvent injection) the previous versions of IMEX monitored the gas+solvent-oil ratio when *MONITOR *GOR was used. In this version of IMEX *MONITOR *GOR refers to gas-oil ratio only. If it is desired to monitor the gas+solvent-oil ratio the *MONITOR *GSOR should be used.
2. Minimum rate operating constraints have been replaced by minimum rate monitoring constraints. If a minimum rate well constraint is specified, it is converted automatically to a monitoring constraint and a warning message is printed.
3. Well names now have to be unique i.e. it is no longer possible to have two or more wells with the same name. This is because wells can now be referred to by their names only. Well numbers are optional.
4. *PERFV can no longer be used to perforate wells located in hybrid grids. The *PERFVHY or *PERFRG keywords should be used to perforate wells located in hybrid grid blocks.
5. The *GATHER keyword and the related keywords *PTARG and *ITARG are now obsolete and cannot be used. The *GROUP and *GCONP, *GCONI and *GCONM keywords must be used instead.
6. Subkeywords *SRFIN, *SRFOUT, *RESTARTIN and *RESTARTOUT under the *FILENAMES keyword are now obsolete and will not be recognized. They have been replaced with the SR2 file system keywords: *INDEX-OUT, *MAIN-RESULTS-OUT, *REWIND-OUT, *INDEX-IN, *MAIN-RESULTS-IN and *REWIND-IN.

NEW KEYWORDS AND OPTIONS ADDED TO IMEX 93.00

Input/Output Control

Water-gas ratio is now printed for gas-water reservoirs instead of water-cut.

***MODSI**

This option allows for the input/output of pressure in Kg/cm².

***SECTOR**

The *SECTOR subkeyword in the *WPRN and *WSRF keywords now controls the sector and lease line printing frequency to the output and SR2 files.

***OUTDIARY**

Printing to the diary-file (normally the screen) can now be controlled by the use of *OUTDIARY keyword.

Reservoir Description

SECTOR OUPUT

The sector output has been revised and considerably enhanced. This option now prints a table of sectors with aquifer influx, and production/injection from each sector, in addition to all the previous quantities.

***AQLEAK**

In previous versions of IMEX, the water OUTFLOW from the reservoir into the aquifer was not modelled when the *AQUIFER (analytical aquifer model) was used. If the reservoir pressure climbs above the aquifer pressure, this implementation can give incorrect results. Thus an option (*AQLEAK) was added to model water outflow into the aquifer when the reservoir pressure rises.

Fluid Properties

***COT**

This keyword allows the user to specify oil compressibility as a function of bubble point pressure and block pressure above the bubble point pressure.

Initial Conditions

***GASCAP**

In previous versions of IMEX, when *VERTICAL *ON option is used, the gas-cap is initialized not to have any oil EVEN if the immobile liquid saturation from the gas-liquid relative-permeability table is GREATER than the connate water saturation from the water-oil relative-permeability table. An option (*GASCAP *OIL) is now available to initialize the gas-cap with residual oil.

The SR2 File System

SR2 FILE SYSTEM

The output for graphics and restarts is now handled in a completely different way. The graphics and restart information is now output to a set of up to three files.

The new file system called the SR2 file system combines the graphics and restart information for greater space efficiency and smaller output files during run time.

The SR2 file system consists of two or three files. These are the index-results-file (IRF), the main-results-file (MRF) and the rewritable-results-file (RRF). The RRF file exists only if the simulator is directed to rewind the restart record. In such a situation, only the IRF and the MRF files will exist.

The two or three files belonging to the SR2 file system work together. Thus an IRF will be useless without a MRF and vice-versa. The IRF is an ASCII file which is an index to the MRF and the RRF i.e. the IRF is a user readable ASCII file which details the contents of the binary MRF and RRF files. The MRF and RRF files are both binary files and hence cannot be read by the user. However a summary of the contents of the MRF and RRF files can be obtained by examining the IRF.

SRF FILE

Since the SR2 file system is more flexible and space efficient, the specifications of the SR2 are completely different to those of the SRF. For those user's who rely on the SRF for their graphics and other output processing, a SR2 to SRF converter is provided with this release. Using this converter any set of SR2 files can be converted to a SRF file.

STRUCTURE OF RESTART RECORD

Due to the new SR2 file system, the writing of the restart record has been completely changed. Thus it is not possible to use restart records generated by previous versions of IMEX.

Introduction to IMEX

IMEX is a three-phase black-oil simulator with gravity and capillary terms. Grid systems may be Cartesian, cylindrical, or variable depth/variable thickness. Two dimensional and three dimensional configurations are possible with any of these grid systems. Gas phase appearance/disappearance is handled by variable substitution.

Some of the novel features of IMEX are:

Adaptive Implicit Formulation

IMEX 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 by the explicit method. The adaptive implicit option accomplishes this and is useful for coning problems where high flow rates occur near the wellbore, or in stratified reservoirs with very thin layers.

By using the adaptive implicit option, a savings of one third to one half of the execution time may occur because timesteps are as large as those obtained using the fully implicit method. Options are available so that a user can select a fixed pattern of fully implicit blocks. IMEX also selects these blocks dynamically, based on user specified thresholds or on matrix switching criteria.

Dual Porosity/Dual Permeability

The dual porosity option allows the discretization of matrix blocks in either a nested format called the multiple-interacting continua (MINC) approach or a layered format called the subdomain method. The dual porosity approach idealizes the fractured reservoir as consisting of two parts: the primary porosity and the secondary porosity.

The primary porosity (the matrix) represents the small intergranular pores in the rock matrix. The secondary porosity (the fractures) consists of fractures, joints, and/or vugs. The dual porosity approach is characterized by representing one reservoir volume with two continua. The fractures, having small storativities, are the primary conduits of fluid flow, whereas the rock matrices have low fluid conductivities but larger storativities.

A simple dual porosity model using a one matrix/one fracture system may be specified. Shape factors based on the work of Warren and Root, or Gilman and Kazemi may be used. In this case the matrix-fracture transfer is assumed to be under semi-steady state flow.

To achieve adequate representation of the matrix-fracture transfer mechanism, it may be necessary to subdivide the matrix volume into smaller volume elements. The MINC approach achieves this by subdividing the matrix into a series of nested volume elements. The discretization is based on the assumptions that the elemental surfaces are equipotential surfaces, and that the variation of the primary variables across one fracture block is small.

The MINC model is a good representation of transient matrix-fracture flow. Gravity effects, however, are not handled properly which may become a serious disadvantage when the matrix block height is large. A model which improves the representation of gravity effects and fluid phase segregation is the subdomain method. In this model, fluid and pressure distribution in the matrix are calculated, but gravity segregation is assumed to exist in the fracture.

Besides the MINC model and the subdomain model, the dual permeability model allows fluid communication between the matrices. This option is useful when fracture connectivities are oriented in one preferred direction, or when matrix-matrix connectivity is important, such as in gas-oil gravity drainage processes.

Pseudo-Miscible Option

A pseudo-miscible option models first-contact and multiple-contact miscible processes. It can model solvent in the water phase and also chase gas.

Polymer Option

A polymer option models rock adsorption, polymer inaccessible pore volume, resistance factors, viscosity mixing, and dispersion.

API Tracking

An API tracking option models reservoirs with compositional gradients. Oil is divided into two components. Oil properties are functions of composition as well as pressure and bubble point pressure.

Faulted Reservoir Option

A fault option models stratified reservoirs with one or more faults. This option accurately accounts for flows between offset layers and enables simulation to be performed without unrealistic averaging of properties. Sloping faults which are nonparallel to the co-ordinate axes are also modelled.

Fully Implicit Wells

Wells are solved in a very robust fashion. The bottomhole pressure and the block variables for the blocks where the well is 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. Also, a comprehensive well control facility is available. An extensive list of constraints (maximum, minimum bottomhole or wellhead pressures, rates, GOR, etc.) can be entered. As a constraint is violated, a new constraint can be selected according to the user's specifications.

Matrix Solution Method

IMEX uses a state-of-the-art solution routine based on incomplete Gaussian elimination as a preconditioning step to GMRES acceleration. AIMSOL has been developed especially for adaptive implicit Jacobian matrices.

For most applications the defaults control values selected by IMEX will enable AIMSOL to perform efficiently. Thus, IMEX users do not require detailed knowledge of matrix solution methods.

Local Grid

A facility for local grid refinement is included in IMEX. The user specifies a region of the reservoir which is to be subdivided. All interblock connections and transmissibilities are automatically calculated by IMEX. This option can be used to study near well effects in field-scale simulation and eliminates the need for well pseudo-functions. Static fractures can

also be efficiently modelled with this technique. All extra terms are correctly handled by the matrix solution routine.

Pinched Out Layers

Pinch outs are efficiently modelled by using the pinch out option. This option removes pinched out blocks from the active block list, and automatically couples together layers above and below the pinch out.

Reservoir Initialization

Initial reservoir conditions can be established with given gas-oil and oil-water contacts.

Flexible Grid System

Several grid options are available with IMEX: Cartesian coordinates, cylindrical coordinates, variable thickness/variable depth and corner point grids. Two dimensional and three dimensional systems are possible with any one of these options.

Variable Bubble Point

A rigorous variable bubble point formulation using variable substitution is used.

Undersaturated regions, and mixing of oils with different PVT data are properly modelled.

Aquifer Models

Aquifers are modelled by either adding boundary cells which contain only water or by the use of the analytic aquifer model proposed by either Carter and Tracy or Fetkovitch. The first method is useful in the situation where the aquifer dimensions and location are well known and its inclusion in the reservoir can be achieved by a relatively small number of additional blocks. The later methods are more useful for large to infinite aquifers where an approximate calculation of water influx into the reservoir is desired, but their representation through the addition of boundary reservoir blocks is not feasible.

Input/Output Units

SI, field, laboratory or modified SI units can be specified.

Portability

IMEX is written in standard FORTRAN 77 and has been run on various hardware platforms. These include: IBM main frames, CDC, CRAY, Honeywell, DEC, Prime, Gould, Unisys, Apollo, SUN, HP, FPS and IBM-PC compatible 386 and 486, Pentium, Pentium Pro, and Pentium II computers.

Graphics

CMG's graphics system RESULTS, uses the SR2 file system for post-processing of simulation output.

RESULTS can also be used for input data preparation and grid design.

Run Time Dimensioning

IMEX is written to automatically dimension itself to your problem size. The input file is pre-scanned so that the model dimensioning parameters can be found. IMEX then assigns storage based on the scanned model parameters.

Crossflow in the Well

IMEX can use an advanced fully mixed crossflow model to describe well crossflow/backflow phenomena.

Condensate Modelling

IMEX can be used to model gas reservoirs with condensate. Oil is permitted to exist in the gas phase in the *GASWATER_WITH_CONDENSATE option (the gas phase must always exist and the oil phase must initially not be present). The *VOLATILE_OIL model can model the most general case allowing both saturated and undersaturated oil and gas. Any of the oil, gas, and water phases may initially be present in the reservoir.

Parallelized Jacobian Building and Matrix Solver

The Jacobian Building and Solver have been rewritten to allow IMEX to run in parallel on some platforms. Please refer to the Numerical Method Control section of the user manual for more information.

Wellbore Friction Model

A wellbore friction model has been introduced within IMEX. This includes several powerful frictional pressure drop correlations and is robust enough to handle complex scenarios (such as multilateral wells).

Seawater Injection/Scale Buildup Model

The tracking of seawater injection within the reservoir has been added and coupled to a scale deposition/well productivity damage model. This allows IMEX to model the affects of seawater and formation water mixing when calculating scale deposition at producing wells.

Grid Amalgamation

IMEX can statically amalgamate, re-refine and de-refine grid blocks

Gas Desorption

Langmuir Isotherm parameters can be input to model the adsorption/desorption of gas to/from rock. This feature can be used to model simple adsorption effects in shale gas and CBM problems.

Tutorial

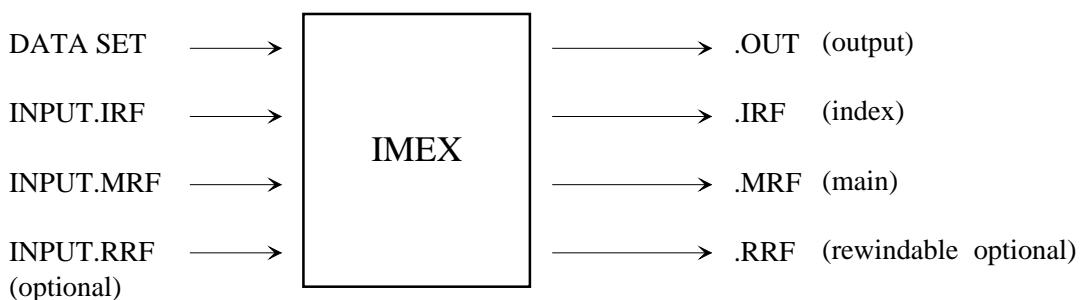
Introduction

The Tutorial section is a guide for the novice user of the keyword input system and does not replace the User Guide. 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.

IMEX uses the data set that you create initially and then creates three or four other files. Each IMEX run creates an output-file (OUT), an index-results-file (IRF) and a main-results-file (MRF). In addition a rewritable-results-file (RRF) may or may not be created depending on the options selected by the user.



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:
 - Input/Output Control
 - Reservoir Description
 - Component Properties
 - Rock-Fluid Data
 - Initial Conditions
 - Numerical Methods Control
 - Well 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 and Recurrent 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 SR2 file system, 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 ARE RESTART FILES?

Restart files are a set of SR2 files. These include at least an index-results-file (IRF) and a main-results-file (MRF). The set may or may not include a rewritable-restart-file. Please review keyword *RESTART_SR2 for information about separating and combining graphics and restart data. Restart information (usually called a restart record) about the simulation variables is written to these files at the frequency chosen by the user. Restarts can be done from only those timesteps at which a restart record is written. Writing a restart record 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 run, you may select a timestep 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 timestep you have chosen.

HOW TO DO A RESTART

Restart records are optional and do not have to be written. But if you do plan on doing restarts, you need to create one with your initial run.

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

*WRST indicates the frequency of writing to the restart record. *RESTART 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. This is the default. Otherwise, enter a timestep number.

Example:

```
*RESTART 30
*WRST    10
```

To do a restart run:

- a) Do not change any of the original reservoir data, but do add *RESTART 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 IRF file, an input MRF file and possibly an input RRF file.
- d) You will also need names for the new set of output files. Be sure that all the input files were created from the same simulation run.

Controlling Contents of the Output File

To control the contents of the output 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 and Recurrent Data section.

*WPRN indicates how often to write grid block data, sector & lease plane 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  
*WPRN *SECTOR  0
```

If either of these is left out of the data set, then the default is to print the *GRID and *SECTOR information at every *TIME or *DATE keyword in the Well and Recurrent Data section. For *WELL the default is to print information at every timestep. These defaults can produce a very large output file, which can fill up the available disk space on your computer very quickly.

*OUTPRN limits what well data, grid data, reservoir data, and how many property tables are printed. You may actually list the grid data types that you want.

Well data is treated differently. You may print out everything possible or print a well summary only. To print out information per layer for all wells use *OUTPRN *WELL *LAYER. To print out production at reservoir conditions use *OUTPRN *WELL *RESERVOIR. To print out a one line summary for each well use *OUTPRN *WELL *BRIEF. The latter is the default.

The use of the *TNEXT subkeyword of *WPRN is an efficient method of specifying output on a specific date. *TNEXT turns on printing at the next date card, then resets itself to not print after that date.

Controlling Contents of the Graphics File (SR2)

To control the contents of the SR2 file system, use:

- a) *WSRF, and
- b) *OUTSRF

These keywords may appear in the Input/Output Control section or the parameters may be changed later in the Well and Recurrent Data section of the input data file.

*WSRF indicates how often to write grid block, sector and well data. This information may be written at a certain frequency.

If no grid or well information is desired in the SR2 files, 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 every timestep. *GRID and *SECTOR are printed when *TIME is found in the Well and Recurrent Data section. The amount of information can be controlled by the *OUTSRF keyword.

The keyword combination *WSRF *GRID 0 ensures that grid information is never written to the simulation results file. This can be changed in the well data at subsequent well changes.

*OUTSRF limits what grid data are printed. You may also ask to have special variables printed at given grid block locations. Separate lists of variables are available for grid information and well information.

The use of the *TNEXT subkeyword of *WSRF is an efficient method of specifying output on a specific date. *TNEXT turns on SR2 writing at the next date card, then resets itself to not print after that date.

Describing Your Grid System

To describe your grid system, you need:

- a) *GRID
- b) *KDIR
- c) *DI
- d) *DJ
- e) *DK

Use either

- f) *DEPTH

or

- g) *PAYDEPTH

or

- h) *DTOP, and
- i) *DIP

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 and must appear in the data set before the *NULL and *POR keywords.

*GRID describes the type of grid system that is being used. There are 4 choices: regular Cartesian, variable depth/variable thickness, and radial-angular cylindrical and corner point grid. 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) 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 for a coning study. It is 10x1x15. The last sets up a corner point grid.

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

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
```

where 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 blocks 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. Please note that your data starts with the bottommost layer when using *KDIR *UP.

Describing Refined Grid

To describe the location of 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 and must appear in the data set before the keywords *NULL and *POR.

*REFINE requires the number of refined blocks the fundamental grid blocks will be split up into, in 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, 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 keyword looks like this:

Example:

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

You are allowed to split up a fundamental block into and including 4 refined blocks. 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). In some cases areas with different degrees of refinement must be separated by at least one fundamental (unrefined) grid block. Also note that grid refinement is not allowed when the dual-porosity option is being used.

The variable thickness, variable depth option may be used with refined grid. However, the thicknesses of individual refined blocks are assumed to be equal within each individual fundamental grid block.

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 inherited.

Using Dual Porosity/Dual Permeability

The simulation of naturally fractured reservoirs can be conducted through the use of dual porosity models. In these models, the reservoir is discretized into two collocated continua (two sets of grid blocks located in the same space), one called the matrix and the other called the fracture. The matrix continuum is assumed to be comprised of matrix blocks which are separated spatially by fractures. The dimensions of these matrix blocks can be variable throughout the reservoir and are a function of the fracture spacing, orientation and width. Within the computational grid, the matrix-fracture transfer is represented by a single flow term. The flow of fluids through the reservoir occurs through the fracture system, which is the primary conduit for fluid flow. Wells in a fractured block are always connected to the fracture.

IMEX contains four different types of dual-porosity models that can be used. When using any of the dual porosity options, both matrix and fracture properties must be specified. This is done by using the *MATRIX or *FRACTURE modifier after the main keyword.

Example: Specify matrix and fracture porosities and permeabilities. Matrix porosity is 0.10 and fracture porosity is 0.05. Matrix permeability is 50.0 md areally with $kv/kh = 0.10$. Fracture permeability is 500 md areally with $kv/kh = 0.10$.

In the Reservoir Description section enter the following:

```
*POR *MATRIX      *CON    0.10
*POR *FRACTURE   *CON    0.05
*PERMI *MATRIX    *CON    50.0
*PERMJ *MATRIX    *CON    50.0
*PERMK *MATRIX    *CON    5.0
*PERMI *FRACTURE *CON  500.0
*PERMJ *FRACTURE *CON  500.0
*PERMK *FRACTURE *CON  50.0
```

In a similar manner, other properties such as assigning relative permeability tables to matrix and fracture blocks or specifying blocks as IMPES or fully implicit can be accomplished.

Example: Initially set matrix blocks to IMPES and fracture blocks to fully implicit.

In the Well and Recurrent Data section after the first *DATE keyword, enter the following:

```
*AIMSET *MATRIX    *CON  0
*AIMSET *FRACTURE *CON  1
```

In addition to specifying the type of dual porosity option to use, the method of calculating the shape factor must also be specified. IMEX has two methods available for calculating the shape factor. They are based either on the Warren and Root model or the Gilman-Kazemi model. Please refer to the appropriate section of the User Guide for further reading on the two methods for calculating shape factor. To invoke the Warren and Root method, specify *SHAPE *WR after specifying the type of dual porosity option to use in the Reservoir Description section of the data set. Similarly, to invoke the Gilman-Kazemi method, specify *SHAPE *GK after specifying the type of dual porosity option to use in the Reservoir Description section of the data set.

In order to calculate shape factors, IMEX requires information on the geometry of the fractures. This is done by specifying the fracture spacing in the three principal directions (I, J and K directions). The fracture spacing can vary spatially throughout the reservoir. Sections of the reservoir can be dual porosity and sections can be set to single porosity by setting the fracture porosity to zero. Specifying the fracture spacing in the three principal directions is done by using the *DIFRAC, *DJFRAC and *DKFRAC keywords in the Reservoir Description section of the data set. Note that specifying a very large fracture spacing indicates that the shape factor, sigma, will be small and hence the matrix fracture transfer will be reduced. Specifying small values of fracture spacing indicates the block contains many fractures and consequently, the matrix-fracture transfer will be increased. Specifying a fracture spacing of zero indicates that there are no fracture planes perpendicular to those axes. Note that the fracture spacing affects only the matrix-fracture transfer term and not the number of connections between matrix and fracture. In order to specify a region of the reservoir to be single porosity, it is necessary to set the grid block fracture porosity to zero. Note that in order to null out matrix or fracture blocks, the porosities must be set to zero. It is not possible to use:

```
*NULL    *MATRIX      *CON 0
-OR-
*NULL    *FRACTURE   *CON 0
```

This is because *NULL applies to spatial grid blocks (in this case both matrix and fracture blocks) and not to matrix and fracture blocks separately (active blocks).

Example: A reservoir model consists of a 10x10x5 grid. All layers are fractured with fracture spacings of 50 ft in the I, J directions and 10 ft in the K direction.

In the Reservoir Description section specify:

```
*DIFRAC  *CON  50.0
*DJFRAC  *CON  50.0
*DKFRAC  *CON  10.0
```

Layer 3 and 4 have been found to contain no fractures (they can be represented as a single porosity system). Modify the data accordingly. Assume that the fracture porosity is 0.05.

```
*POR  FRACTURE  *IJK
1:10  1:10  1:5  0.05
1:10  1:10  3.4  0.0
```

The four different dual porosity options are described below. Only one type of dual porosity model can be specified in a data set.

a) Standard Dual Porosity

Fluid flow through the reservoir takes place through the fracture network. The matrix blocks essentially act as source and sink terms. This model is invoked by specifying the keyword *DUALPOR in the Reservoir Description section.

Example: Use standard dual porosity model with the Gilman-Kazemi method of calculating the shape factor.

In the Reservoir Description section of the data set, enter the following:

```
*DUALPOR
*SHAPE  *GK
```

b) Dual Permeability

This model differs from the standard dual porosity model in that each matrix block is connected to both the fracture blocks and the surrounding matrix blocks. Now, fluid flow occurs through both the fracture network as well as through the matrix blocks. The dual permeability option can be important for cases where there is capillary continuity. For these cases, dual permeability is often important for reservoirs with free gas, oil and large variations in depth. Usually, only the vertical k direction matrix-matrix transfer is important. The dual permeability option is invoked by specifying *DUALPERM in the Reservoir Description section. Matrix-matrix flow can be reduced or set to zero through the use of matrix transmissibility multipliers.

Example: Use the dual permeability model with the Gilman-Kazemi method of calculating the shape factor. Also set matrix-matrix transfer in vertical direction only.

In the Reservoir Description section of the data set, enter the following:

```
*DUALPERM  
*SHAPE      *GK  
*TRANSI     *MATRIX   *CON   0.0  
*TRANSJ     *MATRIX   *CON   0.0  
*TRANSK     *MATRIX   *CON   1.0
```

c) Subdomain

This model is a variation of the standard dual porosity option. It allows the user to refine the matrix blocks in the vertical direction in order to more accurately represent the gravity drainage process from matrix to fracture which is essentially vertical. Each matrix block is refined by the number of subdivisions specified in order to more accurately represent the fluid pressures and saturations within the matrix blocks. This resolution is not needed in the fracture, therefore only the matrix blocks are refined. The subdomain option is invoked by specifying *SUBDOMAIN ndiv in the Reservoir Description section, where ndiv is the number of refined layers for each matrix block. Note that the value for ndiv is constant throughout the reservoir.

Example: Modify the dual porosity example case above to subdomain. Assume the number of vertical subdivisions are 3.

In the Reservoir Description section:

```
*SUBDOMAIN 3  
*SHAPE      *WR
```

Assume the reservoir model consists of a 10x10x5 grid. All layers are fractured with fracture spacings of 50 ft in the I, J directions and 10 ft in the K direction.

Add the following in the Reservoir Description section:

```
*DIFRAC    *IJK  
1:10  1:10  1:5  50.0  
*DJFRAC   *IJK  
1:10  1:10  1:5  50.0  
*DKFRAC   *IJK  
1:10  1:10  1:5  10.0
```

d) MINC

Another extension of the standard dual porosity concept is the multiple interacting continua (MINC) approach proposed by Pruess and Narasimhan. The main advantage of MINC is the representation of the matrix-fracture transfer calculation using a nested discretization of the matrix blocks. This allows a very efficient representation of the transient fluid regime, which is often neglected, in the standard dual porosity model. The nested discretization is one-dimensional and can represent the pressure, viscous and capillary forces. The gravity force however is not considered in this nested, one-dimensional matrix refinement. The MINC option is invoked by specifying *MINC ndiv in the Reservoir Description section, where ndiv is the number of nested layers for each matrix block. Note that the value for ndiv is constant throughout the reservoir.

Example: Modify the dual porosity example case above to subdomain. Assume the number of nested subdivisions is 3.

In the Reservoir Description section:

```
*MINC 3  
*SHAPE *WR
```

Assume the reservoir model consists of a 10x10x5 grid. All layers are fractured with fracture spacings of 50 ft in the I, J directions and 10 ft in the K direction.

Add the following in the Reservoir Description section:

```
*DIFRAC *IJK  
1:10 1:10 1:5 50.0  
*DJFRAC *IJK  
1:10 1:10 1:5 50.0  
*DKFRAC *IJK  
1:10 1:10 1:5 10.0
```

Summary: To invoke the dual porosity/dual permeability options you may use:

- a) *DUALPOR
- b) *MINC
- c) *SUBDOMAIN
- d) *DUALPERM
- e) *SHAPE
- f) *DIFRAC
- g) *DJFRAC, and
- h) *DKFRAC

These keywords must appear in the Reservoir Description section. For the different options that are available, only one may be used in any given data set.

If any of these are used, locate them before *NULL and *POR.

Input of Null Blocks

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

- a) *NULL and
- b) *POR

Both must appear in the Reservoir Description section. *NULL is optional, while *POR must appear in your data set, in any case. *POR is the keyword used for specifying porosities.

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.

You may use the *IJK array reading option 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 the designation in *POR. If a nonzero porosity is assigned to a block that has already been designated a null block by *NULL, then the *POR assignment is ignored.

Null blocks using *POR are indicated by a zero porosity, while active blocks are assigned nonzero values.

The aforementioned example may also be represented by *POR:

Example:

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

DUAL POROSITY/DUAL PERMEABILITY CASE

In the case of a dual porosity/dual permeability model, the input of porosity values requires input CASE for the matrix and the fracture. Data for the matrix must be entered first and then the data for the fracture. This procedure is similarly expected for other data.

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.

In dual porosity, null blocks imply that both the matrix and the fracture have zero porosity. In general, either the matrix porosity or the fracture porosity may be set to zero and the other nonzero.

Using the Aquifer Option (see Appendix G)

The equations that are used in IMEX to model an aquifer are given in Appendix G and in Carter, R.D. and Tracy, G.W., "An Improved Method for Calculating Water Influx," Trans., AIME, vol. 219, (1960), pp. 415-417. These equations are an approximation to the equations of Everdingen and Hurst. Please refer to van Everdingen, A.F. and W. Hurst, "The Application of the Laplace Transform to Flow Problems in Reservoirs," AIME, Dec. 1949, pp. 305-324B and Chapter 9 in "Fundamentals of Reservoir Engineering" by L.P. Dake.

In order to use the aquifer option the following keywords could be used:

- a) *AQUIFER
- b) *AQPROP
- c) *AQMETHOD
- d) *AQLEAK
- e) *AQFUNC

These keywords are expected in the Reservoir Description section and are completely optional. If these options are used, they must be placed after *NULL and *POR.

The keyword *AQUIFER is required to activate the aquifer option. The location of the aquifer is specified through the subkeywords *BOUNDARY *BOTTOM and *REGION.

*AQUIFER *REGION i1:i2 j1:j2 k1:k2 *IDIR will indicate that the aquifer is connected to the grid blocks specified by the i1:i2 j1:j2 k1:k3. *IDIR specifies that the aquifer is connected to faces perpendicular to the I direction. For a grid block i,j,k the aquifer is connected to the face bordering this grid block and grid block i+1,j,k .

*AQUIFER *BOUNDARY is used to connect the aquifer to all the grid blocks at the boundary of the reservoir.

*AQUIFER *BOTTOM is used to connect the aquifer to all grid blocks at the bottom of the reservoir.

The properties of the aquifer can be specified using the keyword *AQPROP followed by the values of the aquifer thickness, porosity, permeability, radius, and angle and the ratio of the aquifer's external radius to the reservoir's effective radius (Fetkovich aquifer only).

*AQPROP thickness porosity permeability radius angle (R-ratio)

The simulator calculates default values for the thickness, radius and angle. In order to use these default values please input 0.0 at the appropriate location.

Carter and Tracy's equations require the usage of a dimensionless pressure influence function $P(t_d)$ as a function of the dimensionless time t_d . This function can be specified using the keyword *AQFUNC. The default dimensionless pressure function used is the one given by van Everdingen and Hurst for a constant terminal rate solution and an infinite radial aquifer. Influence functions for limited extent aquifers can also be found in the same reference.

Please note that Carter and Tracy's formulation requires the usage of a dimensionless pressure influence function $P(t_d)$ instead of the dimensionless rate influence function $Q(t_d)$ that are normally used to calculate flux rates for terminal pressure problems.

If a Fetkovitch aquifer 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, pp. 814-828). This approach is able to model finite aquifers without having to generate dimensionless pressure functions. Only a "R-ratio" parameter is required.

The *AQLEAK option allows for the analytical aquifer model to have negative influx or outflow from the reservoir when the reservoir pressure exceeds the aquifer pressure. Since the outflow from the reservoir is modelled analytically it is possible to have water outflow from a grid block attached to the aquifer that has no mobile water. Thus great care should be exercised when using *AQLEAK *ON. It is strongly recommended that aquifers be modelled using grid blocks, for data sets where the reservoir pressure is expected to increase significantly during the course of the simulation.

Using the Pseudo-Miscible Option

Define the pseudo-miscible model by using either of *MODEL *MISCG, which denotes a pseudo-miscible model, including chase gas or *MODEL *MISNCG, which denotes a pseudo-miscible model, not including chase gas. These keywords must appear in the Component Properties section of your data set.

In the *MISNCG model, the gas phase in the reservoir is solution gas. It may be in solution, as well as at or below the bubble point. If gas is injected into the reservoir in addition to solvent or water, it is assumed to be the reservoir gas.

Model *MISCG under certain conditions allows the modelling of injected chase gas. In the *MISCG model, the gas phase in the reservoir is an injected gas (chase gas), it is assumed that the reservoir is at a GOR defined by *GORINT initially. The gas injected in the reservoir is not the reservoir gas. The initial Rs obtained from the PVT table (through bubble point pressure) must be zero (as there is no chase gas in the reservoir initially), GORINT is added to this value. There must also be no free gas in the model initially (as this would be chase gas).

The gas PVT properties entered in the PVT table are that of the chase gas. Oil properties in the PVT table must also reflect chase gas in solution, not reservoir gas in solution.

The following keywords are additional required keywords to be included in your data set when you use the pseudo-miscible option.

You need:

- a) *PVTS (a single table for all PVT regions, without a PVT region number specified, or one table per PVT region, with PVT region numbers specified)
- b) *DENSITY *SOLVENT (a single keyword for all PVT regions or one keyword per PVT region)
- c) *OMEGASG (a single keyword for all PVT regions or one keyword per PVT region)
- d) *MINSS
- e) *GORINT (*MODEL *MISCG only)

These keywords must be located in the Component Properties section.

- f) *SWT, where capillary pressure and the irreducible oil saturation is required input. *SWT must appear in the Rock-Fluid Data section of your data set.
- g) *PBS; this keyword must appear in the Initial Conditions section of your data set.
- h) *INCOMP *SOLVENT *GLOBAL
- i) *OPERATE *SOLVENT, and
- j) *ALTER, perhaps, to change the solvent rate at subsequent well changes. h), i) and j) must appear in the Well and Recurrent Data section of your data set.

Using the Rock Compaction/Dilation Model Option

The following keywords are additional required keywords to be included in your data set when you use the rock compaction model option. These keywords must be located in the Reservoir Description section (see the keywords *CROCKTYPE, *CROCKTAB, *CROCKTABH, *CCPOR, *CCPRPOR, *CIRREVERS, and *CTYPE for information).

It is always necessary to define both *CPOR and *PRPOR when the analytical aquifer model is used. The aquifer compressibility is not defined by any of the options available on the keywords following *CROCKTYPE, only *CPOR defines aquifer compressibility.

When a region of the reservoir is not explicitly assigned a value of *CTYPE while other regions are assigned *CTYPE values, it is assumed the unassigned region's compressibility and reference pressure are defined by the *CPOR and *CCPRPOR keywords.

Please note that the *INT" option used with input tables is not operational with the “*CROCKTAB” and “*CROCKTABH” tables.

You need:

- a) For the case of constant rock compressibility which varies by *CTYPE region
 - *CROCKTYPE
 - *CCPOR
 - *CCPRPOR

For example:

```
*CROCKTYPE 1           (Note: for rock region 1,  
*CCPOR    3.0e-6      fracture properties defaulted to  
*CCPRPOR   14.7      matrix properties)  
*CROCKTYPE 2  
*CCPOR    *MATRIX     3.0e-6  
*CCPRPOR   *MATRIX     14.7  
*CCPOR    *FRACTURE   5.0e-6  
*CCPRPOR   *FRACTURE  14.7
```

or

- b) For the case of reversible nonconstant rock compressibility or the case when permeability varies with pressure. (This option can only be applied on single porosity, dual porosity and dual permeability models).

```
*CROCKTYPE  
*CROCKTAB
```

For example:

```
*CROCKTYPE 1  
*CROCKTAB  
**  
**pressure  porosity multiplier      permeability  
          multiplier (hor.)  
1000       1.002956            1.014445  
2000       1.005956            1.029286  
3000       1.008956            1.044309  
4000       1.011956            1.059518
```

The above porosity multipliers are computed by using the following equation:

$$\text{porosity multiplier} = 1 + c_{\text{por}} * (p - p_{\text{prpor}}) \quad (1)$$

where:

$$\begin{aligned}c_{\text{por}} &= 3.0e-6 \text{ 1/psi} \\p_{\text{prpor}} &= 14.7 \text{ psi}\end{aligned}$$

p is pressure on the table.

and permeability multipliers are determined by the modified equation (2) (Espinoza, C.E. "A New Formulation for Numerical Simulation of Compaction, Sensitivity Studies for Steam Injection", SPE 12246, 1983.):

$$\begin{aligned}\text{perm.multiplier} &= [(\text{por.mult})^{**m}] \\&\left[((1 / \text{por_input} - 1) / (1 / \text{por_input} - \text{por.mult}))^{**2} \right] \quad (2)\end{aligned}$$

where:

$$\begin{aligned}\text{por.mult} &= \text{porosity multiplier} \\ \text{por_input} &= \text{initial porosity which equals 0.3 in the examples}\end{aligned}$$

m is an adjustable parameter that depends on the rock type. Assuming its value to be 4 in the examples.

In the example above because separate horizontal and vertical permeability multipliers were not entered, vertical permeability multipliers default to horizontal permeability multipliers.

or

- c) For the case of irreversible rock compressibility and permeability multipliers. (This option can only be applied on single porosity, dual porosity and dual permeability models).

*CROCKTYPE
*CROCKTAB
*CIRREVERS

For example:

```
*CROCKTYPE 1
*CROCKTAB
*CIRREVERS
**
**pressure    porosity multiplier      permeability
              multiplier (hor.)
1000          1.002956                1.014445
2000          1.005956                1.029286
3000          1.008956                1.044309
4000          1.011956                1.059518
```

or

- d) For the case of rock compressibility and permeability multipliers which undergo hysteresis. (This option can only be applied on single porosity, dual porosity and dual permeability models).

```
*CROCKTYPE
*CROCKTAB
*CROCKTABH
```

For example:

```
*CROCKTYPE 1
*CROCKTAB
**
**pressure    porosity multiplier      permeability
              multiplier (hor.)
1000          1.002956            1.014445
2000          1.005956            1.029286
3000          1.008956            1.044309
4000          1.011956            1.059518
*CROCKTABH
1000          1.002956            1.014445
2000          1.004962            1.024348
3000          1.006968            1.034333
4000          1.008974            1.044400
*CROCKTABH
2000          1.005956            1.029286
3000          1.007968            1.039341
4000          1.009980            1.049480
*CROCKTABH
3000          1.008956            1.044309
4000          1.010974            1.054519
```

The above porosity multipliers on hysteresis branches are also computed on the basis of equation (3) as below:

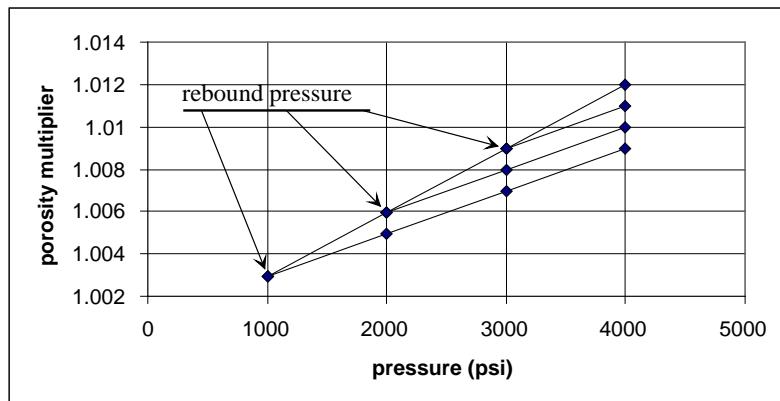
$$\text{porosity multiplier} = [1 + c_{\text{por}} * (p - p_{\text{por}})] * [1 + c_{\text{por}'} * (p - P_r)] \quad (3)$$

where:

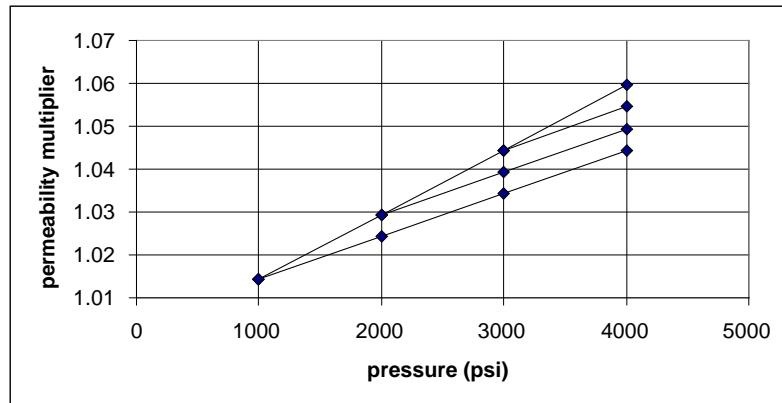
$$\begin{aligned}c_{\text{por}} &= 3.0e-6 \text{ 1/psi} \\p_{\text{por}} &= 14.7 \text{ psi} \\c_{\text{por}'} &= 2.0e-6 \text{ 1/psi}\end{aligned}$$

P_r is rebound pressure (first pressure on a hysteresis branch)
 P is pressure on the table

A graph of hysteresis branches based on the above data is shown below.



and



Note that the first point in all the hysteresis tables must match exactly a point in the main table. Failure to ensure this is true could cause numerical difficulties.

When the *CTROCKTAB table is quite nonlinear it is important to define a sufficient number of hysteresis branches using *CROCKTABH. If the *CROCKTABH data is sparse it is possible that there will be a porosity discontinuity at the rebound pressure. This can be minimized by increasing the number of *CROCKTABH tables and can be eliminated entirely by defining a *CROCKTABH table for every pressure point in the main table.

- e) When more than one rock type are used

*CTYPE

For example: A 10x10x4 grid blocks reservoir consists of 4 layers and each layer has one rock type except the first layer which has two rock types. Distribution of rock types on grid blocks of the reservoir is given as follows assuming 4 crocktypes:

```
*CTYPE    1:5    1:10    1    1
*CTYPE    6:10   1:10    1    2
*CTYPE    1:10   1:10    2    2
*CTYPE    1:10   1:10    3    3
*CTYPE    1:10   1:10    4    4
```

When the rock dilation model option is used, a number of additional keywords should be included in a data set. Those keywords must be located in the RESERVOIR DESCRIPTION section (see the keywords *CROCKTABE, *CROCKTABD, *CROCKTABU, *CROCKTABC, *CROCKTABR for more details).

It is also noted that those above keywords can not be combined with any other keywords of the rock compaction model (*CROCKTAB, *CROCKTABH and *CIRREVERS) and vice versa for one rock type.

In one reservoir, rock compaction model and rock dilation model can not be used simultaneously. However, if there are more than one *CROCKTYPE (reservoir) in a simulation, each reservoir can use one of the rock models if those *CROCKTYPE's (reservoirs) are not in contact to each other.

Similar to the rock compaction model, it is always necessary to define both *CPOR and *PRPOR when the analytical aquifer model is used.

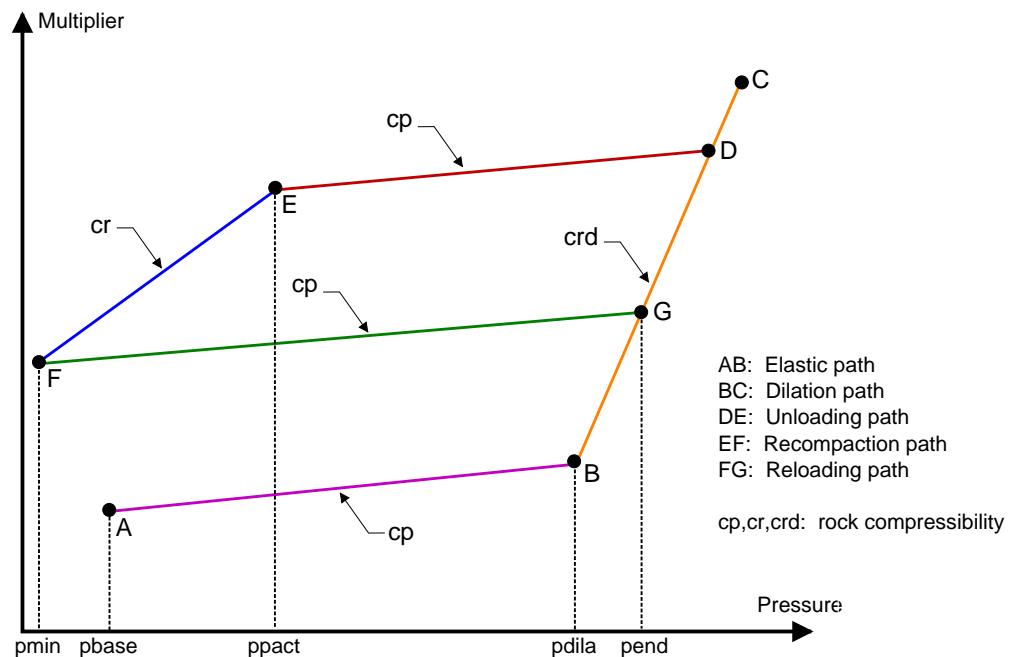
*CROCKTYPE and *CTYPE keywords are also used as in the case of the rock compaction model to define rock type numbers and rock type regions.

A following sample data for rock dilation model that is based on formulae by Beattie, C.I., Boberg, T.C. and McNab, G.S.: "Reservoir Simulation of Cyclic Steam Stimulation in the Cold Lake Oil Sands", SPE Reservoir Engineering, May 1991, pp. 200-206 and STARS User Guide Version 98 is under keywords *CROCKTABE, *CROCKTABD, *CROCKTABU, *CROCKTABC and *CROCKTABR.

Parameters that were used in the formulae to create the data are:

pbase	=	14.7 psi	Initial pressure at which elasticity begins
pdila	=	400.0 psi	Pressure at which dilation begins
ppact	=	200.0 psi	Pressure at which recompaction begins
crd	=	0.0007 (1/psi)	Dilation rock compressibility
cp	=	0.0001 (1/psi)	Elastic rock compressibility
fr	=	0.1	Residual dilation fraction
phi0	=	0.3	Initial porosity
pmin	=	0.001 psi	Minimum pressure that recompaction ends
m	=	1.0	An adjustable parameter used in permeability multiplier formula

Please see the below figure for locations of the above corresponding pressures.



A Reservoir Rock Dilation Model Based on Formulae

Also noted that when using the formulae to create data for rock dilation model, the elastic path, unloading path and reloading path have the same value of rock compressibility.

*CROCKTABLE			
** PRESSURE	POR_MULT	HOR_PERM_MULT	VER_PERM_MULT
14.7000	1.0000000	1.0000000	1.0000000
100.0000	1.0085665	1.0160131	1.0000000
150.0000	1.0136219	1.0255614	1.0000000
200.0000	1.0187027	1.0352320	1.0000000
250.0000	1.0238090	1.0450268	1.0000000
300.0000	1.0289409	1.0549481	1.0000000
350.0000	1.0340985	1.0649980	1.0000000
400.0000	1.0392819	1.0751787	1.0000000

Porosity multipliers in the *CRCOCKTABLE table are computed by a formula:

$$\text{por_mult} = \exp(cp^*(p - pbase)) \quad (1)$$

where:

p (psi) is pressure in the first column of *CROCKTABLE table.

Horizontal permeability multipliers are given by Espinoza, C.E.: "A New Formulation for Numerical Simulation of Compaction, Sensitivity Studies for Steam Injection", SPE 12246, 1983.

$$\begin{aligned} \text{hor_perm_mult} &= [(\text{por_mult})^{**m}] * [((1/\phi_0 - 1) / \\ &(1/\phi_0 - \text{por_mult}))^{**2}] \end{aligned} \quad (2)$$

*CROCKTABD			
** PRESSURE	POR_MULT	HOR_PERM_MULT	VER_PERM_MULT
400.0000	1.0392819	1.0751787	1.0000000
500.0000	1.1146383	1.2327992	1.0000000
600.0000	1.1954587	1.4240450	1.0000000
800.0000	1.3751049	1.9523725	1.0000000
1000.0000	1.5817471	2.8069041	1.0000000
1200.0000	1.8194423	4.3221774	1.0000000
1300.0000	1.9513667	5.5628500	1.0000000
1400.0000	2.0928568	7.4048444	1.0000000
1500.0000	2.2446060	10.3099231	1.0000000

Porosity multipliers in the *CROCKTABD table are given by:

$$\text{por_mult} = \exp(cp^*(pdila-pbase)) * \exp(crd^*(pd(i) - pdila)) \quad (3)$$

where:

pu(i) is pressure at i row in the *CROCKTABU table.

Horizontal permeability multipliers are computed by using the formulae (2).

*CROCKTABU			
** PRESSURE	POR_MULT	HOR_PERM_MULT	VER_PERM_MULT
700.0000	1.2821393	1.6591104	1.0000000
600.0000	1.2693818	1.6223585	1.0000000
500.0000	1.2567512	1.5867360	1.0000000
450.0000	1.2504832	1.5693339	1.0000000
400.0000	1.2442463	1.5521972	1.0000000
300.0000	1.2318659	1.5186989	1.0000000
250.0000	1.2257219	1.5023269	1.0000000
200.0000	1.2196086	1.4861998	1.0000000

Porosity multipliers in this *CROCKTABU table are computed by:

$$\text{por_mult} = \exp(cp * (\text{pdila} - \text{pbase})) * \exp(crd * (p - \text{pdila})) * \exp(cp * (\text{pu}(i) - \text{pu}(1))) \quad (4)$$

where:

$\text{pu}(i)$ is pressure at i row in the *CROCKTABU table.

$\text{pu}(1)$ is the first pressure in the *CROCKTABU table.

In the equation (4), unloading path has the same rock compressibility as that of the elastic path.

Horizontal permeability multipliers are computed similarly by using the formulae (2).

*CROCKTABC			
** PRESSURE	POR_MULT	HOR_PERM_MULT	VER_PERM_MULT
200.0000	1.2196086	1.4861998	1.0000000
150.0003	1.1671184	1.3541457	1.0000000
100.0005	1.1168873	1.2377947	1.0000000
50.0008	1.0688181	1.1347675	1.0000000
0.0010	1.0228177	1.0431193	1.0000000

Porosity multipliers in the *CROCKTABC table are computed by:

$$\text{phii} = \text{phi0} * \exp(-cp * \text{pbase}) + fr * (\text{phi0} * \exp(cp * (\text{pdila} - \text{pbase}))) * (\exp(crd * (\text{pu}(1) - \text{pdila})) - 1) \quad (5)$$

$$\text{cr} = (\ln(\text{phii}) / (\text{phi0} * \exp(cp * (\text{pdila} - \text{pbase}))) * (\exp(crd * \text{pu}(1) - \text{pdila})) - cp * (\text{ppact} - \text{pu}(1))) / \text{ppact} \quad (6)$$

$$\text{por_mult} = \text{phii} * \exp(-\text{cr} * \text{pc}(i)) / \text{phi0} \quad (7)$$

where:

$\text{pc}(i)$ is recompaction pressure which is determined by dividing equally interval pressure (0.001, ppact) by a number of intervals such as 5 as in this example.

Horizontal permeability multipliers were obtained by using equation (2).

*CROCKTABR			
** PRESSURE	POR_MULT	HOR_PERM_MULT	VER_PERM_MULT
0.0010	1.0228177	1.0431193	1.0000000
88.0112	1.0318593	1.0606251	1.0000000
176.0214	1.0409808	1.0785331	1.0000000
264.0316	1.0501829	1.0968556	1.0000000
352.0418	1.0594664	1.1156055	1.0000000
440.0520	1.0688319	1.1347960	1.0000000

In the above table, porosity multiplier is computed as follows:

- i) Pressure at the intersection point between reloading path and dilation path:

Assuming that the reloading path has the same rock compressibility as that of the elastic path. Pressure at the intersection point between reloading path and dilation path is computed as:

$$\text{pend} = (\ln(\text{phi0} * \exp(cp * (\text{pdila} - \text{pbase}))) / \text{por_cend}) + \text{cp} * \text{pc_end} - \text{crd} * \text{pdila}) / (\text{cp} - \text{crd}) \quad (8)$$

where:

pend : pressure at the intersection point
por_cend : porosity at the last pressure pc_end in the *CROCKTABC table
pc_end : last pressure in the *CROCKTABC table (= 0.001 psi as shown above)

Reloading pressure pl(i) in the CROCKTABR table is determined by dividing the pressure interval (0.001,pend) by a number of intervals such as 6 as in the example.

- ii) Porosity multiplier is obtained as:

$$\text{por_mult} = \text{por_cend} \cdot \exp(\text{cp} \cdot (\text{pl}(i) - \text{pc_end})) / \phi_0 \quad (9)$$

In the equation (9), reloading path has the same rock compressibility as that of the elastic path. Horizontal permeability multipliers in the *CROCKTABR table are computed through equation (2).

Using the Polymer Option

Define the polymer option by either *MODEL *POLY which models the flow of oil, water, gas, and polymer, or by *MODEL *POLYOW which denotes a polymer model with no gas flow or no variation in solution gas. These keywords must be located in the Component Properties section.

The following keywords are additional keywords to be included in your data set when you use the polymer option.

You need:

- a) *PADSORP (*MODEL *POLY only)
- b) *PMIX (*MODEL *POLY only)
- c) *PVISC
- d) *PREFCONC
- e) *PPERM

These keywords must be located in the Component Properties section of your data set.

- f) *DISPI
- g) *DISPJ
- h) *DISPK

These must appear in the Rock-Fluid Data section.

- i) *INCOMP *WATER, followed by the initial polymer concentration
- j) *ALTERCP

These keywords must appear in the Well and Recurrent Data section of your data set.

Problems with Small Timesteps or Long Execution Times

When problems of a numerical nature are encountered it is extremely helpful to rerun the simulation with

*WPRN *ITER *MATRIX

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

Convergence failure may result due to:

- a) Inner (linear solver) iteration convergence failure
- b) Newtonian convergence failure resulting in timestep 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 timesteps. This is done by setting a smaller maximum timestep size with *DTMAX or reducing the desired changes per timestep 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 timestep cuts and are caused by maximum changes, in the primary variables, which exceed the specified nominal changes by a factor more than two per timestep. 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-cuts occur, then you can try the following remedies:

4. Check the rock and PVT curves for nonlinearities, discontinuities or inflection points. The curves should be smooth. Also note whether IMEX warned you of negative total compressibility in your PVT table. If negative total compressibility was observed, correct this.
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 and by messages in the output file that gas is on or off in the grid blocks, then smoothing nonlinearities (4) or reducing the timestep size (1) are better solutions.

8. Increase the number of timestep cuts available by using the keyword *NCUTS.
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 neighbor 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, then set the problem regions to implicit.

It is also possible to use the '*AIM *STAB *AND-THRESH' option to check blocks adjacent to wells using stability constraints and all explicit blocks using threshold limits.

10. Ensure that the adaptive implicit method is being used that *AIMWELL *WELLNN is employed. The default for AIMWELL is to only set well blocks implicit.
11. Ensure that reservoir block pore volumes are of a reasonable size. Very small block can cause serious oscillations when connected to large blocks. The use of *PVCUTOFF to null out small blocks can in many cases eliminate this problem.

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. Using 0.1 kPa as the pressure convergence tolerance is recommended for high permeability areas.

Material balance errors can be caused by convergence tolerances being too large compared to the nominal change at each timestep. 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 *PRESS and *NORM *SATUR, and the tolerances, *CONVERGE *PRESS and *CONVERGE *SATUR, are adequate.

However, in cases where you are trying to simulate fractures or horizontal wells, it is best 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 is used) include:

kcyc	Newton cycle counter
dpmx	maximum change in pressure over a timestep
iblk	grid block number

dpbmax	maximum change in bubble point pressure over a timestep
dsomax	maximum change in oil saturation over a timestep
dsgmax	maximum change in gas saturation over a timestep
dswmax	maximum change in water saturation over a timestep
iconv	convergence check counter
nitr	current iteration in subroutine jsolve
nimp	number of implicit blocks
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 * relative tolerance

Using the API Tracking Option

Define the API tracking model by using either of *MODEL *API-INT, which denotes a three phase model, or *MODEL *API-INTOW, which denotes a two phase (oil-water model). These keywords must appear in the Component Properties section of your data set.

The following keywords are additional required keywords to be included in your data set when you use the API tracking option.

You need:

- a) *API or *APIT (INITIAL CONDITIONS)
- b) *PVTAPI (COMPONENT PROPERTIES)
- c) *APIGRAD (COMPONENT PROPERTIES)
- d) *APIGRAV (INPUT / OUTPUT CONTROL)

When using the API tracking option, the oil is assumed to be comprised of two components, a light component and a heavy component.

The volume fraction of light oil at STC is defined as V_l , the volume fraction of heavy oil is $V_h = (1-V_l)$.

The density of pure light oil at STC is denol(STC), and the density of pure heavy oil is denoh(STC). The density denoh(STC) is the density of the oil mixture at $V_l=0.0$. This is the first *APIGRAD table which must be defined. The last *APIGRAD table must be defined for pure light oil ($V_l=1.0$). The value of *APIGRAD for this table is therefore denol(STC).

The density of the mixture deno(STC) is:

$$\text{deno(STC)} = \text{denoh(STC)} * V_h + \text{denol(STC)} * V_l$$

The solution gas ratio of the mixture R_s is:

$$R_s = R_{s(\text{mix})}$$

where $R_{s(\text{mix})}$ is obtained by interpolation of *APIGRAD tables whose oil mixture densities are just lower and just higher than the density defined by the equation for deno(STC) above.

If only two APIGRAD tables are used for a PVT region, this is equivalent to:

$$R_{s(\text{mix})} = R_{sh} * V_h + R_{sl} * V_l$$

where R_{sh} is the R_s of the heavy component and R_{sl} is the R_s of the light component.

Oil mixture formation volume factor oil compressibility (c_o) and viscosity is derived in the same manner.

Using the *API keyword or the *APIT keyword, light oil volume fraction (at STC) is initialized. Using the *PVTAPI keyword a gas PVT property table is input for each PVT region (note: separate gas and liquid tables are used in this option).

The *APIGRAD tables follow the *PVTAPI gas table for each PVT region. Each APIGRAD table is associated with a oil mixture density for a PVT region. Oil mixture densities must be entered from heaviest ($V_l=0$) to lightest ($V_l=1.0$). Multiple PVT Regions may be used with this option.

Problems with Solver Convergence

Larger problems (50,000 blocks plus) or problems with high permeability and low porosity may suffer from poor solver convergence and poor material balance.

A possible solution for this behavior is to increase both the number of orthogonalizations the solver performs and the number of iterations the solver performs.

The default is currently:

```
*NORTH      30  
*ITERMAX    40
```

A possible combination which could be used to improve solver convergence is:

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

It is critical to ensure that the number of orthogonalizations is as large as storage (memory) and CPU time restrictions allow.

The user must increase the internally allocated storage to access *NORTH > 30 (say 40), by adding the following keywords in the Run Time Dimensioning section.

```
*DIM SOLVER_DIMENSIONING 'ON'  
*DIM MAX_ORTHOGONALIZATIONS 40
```

It is critical to ensure that the number of orthogonalizations is as large as storage (memory) and CPU time restrictions allow.

Values of 40 or higher for *NORTH are recommended as long as the extra memory requirements for high values of *NORTH do not access virtual memory during the run, which in turn would slow the model run times significantly.

If increasing the number of orthogonalizations has no effect try increasing the degree of the factorization from one to two (*SDEGREE 1 → *SDEGREE 2).

Another option which has been shown to improve difficult problems, is to increase the solver convergence tolerance using the *PRECC keyword. Increase the solver tolerance to 1.00E-6.

(*PRECC 1.00E-6)

Simulating a Gas-Water or Gas Reservoir

To simulate dry gas in a gas-water or single-phase gas reservoir:

1. Use *MODEL *GASWATER.
2. Enter gas PVT properties using the *PVTG table, do not enter any oil phase properties such as *DENSITY *OIL, *CO, *COT, *CVO, *SO, *PB, or *PBT.

These keywords must be located in the Component Properties section of the data set.

3. Enter only S_w (column 1), K_{rw} (column 2) and optionally P_{cgw} (column 3) in the *SWT table. K_{ro} is not entered.
4. Enter * S_l or * S_g (Column 1) and K_{rg} (column 2) in the *SLT (or *SGT) table. K_{rog} and P_{cog} are not entered.

These keywords must be located in the Rock-Fluid Data section of the data set.

5. Establish the gas-water contact with the following steps:
 - a) Use *VERTICAL *BLOCK_CENTER *WATER_GAS, or
 - b) Use *VERTICAL *DEPTH_AVE *WATER_GAS, if P_{cgw} is zero, or
 - c) USE *VERTICAL *DEPTH_AVE *WATER_GAS *TRANZONE, if P_{cgw} is not zero.
 - d) Set gas-water contact using *DWGC.
 - e) Do not define initial oil saturation, or bubble point pressure.

These keywords must be located in the Initial Conditions section of the data set.

6. Ensure that the irreducible oil saturations S_{oirw} , and $S_{oирg}$ are 0.0 with the following steps:
 - a) Input a water relative permeability table. The first *SWT saturation entry is S_{wcon} , the last *SWT saturation entry is $(1-S_{oirw})$. Ensure the last *SWT entry is 1.0 to ensure S_{oirw} is 0.0.
 - b) In your gas table it is necessary to ensure that $S_{oирg}$ calculated from the table is zero.

There are four possible ways to enter the liquid-gas relative permeability table, and thus four ways to ensure that $S_{oирg}$ is equal to zero. The four are listed below.

1. Using a *SGT table, not using the *NOSWC option: the final gas saturation entry must be $(1-S_{wcon})$.
2. Using a *SLT table, not using the *NOSWC option: the first liquid saturation entry must be S_{wcon} .
3. Using the *SGT table, using the *NOSWC option: the final gas saturation entry must be 1.0.
4. Using the *SLT table, using the *NOSWC option: the first liquid saturation must be zero.

These keywords must be located in the Rock-Fluid Data section of your data set.

Simulating a Gas-Water with Condensate Reservoir

To simulate a wet gas and water reservoir:

1. Use *MODEL *GASWATER_WITH_CONDENSATE.
2. Enter saturated gas and oil phase PVT properties using the *PVTCOND table, do not enter any oil phase properties which are required for pressures other than the bubble point or those which define the bubble point, such as *CO, *COT, *CVO, *PB, or *PBT. Optionally, gas properties for undersaturated gas may be entered using the *EGUST (*BGUST or *ZGUST) and *VGUST tables.

These keywords must be located in the Component Properties section of the data set.

3. Enter S_w (column 1), K_{rw} (column 2), K_{row} (column 3) and optionally liquid-gas capillary pressure P_{cgl} (column 4) in the *SWT table.
4. Enter * S_l or * S_g (Column 1), K_{rg} (column 2), and K_{rog} (column 3) in the *SLT (or *SGT) table. P_{cog} is not entered.
5. P_{cgl} is output in the P_{cog} output arrays for both ascii and SR2 output. The P_{cow} output array is not used.

These keywords must be located in the Rock-Fluid Data section of the data set.

6. Establish the gas-water contact with the following steps:
 - a) Use *VERTICAL *BLOCK_CENTER *WATER_GAS, or
 - b) Use *VERTICAL *DEPTH_AVE *WATER_GAS, if P_{cgl} is zero, or
 - c) Use*VERTICAL *DEPTH_AVE *WATER_GAS *TRANZONE, if P_{cgl} is not zero.
 - d) Set gas-water contact using *DWGC.
 - e) Do not define initial oil saturation, or bubble point pressure.
 - f) Define the initial Dew Point using the *PDEW or *PDEWT keywords.

These keywords must be located in the Initial Conditions Data section of the data set.

7. Enable output options which aid in model analysis with the following steps:
 - a) *OUTPRN/*OUTSRF *GRID *RV *DPP outputs Oil Content and Dew Point Pressure to the output and SR2.
 - b) *PSPLIT *ON adds well production split information to the well output (Output file and SR2 file). The surface oil production is made up of oil from condensate and black oil. The surface gas production is made up of free gas and solution gas.

These keywords must be located in the Input/Output Control section of the data set.

8. Enable numerical control options if necessary:
 - a) *NORM *PDW allows you to control timestep size based on dew point pressure changes.

- b) *MAXCHANGE *PDW allows you to control maximum timestep size allowed based on dew point pressure changes.

These keywords must be located in the Numerical Methods Control section of the data set.

Simulating a Tilted Water-Oil/Water-Gas Contact

To simulate a tilted water contact it is necessary to properly define the grid array “Water Cap Pres Shift/Offset” which corresponds to the IMEX keyword PCOW_SHIFT.

“Water Cap Pres Shift/Offset” can be used to add an offset to each block’s water-oil or water-gas capillary pressure. This is used to set up a stable offset to each block’s water-oil contact.

The PCOW_SHIFT option can only be used only if the *DEPTH_AVE initialization method is in effect.

In order to use this technique to model a tilted contact (1) each PVT region’s WOC_PC must either be defaulted (to zero) or explicitly set to zero and (2) the water-oil /water-gas capillary pressure curves must extend to a water saturation of 1.00.

This tutorial assumes that PVT region averaged estimates of reservoir pressure and bubble point pressure can be used to calculate average PVT properties in a region. The average PVT properties are used to convert between contact heights and capillary pressure offsets.

This tutorial also assumes that the effect of water and oil compressibility on the initial water and oil density can be ignored or accounted for by slightly altering the average value of B_w or B_o .

A digitized WOC map should be opened in BUILDER. This map is then used to calculate the input array “Water Cap Pres Shift/Offset”. The single map should be used to calculate the “Water Cap PresShift/Offset” for all layers.

An estimate of each PVT region’s average pressure $P(\text{est})$ must also be made in order to approximate an average water formation volume factor $B_w(\text{est})$. The water formation volume factor should also approximately account for water compressibility (i.e. $B_w(\text{est}) = B_{wi}^*(1 - C_w^*(P(\text{est}) - P_{rw}))$). Normally the input reference pressure REFPRES in a PVT region is a good estimate of $P(\text{est})$.

If the model being initialized is a Black Oil Model, an estimate of each PVT region’s average bubble point pressure $P_b(\text{est})$ must be made in order to estimate oil properties. Normally the input bubble Pressure table PBT in a PVT region can provide a starting point in determining $P_b(\text{est})$.

If the model being initialized is a Gas Water model, $P(\text{est})$ is also used to estimate the gas formation volume factor $B_g(\text{est})$.

Note: In a Gas Water model the input array is still “Water Cap Pres Shift/Offset” and the IMEX input keyword is still PCOW_SHIFT.

Procedure for Black Oil Models:

1. Open the WOC map you wish to use in BUILDER.
2. Define the model’s PVT regions. Normally a single PVT region would be used. The input requires the user to define a WOC depth (DWOC) which is the PVT region’s water-oil contact. This water-oil contact will be used as a reference value for the shifted water-oil contact calculation (the WOC when PCOW_SHIFT=0.0). The value of the reference WOC must lie between the highest and lowest WOC in the input WOC map to minimize the offset from the reference.

3. Use $P_b(\text{est})$ in each PVT region to determine a value of $R_s(\text{est})$ and $B_o(\text{est})$. $B_o(\text{est})$ may need to approximately account for oil compressibility if the reservoir is significantly above its bubble point. $B_o(\text{est}) = B_o(P_b(\text{est}))*(1-C_o(p(P(\text{est}))-P_b(\text{est})))$.
4. Use $P(\text{est})$ in each PVT region to determine $B_w(\text{est})$. $B_w(\text{est})$ may need to be altered to approximately account for water compressibility if the reservoir pressure is significantly different from P_{rw} . $B_w(\text{est}) = B_{wi}*(1-C_w*(P(\text{est})-P_{rw}))$.
5. Calculate the approximate reservoir density difference between water and oil phases in each PVT region.

$$\text{Den_diff} = \text{Denw(STC)}/B_w(\text{est}) - (\text{Denw(STC)} + R_s(\text{est})*\text{Deng(STC)})/B_o(\text{est})$$

This enables the user to relate a WOC height difference in each block to a pressure difference in each block. If Den_diff changes significantly between PVT regions separate calculations must be done for each PVT region.

6. In BUILDER, specify that the temporary grid array “CMGTemp Prop 1” for a single layer is to be calculated from the WOC map. Copy the input specification to all layers in the model. Calculate the array.
7. In BUILDER, define a formula to subtract the calculated WOC array from the reference WOC for each PVT region from the calculated WOC array and then multiply the result by Den_diff .
 $(\text{Reference WOC} - \text{CMGTemp Prop1})*\text{Den_diff}$
8. Use the formula above to define “Water Cap Pres Shift/Offset”. This will change the WOC map into a capillary pressure offset map. If the reference DWOC or Den_diff is significantly different in different PVT regions, the formula must account for individual PVT region’s unique reference WOC and/or Den_diff .
9. Calculate “Water Cap Pres Shift/Offset”. The array now contains the capillary pressure offset required to maintain the desired tilted WOC.

In Field Units, the conversion constants in the Den_diff equation results in the following equation.

$$\text{Den_diff (psi/ft)} = (\text{Denw(STC)}/B_w(\text{est}) - (\text{Denw(STC)} + R_s(\text{est})*\text{Deng(STC)})/B_o(\text{est}))/144$$

In SI Units,

$$\text{Den_diff (kpa/m)} = (\text{Denw(STC)}/B_w(\text{est}) - (\text{Denw(STC)} + R_s(\text{est})*\text{Deng(STC)})/B_o(\text{est}))*0.009806$$

Procedure for Gas Water Models

1. Open the WGC map you wish to use in BUILDER.
2. Define the model’s PVT regions. Normally a single PVT region would be used. The input requires the user to define a WGC depth (DWGC) which is the PVT region’s water-gas contact. This water-gas contact will be used as a reference value for the shifted water-gas contact calculation (The WGC when PCOW_SHIFT=0.0). The value of the reference WGC must lie between the highest and lowest WGC in the input WGC map to minimize the offset from the reference
3. Use $P(\text{est})$ in each PVT region to determine $B_g(\text{est})$.

4. Use P(est) in each PVT region to determine $B_w(\text{est})$. $B_w(\text{est})$ may need to be altered to approximately account for water compressibility if the reservoir pressure is significantly different from P_{rw} . $B_w(\text{est}) = B_{wi} * (1 - C_w * (P(\text{est}) - P_{rw}))$.

5. Calculate the approximate reservoir density difference between water and gas phases in each PVT region.

$$\text{Den_diff} = \text{Den}_w(\text{STC})/B_w(\text{est}) - \text{Den}_g(\text{STC})/B_g(\text{est})$$

This enables the user to relate a WGC height difference in each block to a pressure difference in each block. If Den_diff changes significantly between PVT regions separate calculations must be done for each PVT region.

6. In BUILDER, specify that the temporary grid array “CMGTemp Prop 1” for a single layer is to be calculated from the WGC map. Copy the input specification to all layers in the model. Calculate the array.

7. In BUILDER, define a formula to subtract the calculated WGC array from the reference WGC for each PVT region from the calculated WGC array and then multiply the result by Den_diff.

$$(\text{Reference WGC} - \text{CMGTemp Prop1}) * \text{Den_diff}$$

8. Use the formula above to define “Water Cap Pres Shift/Offset”. This will change the WGC map into a capillary pressure offset map. If the reference DWGC or Den_diff is significantly different in different PVT regions, the formula must account for individual PVT region’s unique reference WGC and/or Den_diff.

9. Calculate “Water Cap Pres Shift/Offset”. The array now contains the capillary pressure offset required to maintain the desired tilted WGC.

In Field Units, the conversion constants in the Den_diff equation results in the following equation.

$$\text{Den_diff (psi/ft)} = (\text{Den}_w(\text{STC})/B_w(\text{est}) - \text{Den}_g(\text{STC})/B_g(\text{est})/5.615)/144$$

In SI Units,

Den_diff (kpa/m)	$= (\text{Den}_w(\text{STC})/B_w(\text{est}) - \text{Den}_g(\text{STC})/B_g(\text{est})) * 0.009806$
$\text{Den}_w(\text{STC})$	Standard Condition water density (DENSITY WATER keyword)
$\text{Den}_o(\text{STC})$	Standard Condition oil density (DENSITY OIL keyword)
$\text{Den}_g(\text{STC})$	Standard Condition gas density (DENSITY GAS keyword)
$B_w(\text{est})$	Water formation volume factor at $P=P(\text{est})$
B_{wi}	Water formation volume factor at $P=P_{rw}$ (BWI keyword)
$B_o(\text{est})$	Oil formation volume factor at $P_b=P_b(\text{est})$ in a PVT region
$B_g(\text{est})$	Gas formation volume factor at $P=P(\text{est})$ in a PVT region
$R_s(\text{est})$	Solution gas ratio at $P_b=P_b(\text{est})$ in a PVT region
P_{rw}	Water reference pressure (PRW keyword)
$P(\text{est})$	Estimated average reservoir pressure in a PVT region (estimated from REFPRES Keyword)

P_b (est)	Estimated average bubble point pressure in a PVT region (estimated from PBT table or PB array)
C_o	Oil compressibility above the bubble point (CO keyword)
C_w	Water compressibility (CW keyword)

The simulator array PCOW_SHIFT corresponding to the BUILDER property “Water Cap Pres Shift/Offset” will now be written when the data set is saved.

Finish defining the model and when complete run the simulator for one timestep. Open the resulting SR2 file in RESULTS and examine the WOC/WGC distribution (viewing Water Saturation) and compare this with the desired distribution. Normally differences are very small.

Some Geological packages output Capillary Pressure Entry Pressure values for each grid block. PCOW_SHIFT may be used to accept Entry Pressure input.

Horizontal Wells

Horizontal wells can be easily simulated with IMEX in the following way:

This method involves modelling the well as a line source (injector) or sink (producer). This method neglects wellbore frictional pressure drop and liquid holdup effects. It is important to take wellbore crossflow into account by using the '***XFLOW-MODEL *FULLY-MIXED'** crossflow model.

The same sequence applies for a producer or an injector. Initially, your data set contains:

- a) ***WELL**
- b) ***PRODUCER**
- c) ***OPERATE**

Start with a constant pressure (bottomhole pressure) constraint. Use ***MAX**.

- d) ***GEOMETRY**

You may set the wellbore direction with this keyword. For horizontal wells, you need to use the ***I** or ***J** direction.

- e) ***PERF**

It is ideal for horizontal or deviated wells. Remember to use ***GEO** with it.

***GEOMETRY** and ***PERF** result in the output of well productivities. Run this data set and observe the resulting production rates.

Remove ***GEOMETRY**, input your own productivities directly with ***PERF** and rerun. Keep adjusting the well productivity until you get the desired production rate.

When you get the desired rate, rerun the data set using a rate constraint as the primary operating constraint.

Multilateral horizontal wells can be modelled by using the ***FLOW-TO** ***FLOW-FROM** layer keywords.

Vertical Equilibrium Calculation

IMEX currently provides seven different ways to initialize the reservoir.

1. *USER_INPUT
2. *VERTICAL *BLOCK_CENTER *WATER_OIL
3. *VERTICAL *BLOCK_CENTER *WATER_GAS
4. *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS
5. *VERTICAL *DEPTH_AVE *WATER_OIL
6. *VERTICAL *DEPTH_AVE *WATER_GAS (*TRANZONE)
7. *VERTICAL *DEPTH_AVE *WATER_OIL_GAS

In the following each of these options will be described:

*USER_INPUT:

This option is used to directly specify the values of the pressure and saturation for each grid block. The following keywords, which must be located in the Initial Conditions section, are used with this option.

- a) USER_INPUT
- b) *PRES or *PREST (pressure)
- c) *SW (water saturation, need not be entered if using the two phase model, *MODEL *OILWATER)
- d) *PB or *PBT (bubble point pressure)
- e) *PDEW or *PDEWT (dew point pressure
*GASWATER_WITH_CONDENSATE or *VOLATILE_OIL models)
- f) *PBS (solvent bubble point pressure, if using any of the pseudo-miscible models)
- g) *SO (oil saturation, need not be entered if using the two phase model,
*MODEL *GASWATER)

Example:

```
*INITIAL
*USER_INPUT
*PRES *CON 3400.0
*PB   *CON 2500.0
*SO   *CON    0.5
*SW   *CON    0.2
```

Previous to version 93.10 this option was invoked using the keywords *VERTICAL *OFF. These keywords are still accepted by IMEX.

Using this option may cause a nonequilibrium initial distribution of fluids which may result in large fluid flow during the first few timesteps. This option is typically used when a simulation is started not from initial conditions but from another simulation run.

```
*VERTICAL *BLOCK_CENTER *WATER_OIL or  
*VERTICAL *BLOCK_CENTER *WATER_GAS or  
*VERTICAL *BLOCK_CENTER *WATER_OIL_GAS
```

These options are used to perform gravity-capillary equilibrium calculations based on the values of the water-oil, oil-gas, and water-gas contacts and the value of the pressure at a reference depth. The saturations for the entire grid block are set equal to their values at the grid block center.

*VERTICAL *BLOCK_CENTER *WATER_OIL is used for reservoirs containing only water and oil phases. The value of the water-oil contact depth is specified using the keyword *DWOC.

*VERTICAL *BLOCK_CENTER *WATER_GAS is used for reservoirs containing only water and gas phases initially. The value of the water-gas contact depth is specified using the keyword *DWGC. *VERTICAL

*BLOCK_CENTER *WATER_OIL_GAS is used for reservoirs containing water-oil and gas. The water-oil contact and gas-oil contact depth are specified using the keywords *DWOC and *DGOC.

*GASWATER and *GASWATER_WITH_CONDENSATE models must be initialized using either the *VERTICAL *BLOCK_CENTER *WATER_GAS* or VERTICAL *DEPTH_AVE *WATER_GAS initialization option. Use of the *DWGC keyword is required with the *WATER_GAS initialization option.

The following keywords, which must be located in the Initial Conditions section, are used with these options.

- a) *REFDEPTH (reference depth)
- b) *REFPRES (pressure at reference depth)
- c) *PB or *PBT (bubble point pressure)
- d) *DWOC (water-oil contact)
- e) *DWGC (water-gas contact)
- f) *DWOC and *DGOC (water-oil and gas-oil contacts)
- g) *PDEW or *PDEWT (dew point pressure – Condensate models only)

Optionally the user can define the initial water saturation along with all of the above by using the *SWINIT option. This option attempts to adjust/alter P_{cow} curves to allow user defined water saturations to be in gravity-capillary equilibrium.

Care has to be taken when this option is employed, nonzero P_{cow} curves must be used and reasonable values of saturation must be entered. Highly altered P_{cow} can significantly effect other aspects of the model. Please see the *SWINIT keyword for more details.

Example:

```
*INITIAL  
*VERTICAL *BLOCK_CENTER *WATER_OIL_GAS  
*PB *CON 2500.0 **bubble point is constant  
    **throughout the reservoir.  
*REFDEPTH 900.0  
*REFPRES 3000.0  
*DWOC     1250.0  
*DGOC     788.0
```

For the above *BLOCK_CENTER options, the value of the saturation assigned to a given grid block is given by the value of the saturation at the grid block centre. In this way, for example, if a grid block has its block center slightly below the water-oil contact depth, the water saturation assigned to the block will be equal to the value of the water saturation below the water-oil contact.

```
*VERTICAL *DEPTH_AVE *WATER_OIL or  
*VERTICAL *DEPTH_AVE *WATER_GAS (*TRANZONE) or  
*VERTICAL *DEPTH_AVE *WATER_OIL_GAS
```

These options are also used to perform gravity-capillary equilibrium calculations based on the values of the water-oil, oil-gas, and water-gas contacts and the value of the pressure at a reference depth.

In contrast to the *VERTICAL *BLOCK_CENTER option the value of the saturation assigned to a given grid block is the average over the grid block volume of the saturations, and not simply the value of the saturation at the grid block centre. 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.

The *TRANZONE option on the *WATER_GAS keyword allows the use of Gas water capillary pressure that can be entered in the water relative permeability table (*SWT). The *TRANZONE keyword can be omitted if P_{cgw} is not entered.

*GASWATER and *GASWATER_WITH_CONDENSATE models must be initialized using either the *VERTICAL *BLOCK_CENTER *WATER_GAS* or VERTICAL *DEPTH_AVE *WATER_GAS initialization option. Gas-water and gas-liquid (*GASWATER_WITH_CONDENSATE condensate option) capillary pressures are therefore only input using the *SWT table. Use of the *DWGC keyword is required with the *WATER_GAS initialization option.

The following keywords, which must be located in the Initial Conditions section, are used with these options.

- a) *REFDEPTH (reference depth)
- b) *REFPRES (pressure at reference depth)

- c) *PB or *PBT (bubble point pressure)
- d) *DWOC (water-oil contact)
- e) *DWGC (water-gas contact)
- f) *DWOC and *DGOC (water-oil and gas-oil contacts)
- g) *PDEW or *PDEWT (dew point pressure – Condensate models only
*GASWATER_WITH_CONDENSATE or *VOLATILE_OIL)

Example:

```
*INITIAL
*VERTICAL *DEPTH_AVE *WATER_OIL
*PB *CON 2500.0 **bubble point is constant
      **throughout the reservoir.
*REFDEPTH 900.0
*REFPRES 3000.0
*DGOCL    788.0
```

Defining Multiple PVT Regions/Initialization Regions

Multiple PVT/Initialization regions can easily be created in IMEX. The cards associated with their definition are found in the Fluid Model section and optionally the Initial Conditions section.

This section will clarify the use of MPR's (Multiple PVT Regions) and multiple initialization Regions in light of the new *NREGIONS multiple initialization region entry system.

Fluid Model Section

The component property section is made of three types of data, (1) data which is entered per PVT region, (2) data which is entered once for the entire reservoir and (3) data which can either be entered per PVT region or once for the entire reservoir.

If a single type 3 data entry is found, it is applied to the entire reservoir, if multiple entries are found; the values are applied to each PVT region in turn.

The type 1 data (once per PVT region) are: PVT tables (*PVT, *PVTLO, *DIFLIB, *PVTAPI, *APIGRAD *PVTG *PVTCOND), *COT, *BOT, *VOT, *VGUST, *EGUST, *ZGUST, *BGUST, *DENSITY *OIL, *DENSITY *GAS, and *GRAVITY *GAS (*VOTAPI, *BOTAPI with the API-Tracking option)

The type 2 data (single entry applying to the entire reservoir) are: *REFPST, *GORINT, *MINSS, *PADSORP, *PMIX, *PVISC, *PREFCONC, and *PPERM

The type 3 data (either once per PVT region or a single entry applying to the entire reservoir) are: *DENSITY *WATER, *CO, *CVO, *BWI, *CW, *REFPW, *VWI, *CVW, *DENSITY *SOLVENT, *PVTS, *OMEGASG,

*TRES is a special position dependent case in that it defines the temperature used by PVT Tables which follow its definition. *TRES may be redefined before any PVT table or ZGUST table. Tables read in before each *TRES redefinition use the previous *TRES value, Tables read in after each *TRES redefinition use the new value.

The type 1 and 3 data types support two types of format (excluding PVT Tables and *TRES):

Format 1: A single entry per keyword.

For Example *DENSITY *OIL 45.0
 *DENSITY *OIL 47.0
 *DENSITY *OIL 46.5

The first line refers to the density of oil in region PVT region 1, the second line to region 2 and the third line to region 3.

Format 2: Multiple entries per keyword (WILDCARDS NOT ALLOWED).

For Example *DENSITY *OIL 45.0 47.0 46.5

The first value refers to the density of oil in region PVT region 1, the second value to region 2 and the third value to region 3.

Type 1, 2 and 3 data are position independent (*TRES is an exception). IMEX keeps track of the number of each type of data entered. During input processing IMEX records the maximum number of PVT regions referred to by any type 1 or 3 keywords.

If any of the type 1 data has fewer PVT regions defined than the maximum, an error is issued.

If any of the type 3 data has only a single region defined, this value is applied to the entire reservoir.

When specifying a single *PVTS table for multiple PVT regions, do not include the set_number in the *PVTS line.

If any of the type 3 data has greater than one region defined, but less regions defined than the maximum number of regions, and error is issued.

It is not possible to mix the use of the three oil compressibility options. Oil compressibility may be entered either using *CO (a single value or one value for each PVT region), *COT (one per PVT region), or by including an oil compressibility entry in each PVT table. Use of two or more input options causes an error to be issued. The undersaturated oil *BOT table may be used instead of any of any of the oil compressibility options (one per PVT region).

It is not possible to mix the use of the *CVO option and the *VOT option for determining undersaturated oil viscosity. A single option must be used for all PVT regions.

Initial Conditions Section (Assuming MPRs used to Define Multiple Initialization Regions)

1. As many MPRs are defined as Initialization Regions and *NREGIONS/ *ITYPE not input. The initialization regions are linked to the MPRs and use *PTYPE for their definition.
2. The keywords *PBT *PDEWT *REFPRES *REFDEPTH *DWOC *DGOC *DWGC *GOC_SW *WOC_SW *WOC_PC *GOC_PC are associated with MPR's. The remaining keywords apply to all blocks.
3. *PBT (or *PDEWT) cards expects the user to input the PVT region number which is associated with each table. Use of the *PB (or *PDEW) Cards define *PB (or *PDEW) for all blocks. One PBT (or *PDEWT) table must appear for every MPR defined in the PVT section.
4. The remaining cards (*REFPRES *REFDEPTH *DWOC *DGOC *DWGC *GOC_SW *WOC_SW), if required, are associated with MPR's. The first card is associated with the first MPR, the second card is associated with the second MPR, the n-th card is associated with the n-th MPR. For example:

```
*REFDEPTH 'Depth of PVT region 1'  
*REFDEPTH 'Depth of PVT region 2'  
*REFDEPTH 'Depth of PVT region 3'  
*REFPRES 'Pressure for PVT region 1'  
*REFPRES 'Pressure for PVT region 2'  
*REFPRES 'Pressure for PVT region 3'  
*DWOC    'Depth of WOC for PVT region 1'  
*DWOC    'Depth of WOC for PVT region 2'  
*DWOC    'Depth of WOC for PVT region 3'
```

is valid, as is:

```

*REFDEPTH 'Depth of PVT region 1'
*REFPRES 'Pressure for PVT region 1'
*DWOC 'Depth of WOC for PVT region 1'
*REFDEPTH 'Depth of PVT region 2'
*REFPRES 'Pressure for PVT region 2'
*DWOC 'Depth of WOC for PVT region 2'
*REFDEPTH 'Depth of PVT region 3'
*REFPRES 'Pressure for PVT region 3'
*DWOC 'Depth of WOC for PVT region 3'

```

The following is valid as well (**WILDCARDS NOT ALLOWED**):

```

*REFDEPTH 'PVT reg 1' 'PVT reg 2' 'PVT reg 3'
*REFPRES 'PVT reg 1' 'PVT reg 2' 'PVT reg 3'
*DWOC 'PVT reg 1' 'PVT reg 2' 'PVT reg 3'

```

It is important to not mix the use of the *DWOC-*DGOC keyword pair and the *DWGC keyword to define equilibrium in multiple PVT regions. *DWGC should be used when the initialization option is *WATER_GAS. An error will be issued if both the *DWOC *DGOC keyword pair and the *DWGC keyword are found in the data.

Unlike the Fluid Property section where all MPR data must be defined for type 1 data and possibly for type 3 data (if more than a single entry is found).

The initialization reader will default any undefined data to the value in the last PVT region defined.

For example if six PVT regions are defined and only three *DWOC entries are found:

```

*DWOC 2000.0
*DWOC 2100.0
*DWOC 2200.0

```

or

```
*DWOC 2000.0 2100.0 2200.0
```

The remaining three PVT regions will be assigned the value of *DWOC in *PVT region 3 (the last defined region) equivalent to:

```

*DWOC 2000.0
*DWOC 2100.0
*DWOC 2200.0
*DWOC 2200.0
*DWOC 2200.0
*DWOC 2200.0

```

or

```
*DWOC 2000.0 2100.0 2200.0 2200.0 2200.0 2200.0
```

Initial Conditions Section (Assuming *NREGIONS Input Defines Multiple Initialization Regions)

1. When *NREGIONS is defined, the number of initialization regions in the model is *NREGIONS. The *ITYPE array is used to assign initialization regions to gridblocks.
2. The number of PVT regions no longer determines the number of initialization regions, this is determined by *NREGIONS. It is no longer necessary to create dummy identical PVT regions to be able to model multiple initialization regions.

3. It is required that an initialization region (IR) should contain only one PVT region, but different initialization regions can contain different PVT regions.
4. Multiple PVT regions can still be used along with multiple initialization regions (for example two PVT regions and ten initialization regions), but the PVT regions are no longer required just to model multiple initialization regions. They model distinct PVT types.
5. If the maximum number of initialization regions (*NREGIONS) is greater than the largest value of *ITYPE (i.e. regions are defined which are not associated with any blocks) then the PVT properties of PVT region 1 will be associated with the empty initialization regions.
6. The keywords *PBT *PDEWT *REFPRES *REFDEPTH *DWOC *DGOC, *DWGC *GOC_SW *WOC_SW *WOC_PC *GOC_PC are associated with each of the initialization regions. The remaining keywords apply to all blocks.
7. *PBT (or *PDEWT) cards expects the user to input the initialization regions *ITYPE number which is associated with each table. Use of the *PB (or *PDEW) Cards define *PB (or *PDEW) for all blocks. One PBT (or *PDEWT) table must appear for every initialization region defined using *NREGIONS and *ITYPE.
8. The remaining cards (*REFPRES *REFDEPTH *DWOC *DGOC *DWGC *GOC_SW *WOC_SW), if required, are associated with initialization regions. The first card is associated with the first IR (initialization region); the second card is associated with the second IR, the n-th card is associated with the n-th IR. For example:

```
*REFDEPTH 'Depth of initialization region 1'
*REFDEPTH 'Depth of initialization region 2'
*REFDEPTH 'Depth of initialization region 3'
*REFPRES 'Pressure for initialization region 1'
*REFPRES 'Pressure for initialization region 2'
*REFPRES 'Pressure for initialization region 3'
*DWOC    'Depth of WOC for initialization region 1'
*DWOC    'Depth of WOC for initialization region 2'
*DWOC    'Depth of WOC for initialization region 3'
```

is valid, as is:

```
*REFDEPTH 'Depth of initialization region 1'
*REFPRES 'Pressure for initialization region 1'
*DWOC    'Depth of WOC for initialization region 1'
*REFDEPTH 'Depth of initialization region 2'
*REFPRES 'Pressure for initialization region 2'
*DWOC    'Depth of WOC for initialization region 2'
*REFDEPTH 'Depth of initialization region 3'
*REFPRES 'Pressure for initialization region 3'
*DWOC    'Depth of WOC for initialization region 3'
```

The following is valid as well (WILDCARDS NOT ALLOWED)::

```
*REFDEPTH 'IR 1' 'IR 2' 'IR 3'
*REFPRES 'IR 1' 'IR 2' 'IR 3'
*DWOC    'IR 1' 'IR 2' 'IR 3'
```

It is important to not mix the use of the *DWOC-*DGOC keyword pair and the *DWGC keyword to define equilibrium in multiple initialization regions. *DWGC should be used when the initialization option is *WATER_GAS. An error will be issued if both the *DWOC-*DGOC keyword pair and the *DWGC keyword are found in the data.

The initialization input reader will default any undefined data to the value in the last IR defined.

For example if six initialization regions are defined and only three *DWOC entries are found:

```
*DWOC 2000.0  
*DWOC 2100.0  
*DWOC 2200.0
```

or

```
*DWOC 2000.0 2100.0 2200.0
```

The remaining three initialization regions will be assigned the value of *DWOC in initialization region 3 (the last defined region) equivalent to:

```
*DWOC 2000.0  
*DWOC 2100.0  
*DWOC 2200.0  
*DWOC 2200.0  
*DWOC 2200.0  
*DWOC 2200.0
```

or

```
*DWOC 2000.0 2100.0 2200.0 2200.0 2200.0 2200.0
```

Defining Wells

Wells are defined using the following keywords.

***WELL**

(Required)

***PRODUCER**

(Required keywords which must follow well completion keywords.)

-or-

***INJECTOR**

-or-

***SHUTIN**

-or-

***OPEN**

***IWELLBORE**

(Optional. Use if you want to use the wellbore model for an injector.)

***PWELLBORE**

(Optional. Use if you want to use the wellbore model for a producer.)

***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.)

***GEOMETRY**

(Optional. It must precede a well completion keyword which is followed by subkeyword *GEO. Cannot be used with unweighted injectors.)

***PERF**

(At least one of these three or a combination thereof, is required.)

-or-

***PERFV**

These keywords must all reside in the Well and Recurrent 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.

Defining the Well Type

There are five well types. They are:

- a) *PRODUCER
- b) *INJECTOR
- c) *SHUTIN
- d) *OPEN, and
- e) *AUTODRILL

Each of these keywords must appear in the Well and Recurrent Data section and *PRODUCER or *INJECTOR must be defined before the well may be operated.

When a well is defined using the *WELL keyword, it acquires the status *SHUTIN; when the well's type is defined with *PRODUCER or *INJECTOR, it automatically acquires *OPEN status. *SHUTIN can be specified for a well any time after it is defined with *WELL; *OPEN can only be specified after the well has been typed with *PRODUCER or *INJECTOR.

A well can have its completions specified with *PERF before the type is specified, but it can operate only after it has been typed as a producer or an injector.

Example:

```
*WELL 1 'MED RIVER P1' *VERT 1 1
*WELL 2 'MED RIVER P2' *VERT 15 15
*WELL 3 'MED RIVER I1' *VERT 5 5
*WELL 4 'MED RIVER I2' *VERT 10 10
*PRODUCER 1:2
...
** both wells 3 and 4 are mobility weighted
** injectors.
*INJECTOR 3:4 *MOBWEIGHT
...
*PERFV 1:2
** The producer wells 1 and 2 are completed
** through K layers 1 through 3, each having
** a well index of 1.65
1:3 1.65
** The injector wells 3 and 4 are completed
** through K layers 2 and 3, each having a
** well index of 1.87.
*PERFV 3:4
2:3 1.87
```

How to Shut In a Well and Reopen It

Before shutting in a well:

1. It must be fully defined with
 - a) the operating constraints and any monitoring constraints, and
 - b) the well completion keywords.

After being fully defined a well may be shut in at any time using a *TIME or *DATE keyword. A well may be shut in immediately after it has been defined.

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

```
*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 *MOBWEIGHT 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 *MOBWEIGHT 3  
*INCOMP *SOLVENT  
*OPERATE *MAX *STS 1.2E+7  
*OPERATE *MAX *BHP 10000.0  
*PERF *GEO 3  
** if jf kf ff  
    1 1 3 1.0  
** 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.
```

Voidage Replacement

Voidage replacement option is initiated by specifying a group (gathering centre) using the keywords *GROUP and *WELL.

*GROUP allows the assignment of a number of wells in a voidage replacement group, whether they be producers or injectors. With each group, its own voidage replacement ratio is assigned using the keywords:

```
*GCONI    'group-name'      *VREP      *GAS      vrep_frac  
-OR-  
*GCONI    'group-name'      *VREP      *WATER    vrep_frac
```

Voidage may be replaced by injectors using the instantaneous injection potential or using guide rates specified by the *GUIDEI keyword.

An example data set is given below:

```
*RUN  
*DATE 1986 4 22  
*DTWELL 1.00  
** Define gathering centre for voidage replacement.  
*GROUP 'g1' *ATTACHTO 'field'  
** Define wells attached to the group 'g1'.  
*WELL 1 'GAS INJECTOR' *ATTACHTO 'g1'  
*WELL 2 'PRODUCER'      *ATTACHTO 'g1'  
** Injector 1 matches the voidage from producer 2  
*INJECTOR *MOBWEIGHT 1  
*INCOMP *GAS    ** Gas is being injected.  
*OPERATE *MAX *STG 1.0E+8  
*OPERATE *MAX *BHP 1.0E+8  
*PRODUCER 2  
*OPERATE *MAX *STO 20000.0  
*OPERATE *MIN *BHP 1000.0  
*MONITOR *MIN *STO 1000.0 *STOP  
*MONITOR *GOR      20000.0 *STOP  
*PERF 1  
** if jf kf wi  
1 1 3 1.0E+5  
** Well geometry for the producer.  
** rad geofac wfrac skin  
*GEOMETRY *K 0.25 0.34 1.0 0.0  
*PERF *GEO 2  
** if jf kf ff  
10 10 1 1.0  
** Specify voidage replacement. Complete voidage  
** is being replaced. Instantaneous injection  
** potential is used to allocate injection  
*GCONI 'g1'  
*VREP *GAS 1.0  
*IIP  
*TIME 3650.000  
*STOP
```

Using the Wellbore Model

To indicate that the wellbore model is required, you may use:

- a) *IWELLBORE, and
- b) *PWELLBORE

They are optional keywords and must be located in the Well and Recurrent Data section.

There are two occasions where you may want to use the wellbore model:

- a) When you want to use the wellbore model to calculate the wellhead pressure, or
- b) When you want to use the wellhead pressure as an operating constraint. You must use the wellbore option.

One or both keywords may be used. *IWELLBORE indicates that you want the wellbore option for an injector. *IWELLBORE must follow *INJECTOR immediately for identification. In turn, *PWELLBORE indicates that you want the wellbore model for a given producer. *PWELLBORE must immediately follow the *PRODUCER keyword for identification.

Example:

```
** The wellbore model is desired for well 2
** which is an injector.
*INJECTOR 2
*IWELLBORE
** wd      wl      rough    whtemp   rtemp    wr
  1100.   1225.   .1365    200.3    250.0   .33
```

Example:

```
** The wellbore model is desired for well 1
** which is a producer.
*PRODUCER 12
*PWELLBORE
** depth  ibhp
  1100.     1
```

TUBING DATA FOR PRODUCERS

Tubing data is required when *PWELLBORE is used. It must appear in the Well and Recurrent Data section after *PWELLBORE has been specified AND after the well definitions have been completed. For single-phase gas producers the analytical pressure loss model can be used.

With this keyword, you input on a *PTUBE keyword:

- a) *OIL or *LIQUID *WATER_GAS or *CONDENSATE
- b) a table number and a reference depth
- c) GOR, rates, water-cuts, (etc., see *PTUBE) and wellhead pressures.

Each of these are entered on separate lines with up to a maximum of 20 values for each. Finally, a bottomhole pressure table is entered.

The bottomhole pressure table:

- a) References the values you give for GOR, rate, water-cut, and wellhead pressure.
- b) Each value is indexed and the indices are used to reference the value in your bottomhole pressure table.
- c) The indices are then aligned with a set of up to 20 bottomhole pressure values.

The number of bottomhole pressures must be equal to the number of wellhead pressures already entered.

Example:

```
*PTUBE *OIL 1
*GOR
** ratio(1) ratio(2)
  500.0    1000.0
*QO
** q (1)   q(2)     q(3)     q(4)
  0.0      4000.0  8000.0  16000.0
**          wcut(1) wcut(2)
*WCUT    0.00      0.50
**          whp(1)   whp(2)   whp(3)
*WHP     200.0    900.0   1600.0

*BHPT
**igor iwcut iqo bhp(1) bhp(2) bhp(3)
  1       1     1   2176.1  2873.7  3573.7
  2       1     1   1997.7  2670.9  3370.9
  1       2     1   2646.7  3344.7  4044.7
  2       2     1   2447.7  3124.7  3824.7
  1       1     2   2135.5  2876.6  3576.6
  2       1     2   1682.7  2674.6  3374.6
  1       2     2   2618.0  3351.2  4051.2
  2       2     2   2189.0  3132.3  3832.3
  1       1     3   2133.6  2884.2  3584.2
  2       1     3   1463.1  2684.5  3384.5
  1       2     3   2630.9  3368.4  4068.4
  2       2     3   2022.0  3152.8  3852.8
  1       1     4   2160.1  2912.5  3612.5
  2       1     4   1425.7  2721.3  3421.3
  1       2     4   2696.4  3433.4  4133.4
  2       2     4   2080.0  3231.0  3931.0
```

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. Monitoring constraints are optional.

Each well introduces a new unknown variable P_{bh} , 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 other operating constraints is violated and *CONT has been used, then this violated constraint becomes the current operating constraint.

If more than one operating constraint is violated, then the most drastic assigned action is taken:

Most drastic:

*STOP
*SHUTIN

Least drastic:

*CONT

PRODUCERS

For a producer you should operate

- a) on a rate constraint (the primary operating constraint), and
- b) on a minimum bottomhole 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 pressure 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 primary operating constraint to the one that has just been violated.

*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 bottomhole 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 includes *MONITOR; then the constraint type, a value is then required for all but backflow. Finally, the action taken if there is a violation. Again, the most drastic action is 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 *STO    1200.0  *CONT
*OPERATE *MIN *BHP    2500.0  *CONT
*MONITOR      *GOR    15000.0 *STOP
*MONITOR      *WCUT     .98   *STOP
```

Definition of Well Indices within IMEX

For a Cartesian, vari or corner point grid block, given

```
*GEOMETRY *K rad geofac wfrac skin  
*PERF *GEO wn  
    i j k ff
```

then for the perforation of the well in this block

$$WI = \frac{2\pi \sqrt{\text{perm}_i \text{perm}_j} h ff wfrac}{\ln\left(\frac{re}{rw}\right) + \text{skin}}$$

where perm_i and perm_j are the permeabilities for block (i,j,k) in the x and y directions respectively; h = grid block length in the K direction,

$$re = \text{geofac} \sqrt{\frac{di dj}{wfrac \pi}}$$

and di and dj are the grid block lengths in the I and J directions.

For the same geometry card, but replacing the *PERF and the following line with

```
*PERF *KH wn  
    i j k kh
```

then for the perforation of the well in this block

$$WI = \frac{2\pi kh wfrac}{\ln\left(\frac{re}{rw}\right) + \text{skin}}$$

If

```
*PERF *WI wn  
    i j k wi
```

is used, no GEOMETRY is used and

$$WI = wi$$

For producers and mobility weighted injectors WI has units of md ft for field units and md metres for SI. For unweighted oil or water injectors, the units are bbl / psi for field, metres^{**3} / kPa in SI and metres^{**3}/(kg/cm^{**2}) in modified SI. For unweighted gas or solvent injectors, the units are ft^{**3}/psi for field, metres^{**3}/kPa in SI and metres^{**3}/(kg/cm^{**2}) in modified SI.

For mobility weighted injectors, WI is multiplied by the total (block) mobility.

For producers, WI is multiplied, for each phase, by the corresponding block phase mobility.

These then can be multiplied, for a nonbackflowing well, by the pressure in the wellbore (for the current well element) - the block pressure to obtain a well rate at reservoir conditions. The rate at surface conditions can be obtained dividing by the appropriate block volume formation factor for producers and wellbore volume formation factor for injectors.

Note that the block k_r 's in the above calculations, optionally may be replaced by a well element defined rock type set number and end points as described for the *PERF keyword.

For hybrid grids or cylindrical grids, the WI calculation is the same except r_e is given by the steady-state solution radial direction node location in the innermost grid block (see the description of the *GRID keyword for further details).

Printout of the well indices, as you mentioned, can be enabled using

```
*OUTPRN *WELL *ALL or  
*OUTPRN *WELL *LAYER or  
*OUTPRN *WELL *RESERVOIR
```

The header for this information is: Current Well Definitions.

The well indices for producers is indicated by w_p , for injectors by w_i and are given for o (oil), w (water), g (gas) and if it exists s (solvent).

Note: w_i for injectors, as you mentioned, is nonzero only for the injecting phase if it is not backflowing. Backflowing wells have WI's which are time dependent. See the *XFLOW-MODEL keyword description for details.

Input of Well Indices

To input well indices, these keywords are used:

- a) *GEOMETRY
- b) *PERF, or
- c) *PERFV

These keywords must reside in the Well and Recurrent Data section. The well completion keywords are required data, while *GEOMETRY is optional. *GEOMETRY may be used with mobility weighted injectors or producers.

*GEOMETRY requires the necessary parameters to calculate the well indices internally. The well completion keywords, *PERF, and *PERFV require the location of the well completion and the well index which you calculate.

If *GEOMETRY is used, then a well completion keyword is required with it. *GEO is used with the well completion keyword to indicate that the well parameters have been entered.

*GEOMETRY always proceeds *PERF and *PERFV.

*PERF is ideal for horizontal or deviated wells, but may be used with vertical wells also. It has the format:

Example:

```
*WELL 1 '12-09-18-56'  
*PERF 1  
** if jf kf wi setn  
    1   1   2:4  1.24  1  
  
-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 pertains to well 1.  
*PERF *GEO 1  
** if jf kf ff setn  
    1   1   2:4  1.   1
```

If *VERT was used with *WELL, then you have specified a vertical well. Use *PERFV.

Only the K direction grid block or range of blocks need be entered, since you have already entered the I and J location with *VERT. If you are using *GEOMETRY, use *GEO with *PERFV.

Example:

```
*WELL 2 *VERT 2 2  
*PERFV 1  
** kf wi setn  
    2:4  1.56  1
```

If you are using refined grid blocks and wells are located within the vicinity, then *PERF must be used. *GEO is again required if *GEOMETRY is used. *PERF requires the location of the fundamental grid block(s) where the well is completed and the location of the refined grid block(s) where the well is completed.

Example:

```
...
** Refinement will result in creating 3
** refined grids in the I direction, 3 in the
** J direction and two in the K direction in
** block (1,1,3).
*REFINE 3 3 2
*RANGE 1 1 3
...
*WELL 1
*PERF 1
** if jf kf ir jr kr wi setn
 1 1 3 / 2 2 1:2 1.75 1
```

Stopping a Simulation Run

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

Example:

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

Using Wildcards in Well Lists

A significant feature has been added to all keywords which allow the user to enter well lists (e.g. *ALTER 'well1' 'well2'). Wild cards have been incorporated in the following manner.

* replaces any number of characters at the end of a well name or can be used on its own to represent all wells (e.g. *ALTER '*' or *ALTER 'wel*').

? replaces any single character anywhere in the well name (e.g. *ALTER '?el?1').

The use of '*' followed by a character is not allowed (e.g. *ALTER 'we*11 results in an error)

The well name or wildcarded name must appear in quotes.

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.

For example:

If the wells in the model are 'well1' 'well2' and 'prod1', then:

'*'	would list all 3
'p*'	would list 'prod1'
'?ell?'	would list 'well1' and 'well2'
'?ell?*'	would list 'well1' and 'well2'
'?ell??'	would not match any well

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:

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.

1. Either *TIME or *DATE is required.
2. Define a value for *DTWELL, which is the first timestep 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, but if groups are to be used in the run they must be defined before the wells are defined.
4. Identify all new wells using *WELL. *ATTACHTO may be used to define a group if necessary.
5. Indicate the well locations, geometries, or the well indices using *GEOMETRY and any of the well completion keywords (*PERF, or *PERFV). This may be entered at a time later than that at which *WELL is entered for the well.
- 6a. Define a new well or a well with major operating changes as *PRODUCER and *INJECTOR. 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 pressure loss in wellbore model (tubing flow curves). You must have it if you choose *WHP as one of the operating constraints.
- 6c. Define the operating or monitoring constraints for that well.

Steps 1 through 6 MUST appear in any data set, although a bottom-hole pressure operating constraint will be defaulted if none is entered. Step 3 is optional. 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 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 or *INJECTOR.
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 16:

10. Define new wells. 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 or *INJECTOR. Use with *TIME or *DATE.

12. You may alter the polymer concentration if using the polymer concentration with *ALTERCP, once the well has been defined as a water injector. 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.

13. You may adjust the implicitness/explicitness as required (*AIMSET, *AIMWELL).
14. You may adjust the input/output controls and the transmissibility multipliers as required.
15. The keywords *DTWELL, *DTMAX, *DTMIN may also appear in subsequent well changes.
16. 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:

```
** Well data
*RUN
*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 *BACKFLOW *STOP
  **           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.
*ALTER 1
  100.0
*TIME 50.0
*ALTER 1
  1000.0
  ** Define an injector. Also modify the primary
  ** operating constraint for the producer again.
*TIME 720.0
*WELL 2 'WATER INJECTOR'
*INJECTOR 2 *MOBWEIGHT
*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 2
  **if jf kf ff
    7   7   3   1
```

```
*ALTER 2  
    100.0  
** Stop at 10 years  
*TIME 3650.0  
*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 and Recurrent 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 and Recurrent Data section:

- a) *RANGE
- b) *REFINE
- c) *TRANSI (*TRANLI)
- d) *TRANSJ (*TRANLJ)
- e) *TRANSK (*TRANLK)

From the Rock-Fluid Data section, only the rock type array and the end point arrays may appear in the Well and Recurrent Data section:

- a) *RTYPE
- b) *SWCON
- c) *SWCRIT
- d) *SGCON
- e) *SGCRIT
- f) *SOIRW
- g) *SORW
- h) *SLCON
- i) *SORG
- j) *KRWIRO
- k) *KROCW
- l) *KRGCL
- m) *KROGCG
- n) *PCWMAX
- o) PCGMAX

p) JFWMAX

q) JFGMAX

From the Numerical Methods Control section, these keywords may be changed:

a) *DTMAX, and

b) *DTMIN

Subsidence Output

In IMEX, keyword *SBDZ (for subsidence) has been added to the grid output list to display the vertical deformation of grid blocks. The formula used in IMEX is similar to the one that was implemented in STARS. It is available as a *GRID option for both output files and SR2 files as well as being available as a *SPECIAL variable

It is important to note that the subsidence calculated does not alter reservoir volume or flow characteristics and is used only for display purposes.

The process of development of the formula for subsidence output is shown as follows:

$$\varepsilon_V = \frac{dV_b}{V_b^0} \quad (1)$$

Where:

ε_V is volumetric strain.

V_b^0 is initial bulk volume of a block.

The equation (1) can be written as:

$$\varepsilon_V = \frac{dV_b}{V_b^0} = \frac{(V_b^n - V_b^0)}{V_b^0} \quad (2)$$

Assuming that only pore volume of a block is changing with pressure and the matrix portion of the block is constant. The above equation (2) can be rewritten as:

$$\varepsilon_V = \frac{(V_p^n - V_{matrix}) - (V_p^0 + V_{matrix})}{V_b^0} = \frac{(V_p^n - V_p^0)}{V_b^0} \quad (3)$$

Where:

V_p^n = Pore volume of a block at k or n iteration

V_p^0 = Initial pore volume of a block

V_{matrix} = Matrix volume of a block

However,

$$V_p^n = V_b^0 \varphi_n \quad (4)$$

$$V_p^0 = V_b^0 \varphi_0 \quad (5)$$

Substitute (3) and (4) into (2) and simplify to give:

$$\varepsilon_V = (\varphi_n - \varphi_0) \quad (6)$$

Where:

φ_n = Porosity at k or n iteration

φ_0 = Initial porosity.

Furthermore, assuming that deformation does not occur in the X and Y direction. The cross-sectional area parallel to the XY plane is, thus, constant. Based upon this assumption, equation (2) can be written as:

$$\varepsilon_v = \frac{(V_b^n - V_b^0)}{V_b^0} = \frac{(Ah^n - Ah^0)}{Ah^0} = \frac{\Delta h}{h^0} \quad (7)$$

Where:

A is cross section area that is parallel to the XY plane.

h^n is thickness of the block at k or n iteration.

h^0 is initial thickness of the block.

Comparing equation (6) and equation (7), this will lead to:

$$\Delta h = h^0 (\phi_n - \phi_0) \quad (8)$$

Equation 8 is used to calculate subsidence for each block.

The total vertical subsidence (i.e. the sum of the subsidence of each vertical layer) is also calculated when “*OUTSRF *GRID *SBDZ is requested and can be displayed using Results3D.

Parallel IMEX

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  
*PNTHRDS n
```

“*PPATTERN *AUTOPSLAB n” has been tested for n up to 64 corresponding to 64 processors. 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  
*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”

Using the Seawater Injection Option

In IMEX, seawater can be injected into the reservoir. Injected seawater is treated as an active tracer in the water phase. Combined with the well scale buildup option, seawater tracking allows the user to model perforation damage due to the scale deposition caused by seawater mixing with formation brine. When the seawater injection option is in use, the term ‘water’ throughout the simulation denotes the total water which includes the formation water and the seawater.

There are two reservoir models which use the seawater injection option. Both the *BLACKOIL and *OILWATER can model seawater injection by specifying *MODEL *BLACKOIL_SEAWATER or *MODEL *OILWATER_SEAWATER respectively. These two keywords must be declared in Component Properties section. Additional keywords can optionally be used as below.

1. Input/Output Control section

Use the sub-keyword *SEAWF in the grid_list of *OUTPRN/*OUTSRF *GRID to output the distribution of seawater volume fraction during the run.

2. Component Properties section

Use *SVISC to define the seawater viscosity if applicable. By default, the formation water viscosity of PVT Region 1 will be used.

3. Initial Conditions section

*SEAWATFRC can be used to initialize the reservoir seawater volume fraction as a user defined distribution. The default initial seawater fraction is zero. It is normally not necessary to define this as initially only formation water exists in the reservoir.

4. Numerical Method Control section

Use *CONVERGE *MAXRES *SEAWATER to change the default convergence criteria if necessary. This keyword is normally not required.

5. Well and Recurrent Data section

Use *INCOMP *SEAWATER when defining an injector to indicate the injection of seawater. By default the seawater is 100% of the injection stream. However, a fraction following this keyword, if given, can be used to define the volume fraction of seawater injected. The following example indicates that of the 5000 bbls/day of injection water, 80% is seawater and remainder is formation water.

```
*INCOMP *SEAWATER 0.8
... *OPERATE *MAX *STW 5000.0
```

The volume fraction of injected seawater can be altered by using *ALTERFS in a manner analogous to *ALTERCP.

Using the Scale Buildup Option

Scale deposition is a common problem in hydrocarbon reservoirs where injected seawater mixes with formation brine. The scale buildup option in IMEX allows the user to model well damage (a reduction of productivity) due to the scale deposited on individual perforations of a well.

The well damage can either be modeled as a function of both water production and the fraction of seawater produced in the water or as a function of water production alone and a fixed scale deposition rate per volume of water produced (if the seawater injection option is not used).

In IMEX, the mass of scale deposited within a timestep is calculated as the volume of water which has flowed through the perforation during the timestep multiplied by a corresponding precipitation. The precipitation is a function of the seawater fraction, which is given through a user defined deposition table. The accumulated scale causes damage to the well PI. The relationship between the PI damage factor and the existing scale is also given through a user defined damage table.

Modeling of scale buildup as a function of both seawater fraction and water production requires four steps.

1. Use one of the two seawater injection models to enable seawater tracking in the reservoir. Please see ‘Using the Seawater Injection Options’.
2. In the Rock-Fluid Data section, define the scale deposition tables and well damage tables by using *SCLDPS and *SCLDMG respectively. Multiple tables can be specified.
3. In the Well and Recurrent Data section, assign both the deposition and damage tables to all perforations (layers) in a well or specific perforations (layers) of a well. This is done by using the keyword *SCLtbl-WELL or *SCLtbl-LAYER. The *SCLtbl-WELL keyword is applied to all perforations of the well(s) and the *SCLtbl-LAYER keyword is applied to specified perforations of a well.
4. In the Well and Recurrent Data section define seawater injection wells rather than formation water injection wells using: *INCOMP *SEAWATER

Perforations (layers) linked to deposition and damage tables will keep track of scale deposition at a perforation over a timestep, keep track of the total scale deposited on a perforation, and calculate damage based on total scale deposited/length of perforation.

Within the recurrent data, connections can be reassigned to different tables. Deposited scale accumulates until it is totally or partly be removed. Scale removal is performed by using the keyword *SCLRMV-WELL or *SCLRMV-LAYER. Again the *SCLRMV-WELL keyword is applied to all perforations of the well(s) while the *SCLRMV-LAYER keyword is applied to the specified perforations of a well.

If the scale buildup option is to be used without the seawater injection option, scale precipitation is no longer considered as a function of seawater fraction. The scale buildup option can be used without seawater tracking but damage becomes a function only of the amount of water produced. Scale buildup/well damage can be modeled using the following steps:

1. Use any PVT model.
2. In Rock-Fluid Data section, define the deposition tables (*SCLDPS) with a zero seawater fraction row and constant scale precipitation. For example,

*SCLDPS		
**Seawater fraction		Scale precipitation (lbm/bbl)
0.00		1.10e-05
1.00		1.10e-05

Since seawater flow is not modeled when the seawater injection option is not used, the seawater fraction in the reservoir is always zero. Therefore, in the above table, only the first entry (at zero volume percent seawater) is ever used, and hence a constant precipitation of 1.10e-05 pounds of scale deposited per barrel of produced water is employed. The damage tables (*SCLDMG) can be defined normally.

3. In the Well and Recurrent Data section, scale table assignment and scale removal actions can be defined normally.

Simulating a Volatile Oil Reservoir

To simulate a volatile oil reservoir:

1. Use *MODEL *VOLATILE_OIL.
2. Enter saturated gas and oil phase PVT properties either using:
 - The *PVTCOND table, optionally gas properties for undersaturated gas may be entered using the *EGUST (*BGUST or *ZGUST) and *VGUST tables.
 - The *PVTVO table, *EGUST (*BGUST or *ZGUST) and *VGUST cannot be entered, as both wet and dry (undersaturated) gas properties are entered in the *PVTVO table format.
3. Enter all oil phase properties which are required for pressures above the bubble point, such as *CO, *BOT, *COT, *CVO, *VOT.

These keywords must be located in the Fluid Component Data section of the data set.

4. Establish the gas-oil and oil-water contacts as if initializing a black oil model (*MODEL *BLACKOIL).
5. Define the initial Bubble Point using the *PB or *PBT keywords.
6. Define the initial Dew Point using the *PDEW or *PDEWT keywords.
7. Define all other initial conditions as if this were a black oil model.

These keywords must be located in the Initial Conditions Data section of the data set.

8. Enable output options which aid in model analysis with the following steps:
 - a) *OUTPRN/*OUTSRF *GRID *RV *DPP outputs Oil Content and Dew Point Pressure to the output and SR2.
 - c) *PSPLIT *ON adds well production split information to the well output (Output file and SR2 file). The surface oil production is made up of oil from condensate and black oil. The surface gas production is made up of free gas and solution gas.

These keywords must be located in the Input Output Control section of the data set.

9. Enable numerical control options if necessary:
 - a) *NORM *PDW allows you to control timestep size based on dew point pressure changes. *NORM PBUB and *NORM PDW both set a single timestep control parameter which control the average timestep size as a function of bubble point pressure and/or dew point pressure change.
 - b) *MAXCHANGE *PDW allows you to control maximum timestep size allowed based on dew point pressure changes. *MAXCHANGE PBUB and * MAXCHANGE PDW both set a single timestep control parameter which control the maximum timestep size as a function of bubble point pressure and/or dew point pressure change.

These keywords must be located in the Numerical Methods Control section of the data set.

Using the Subdomain-Dual Permeability-Reinfiltration Model

The SD-DK-R model is controlled by the following keywords:

*SUBDOMAIN “idiv” (turns on subdomain dual porosity model, each subdomain consists of “idiv vertical blocks) – Reservoir Description section.

*FRACVOL (controls the thickness of individual subdomain divisions) – Reservoir Description section

*TRANSD (controls SD-DK-R inter-subdomain connection) – Reservoir Description section

*SD_REINF (determines if SD-DK-R inter-subdomain connection is used to represent reinfiltration) – Rock-Fluid Data section

The SD-DK-R is capable of modeling a wide variety of dual porosity scenarios based on TRANSD and SD_REINF input values.

In general the number of SUBDOMAIN divisions should be set to 5. It is acceptable to use more divisions if CPU time and memory are not constraining factors.

In addition, FRACVOL should be used to reduce the thickness of both the top and bottom subdomain divisions. In the case of *SUBDOMAIN 5, a FRACVOL setting of:

*FRACVOL 0.02 0.32 0.32 0.32 0.02 minimizes the error in capillary holdup.

Modeling Capability of SD-DK Model

1. Standard Subdomain Model

If the *TRANSD keyword is not included in the model *SD_REINF is not required and the model reverts to the standard *SUBDOMAIN model. It is still an excellent idea to use *FRACVOL to minimize the error in the capillary holdup.

2. Gravity Drainage between Subdomains

Using the *TRANSD keyword to set non-zero transmissibility multipliers between domains allows the fluid in the bottom of a subdomain to flow into the top of the subdomain directly below it without having to flow into the fracture first. In essence this is a hybrid method which combines the Subdomain model and the Dual Permeability model and gives similar results. As there is no capillary holdup in the matrix, *FRACVOL use does not significantly improve the models results.

3. Reinfiltration between Subdomains

As in 2) the non-zero *TRANSD keyword enables inter-subdomain flow. The use of *FRACVOL and *SD_REINF together allows the user to use the bottom-most subdomain block as a proxy for a horizontal fracture. The use of *FRACVOL to reduce the thickness of the bottom division of the subdomain reduces the amount of fluid stored in the fracture proxy block, and minimizes the error in capillary holdup. *SD_REINF = 1.0 sets the bottom subdomain division’s capillary pressure to 0.0 (the fracture’s value) and introduces a real capillary discontinuity in the flow between the two subdomains.

4. Gravity Drainage and Reinfiltration between Subdomains

Actual drainage from Subdomain blocks is a combination of gravity drainage and reinfiltration. This may be approximately accounted for by scaling back the bottom subdomain division's capillary pressure rather than setting it explicitly to zero.

This can be accomplished by setting *SD_REINF to a value between 0.0 and 1.0. The fracture proxy block's capillary pressure is scaled by $1 - *SD_REINF$.

Modelling Non Darcy Flow in Hydraulic Fractures Accurately Using a Grid Based Approach

It is possible to use the *NONDARCY *GENERAL 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_{eff}) 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 (K_f) 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 $K_f \times A$ 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_{eff} 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:

$$\text{The value of NDARCYCOR} = (K_f / K_{\text{eff}})^{(2-N1g)}$$

N1g is the exponent of the $(K_{\text{rg}} \times K)$ 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:

$$\text{NDARCYCOR} = 10000/83.33^{(2-N1g)} = 10000/83.33^{1.5} = 1314.6.$$

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 *GENERAL 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_{eff} and NDARCYCOR to account for varying K_f and fracture width.

In addition there should be a correction factor for each phase (if N1 for different phases differed), but normally identical β factor correlations are used for each phase, so this is not an issue.

The combination of the effective fracture permeability (to preserve $K_f \times A$ in the fracture) and the NDARCYCOR factor as calculated above (to correct the Forchheimer number for K_{eff}) 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.

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 D_i (parent block) = 200.0, D_i (iwell) = 1.0

Determine the center refinement:

If $NLGRi = 21$, the central refinement = $\text{Int}(NLGRi/2) + 1 = 11$

This sets the dimension in the I direction of the well block and its location in the I direction within the LGR

D_i (iwell) = D_i (11)

The fracture is symmetric about block 11 in the I direction so

D_i (iwell+1) = D_i (iwell-1), D_i (iwell+2) = D_i (iwell-2),

D_i (iwell+3) = D_i (iwell-3), etc.

or

D_i (12) = D_i (10), D_i (13) = D_i (9), D_i (14) = D_i (8), etc.

The LGR block sizes are all based on the size of the well block (D_i (iwell)) so that

D_i (iwell+I) = $\frac{1}{2}D_i$ (iwell) $\times (R^I - R^{(I-1)})$ where $I = 1, \text{Int}(NLGRi/2)$

and

D_i (iwell-I) = $\frac{1}{2}D_i$ (iwell) $\times (R^I - R^{(I-1)})$ where $I = 1, \text{Int}(NLGRi/2)$

Where $R = (D_i \text{ (Parent Block)} / D_i \text{ (iwell)})^{(1/\text{Int}(NLGRi/2))}$

For the J direction a similar calculation is done:

In the J direction, where D_j (parent block) = 150.0, D_j (jwell) = 1.0

Determine the center refinement:

If $NLGRj = 17$, the central refinement = $\text{Int}(NLGRj/2) + 1 = 9$

This sets the dimension in the j direction of the well block and its location in the j direction within the LGR

D_j (jwell) = D_j (9)

The fracture is symmetric about block 9 in the j direction so

D_j (jwell+1) = D_j (jwell-1), D_j (jwell+2) = D_j (jwell-2),

D_j (jwell+3) = D_j (jwell-3), etc.

or:

D_j (10) = D_j (8), D_j (11) = D_j (7), D_j (12) = D_j (6), etc.

The LGR block sizes are all based on the size of the well block (D_j (jwell)) so that

D_j (jwell+J) = $\frac{1}{2}D_j$ (jwell) $\times (R^J - R^{(J-1)})$ where $J = 1, \text{Int}(NLGRj/2)$

and

D_j (jwell-J) = $\frac{1}{2}D_j$ (jwell) $\times (R^J - R^{(J-1)})$ where $J = 1, \text{Int}(NLGRj/2)$

Where $R = (D_j \text{ (Parent Block)} / D_j \text{ (jwell)})^{(1/\text{Int}(NLGRj/2))}$

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 Data set (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 Data set (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 Keff
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 Data set (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
```

Using the Fracture Clean-up Model

The fracture clean-up model is made up a number of options specifically developed to allow IMEX to model flow back of fracturing fluids around a newly fractured well.

The option was developed to be used with the polymer option in IMEX as the fracturing fluid is modeled as a water-polymer mixture (Only the velocity and polymer concentration dependent water viscosity and fracture width correction features actually require the polymer model to be used).

The option is made up of the following features.

Features

Velocity and polymer concentration dependent water-polymer mixture viscosity can be used to model shear thinning/thickening fluids.

Fracture width correction for reservoir condition velocity to be used in the above water-polymer mixture viscosity calculation. A zero value for the correction eliminates the velocity dependency in regions away from the fracture. A correction value of 1.0 uses the block average reservoir condition velocity directly.

Block and direction dependent pressure gradient thresholds to flow in specified regions. Once a threshold has been exceeded the flow connection will always allow flow regardless of the pressure gradient. This can be used to approximate yield stress

Non equilibrium initial saturation overrides can be entered in user defined regions (e.g. near a fracture) in a model which is initialized everywhere else using a gravity equilibrium initialization technique. This can be used to set up fracture fluid distributions in the fracture without the need to model fracture fluid injection.

The ability to initiate the pressure gradient threshold option at a restart is added. This allows the model to simulate fracture fluid injection followed by clean-up (after the restart).

All of the above features can be used with or without the non Darcy flow option and can be used independently of each other.

Velocity Dependent Polymer Mixture Viscosity (Component Properties Section)

A new option in the *PMIX keyword has been added.

*PMIX *VELTABLE

This signals IMEX to expect velocity dependent polymer-water mixture viscosity data.

IMEX expects to read the keyword VWT followed by an initial velocity value on the same line. Following this at least two relative polymer concentration - relative water-polymer mixture viscosity rows are expected to be read in.

After the entire table at the initial reservoir condition velocity value has been entered the next occurrence of VWT followed by a larger reservoir condition velocity will trigger the reading of the next relative polymer concentration - relative water-polymer mixture viscosity table (at least two rows).

If only a single reservoir condition velocity is entered (only one value of VWT), the table will become velocity independent.

The relative polymer concentration is the actual polymer concentration divided by the reference polymer concentration *PREFCONC. The relative water-polymer mixture viscosity is the mixture viscosity divided by pure water viscosity.

An example of the *PMIX *VELTABLE keyword is given below for a shear thinning fluid at velocities 0.10, 10.0 and 40.0 ft/day. The velocities used for the table lookup are intrinsic reservoir condition velocities and so are calculated using the average velocity in a grid block divided by the block porosity. In addition, the velocities are multiplied by a fracture width correction term which will be discussed shortly ($\text{Vel}_{\text{cor}} = \text{FRWIDTHCOR} * \text{Reservoir Condition Velocity}/\phi$).

```
*PMIX *VELTABLE
*VWT 0.10
 0.0      1.00
 0.2      1.50
 0.6      2.00
 1.0      5.00
*VWT 10.0
 0.0      1.00
 0.2      1.20
 0.6      1.75
 1.0      3.00
*VWT 40.0
 0.0      1.00
 0.2      1.10
 0.6      1.40
 1.0      2.00
```

For velocities below 0.10 ft/day, the table at 0.10 ft/day is used. For velocities above 40.0 ft/day the table at 40.0 ft/day is used.

Fracture Width Correction (Rock-Fluid Data Section)

The model includes a fracture width correction array *FRWIDTHCOR which is used to enable/disable velocity dependent water-polymer mixture viscosity calculations in portions of the reservoir and to increase the effective velocity in the fractured region due to numerical fracture representation.

*FRWIDTHCOR = 0.0 turns off the velocity dependent viscosity calculation for the block it is defined in. The model will always use the lowest velocity (*VWT) in the *PMIX *VELTABLE table in this case.

A non zero *FRWIDTHCOR is used to multiply the intrinsic water-polymer mixture velocity for use in the *PMIX *VELTABLE table. For example if a 0.1 millimeter fracture is represented using a 0.2 meter grid block a *FRWIDTHCOR of 2000.0 might be used to approximate the much higher velocity along the actual fracture.

When *PMIX *VELTABLE is encountered, a two dimensional interpolation is used. Water-polymer mixture velocity is calculated for each grid block. The velocity is divided by grid block porosity and multiplied by a correction factor (*FRWIDTHCOR), or:

$$\text{Vel}_{\text{cor}} = \text{FRWIDTHCOR} * \text{Reservoir Condition Velocity}/\phi$$

Where Vel_{cor} is the corrected velocity used in the VELTABLE table, FRWIDTHCOR is the fracture width correction term entered on a block by block basis, Velocity is the average velocity in a block and ϕ is the block's porosity

Block and Direction Dependent Pressure Gradient Thresholds (Rock-Fluid Data Section)

The fracture clean-up model approximately accounts for yield stress effects by including pressure gradient thresholds in the three directions. The arrays *PTHRESHI, *PTHRESHJ, and *PTHRESHK define these pressure gradients between every grid block and its neighbor.

The array value *PTHRESHI *IJK I, J, K defines the pressure gradient (pressure drop / unit of length) required to initiate flow between block I, J, K and I+1, J, K.

The array value *PTHRESHJ *IJK I, J, K defines the pressure gradient (pressure drop / unit of length) required to initiate flow between block I, J, K and I, J+1, K.

The array value *PTHRESHK *IJK I, J, K defines the pressure gradient (pressure drop / unit of length) required to initiate flow between block I, J, K and I, J, K+1.

The pressure gradient threshold is a one time switch. Once it is exceeded between two blocks in a specific direction, the interblock connection remains open to flow thereafter.

A zero value of the thresholds indicates the connection is always open to flow.

Regions in and around the fracture may have non zero values of *PTHRESHI, *PTHRESHJ, and *PTHRESHK, while regions in the reservoir might have zero values for thresholds.

Non Equilibrium Initial Saturation Override (Initial Conditions Section)

In cases where the initial injection of fracturing fluid into the fracture is not modeled, it would be convenient to be able to initialize the entire reservoir to a gravity-capillary pressure equilibrium state and just override this state in the region in and around the fracture which has just had fracturing fluid pumped into it.

This can be done using the keywords *SONEQ and *SWNEQ. These keywords override the gravity-capillary pressure equilibrium saturations determined when using the *VERTICAL *BLOCK_CENTER or *VERTICAL *DEPTH_AVE equilibrium options in specified regions.

*SONEQ and *SWNEQ should only be defined (using the *IJK array reading option) in the region around the fracture the user wishes to override saturations in. Normally around the fracture we expect a high saturation of the fracturing fluid, this is modeled as water with a high polymer concentration. Using high initial *POLYCONC (polymer concentration) and high *SWNEQ (low *SONEQ) in this region is a convenient way to do this.

Initiating the Pressure Gradient Threshold Option at a Restart

On occasion it may be necessary to also model the fracturing fluid injection period of the fracture clean-up process. It is possible to accomplish this by restarting from the end of the injection period (or shut in period) before the flow back period and only introducing pressure gradient threshold calculations in the restarted run.

The fracture cleanup model can determine if the *PTHRESHI, *PTHRESHJ, *PTHRESHK keywords were in the data set of the model to be restarted from. If they were not, and the pressure gradient threshold keywords are encountered in the current data set, then the pressure gradient threshold option is initiated on the restart. On the restart, the pressure field read from the restart record is set as the initial pressure field. A pressure gradient that will initiate flow in the I direction is then defined as:

$$DP = [(P(I+1,J,K) - \text{Prestart}(I+1,J,K)) - (P(I,J,K) - \text{Prestart}(I,J,K))] / [\text{Distance}_I]$$

Distance_I is the distance between block centers in the I direction.

Prestart is the initial pressure field read from the restart.

P is the current pressure field.

DP is compared with $*\text{PTHRESHI}$, when DP is greater than $*\text{PTHRESHI}$ (defined in block I,J,K) flow is initiated between blocks I,J,K and $I+1,J,K$.

On restarts before or after the introduction of the pressure gradient threshold keywords, the model restarts normally.

As the introduction of the pressure gradient threshold keywords on a restart can only be done once per run, this model cannot be used when wells are fractured at different times.

It is also possible and desirable to change the water-polymer mixture viscosity during the restart which separates the injection period from the flow back period. In the injection portion of the run, the fracturing fluid is a very high viscosity fluid. During flow back the water-polymer mixture fluid has a very much lower viscosity. Similarly, it is also possible to alter $*\text{FRWIDTHCOR}$ on the restart.

When the mixture viscosity is altered on a restart, it is critical that the $*\text{PMIX} *\text{VELTABLE}$ table be used both for injection and production periods. The values in the table, the number of points in each table, and the number of velocity tables may be altered as required, but the $*\text{VELTABLE}$ option must be used.

Output of Pressure Thresholds and Fracture Width Correction (Input/Output Control Section)

The $*\text{RES}$ list of input arrays has been expanded to include FRWIDTHCOR and the three pressure threshold arrays (PTHRESHI , PTHRESHJ , PTHRESHK).

Using the Oil Wet Option

Notes:

1. The original oil wet option initialization in IMEX is still available with the BLOCK_CENTER reservoir initialization method. This version of the oil wet option initialization assumes the reservoir began its existence as an oil wet reservoir.
The new oil wet option initialization option assumes that the reservoir is initially water wet and has undergone a wettability alteration to become an oil wet reservoir. These assumptions have a significant impact on how fluids are placed during reservoir initialization. The user should understand completely the difference in order to obtain the initial conditions which most reflect his reservoir model.
2. Within a PVT 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.

Oil Wet Initialization (BLOCK_CENTER initialization):

The initialization of a BLOCK_CENTER oil wet reservoir assumes that the reservoir is initially oil wet. We assume that initially we have a 100% oil filled reservoir into which water migrates. As water is heavier than oil it fills the region of the reservoir below the present water oil contact.

As the migrating water cannot replace the connate oil:

- The water zone below the water oil contact is made up of water and connate oil.
 - The oil zone does not contain any water at all.
 - The gas zone is made up of gas and connate oil.
- The P_{cwo} (non-wetting minus wetting pressure) controls the transition zone of oil into the water zone.

It is possible to adjust the oil in the oil zone to be less than 100% by using the *WOC_SW keyword. When using water wet initialization this allows the user to adjust the amount of water in the water zone. When using the oil wet option with BLOCK_CENTER initialization, this now refers to the oil saturation (wetting phase saturation) in the oil zone. Hence it is possible to alter the oil phase saturation for each PVT region. Setting a WOC_SW equal to 0.80 limits the oil saturation in the oil zone to a maximum of 80%.

It is also possible to adjust the amount of oil above the gas oil contact by using the *GOC_SW keyword. When using the oil wet option with BLOCK_CENTER initialization, *GOC_SW now controls the wetting phase (oil) saturation in the gas zone. Setting GOC_SW equal to 0.05 can change the maximum oil saturation in the gas cap to 5%.

This initialization option although entirely consistent does not represent the type of oil wet reservoirs normally encountered. More commonly oil wet reservoirs initially begin life as water wet and change wettability at a later time. This type of reservoir should be initialized using the DEPTH_AVE initialization option.

Oil Wet Initialization (DEPTH_AVE initialization):

This type of initialization (when used with the oil wet option) assumes 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 (not water and oil).

Unless specified explicitly, the WOC_PC (the capillary pressure at the water oil contact) is internally set equal to the maximum P_{cwo} (see P_{cwo} definition below), which occurs at the lowest oil (wetting phase) saturation. Therefore the transition zone occurs within the oil zone and not below the water oil contact.

If a WOC_PC equal to the minimum P_{cwo} , which occurs at the largest oil (wetting phase) saturation, is input in an oil wet model, the WOC would represent the free water level in the reservoir. The oil water transition zone would occur below the free water level.

A different WOC_PC can be entered for each PVT region.

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 modelled 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.

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

When modelling 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 nonwetting 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 nonwetting 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 (S_o), the shape of the P_{cwo} curves resembles that of the P_{cow} curves versus water saturation in a water wet system.

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

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 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. When using the BLOCK_CENTER initialization option the capillary pressure curve in the liquid gas table must be the P_{cwg} curve.

If the capillary pressure received is from a water-gas system, $p_{cgw} = p_g - p_w$, to accommodate IMEX table input for DEPTH_AVE 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}) if the *NOSWC option is not used, and equal to irreducible oil saturation (S_{oirg}) alone, if the *NOSWC option is used

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 nonwetting phase (water) saturation
S_{orw}	= Residual nonwetting phase (water) saturation
S_{org}	= Residual nonwetting phase (water) saturation (gas liquid table)

S_{lcon} = Connate liquid saturation which is equal to connate oil (S_{wcon}) plus irreducible water saturation (S_{oorg}) if the *NOSWC option is not used, and equal to irreducible water saturation (S_{oorg}) alone, if the *NOSWC option is used

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)
P_{cwmax}	=	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)
P_{cwmax}	=	Water-oil capillary pressure at connate oil
P_{cgmax}	=	Gas oil capillary pressure at connate liquid (DEPTH_AVE) Gas water capillary pressure at connate liquid (BLOCK_CENTER)
J_{fwmax}	=	Water oil J function at connate oil
J_{fgmax}	=	Gas oil J function at connate liquid (DEPTH_AVE) Gas water J function at connate liquid (BLOCK_CENTER)

Example Oil Wet Rock Type Curves (Relative Permeability)

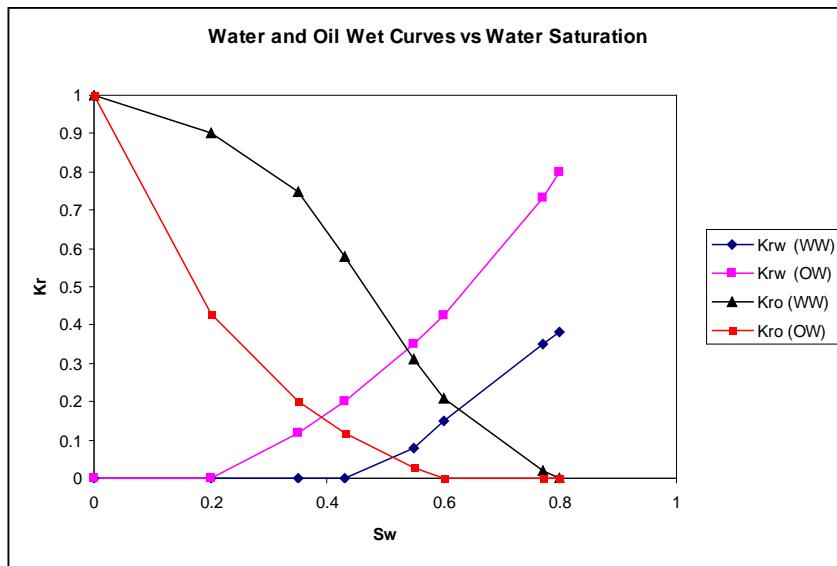


Figure 1

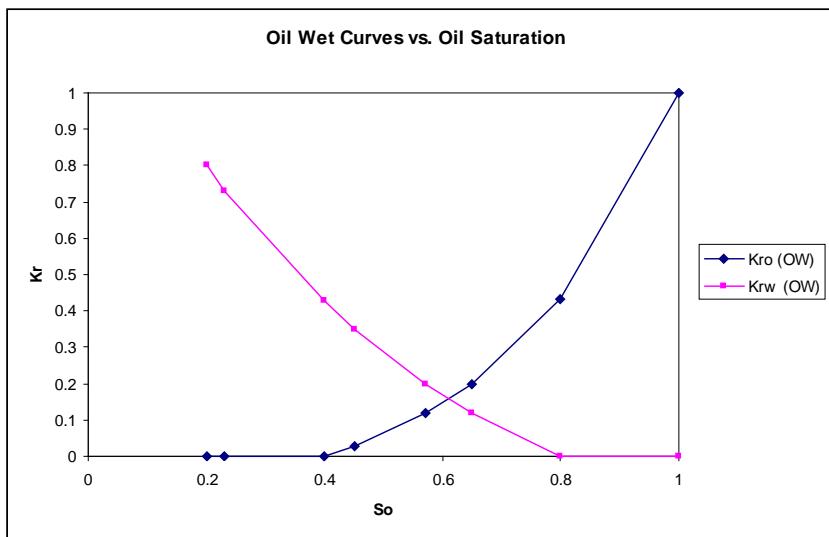
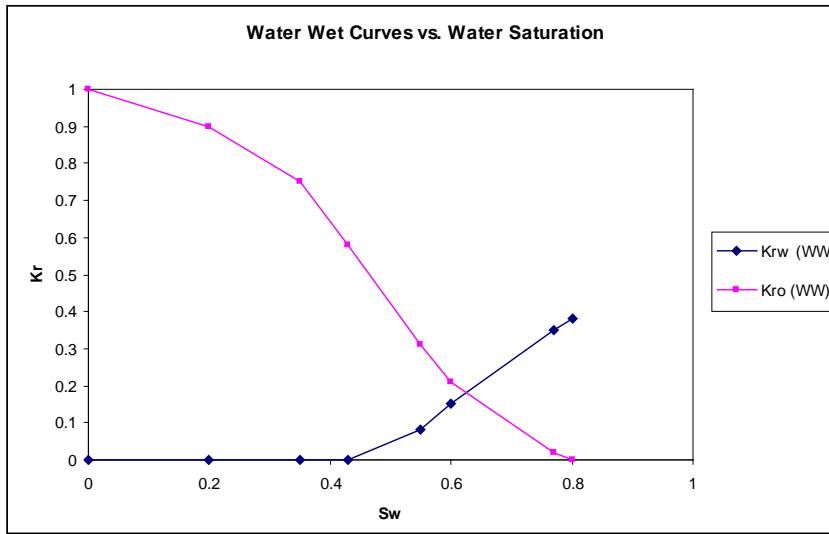


Figure 2

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). It is the curves in figure 2 which 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.



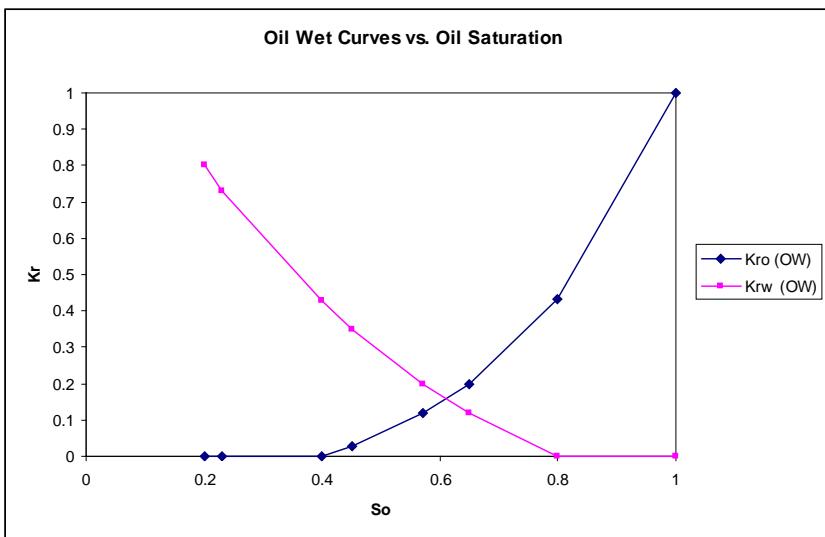


Figure 3

Example Oil Wet Rock Type Curves (Capillary Pressure)

If the capillary pressure received are in terms of P_{cow} (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.

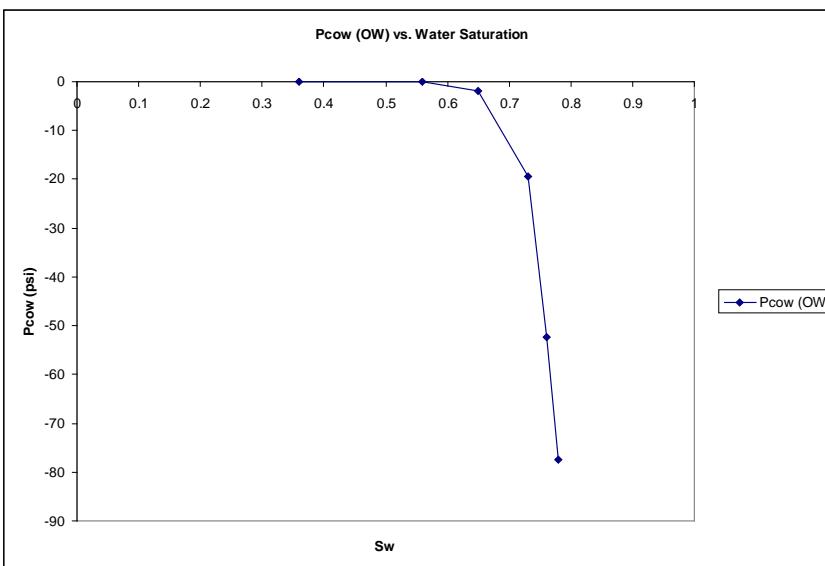
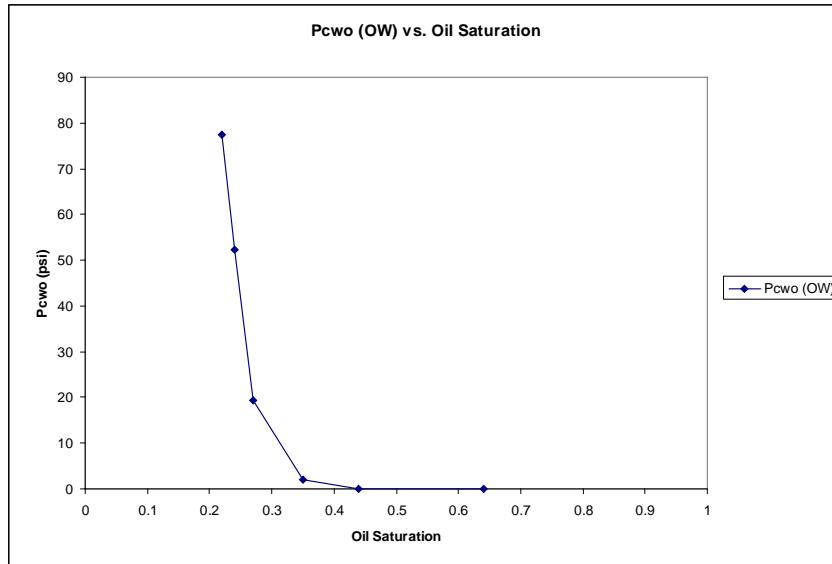


Figure 4

To use this data in IMEX, it is necessary to translate the P_{cow} 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 IMEX function. P_{cwo} hysteresis and nonwetting phase relative permeability hysteresis are modelled. It must be remembered that in the oilwet model the nonwetting liquid saturation referred to is water so HYSKRO 'somax' refers to a water residual saturation and not an oil residual saturation.

Special Keywords for Advanced Users (SHIFT)

Note:

1. These keywords greatly affect the performance of the simulation model and should only be used when the specific issues raised in the section below are seen. These keywords are not listed in the normal keyword section of the manual and in BUILDER as this would make their accidental modification possible.

The Shift Keywords:

In the numerical section the following keywords may be entered:

```
SHIFT PRESS 'p_shift'  
SHIFT SATUR 's_shift'  
SHIFT WHP 'whp_shift'
```

Normally the values of s_shift, p_shift, and whp_shift are defaulted by the simulator and do not cause problems. The shifts are used in the determination of numerical derivatives.

However under the following conditions it is possible to alter them and obtain far better numerical performance.

1. Reservoir block volumes are relatively small, but at least one flow area is very large. This might happen then an areally large block has a very small block thickness. For example a 500 x 500 x 0.01 ft block. In this example the block gross volume is only 2500.00 ft³, but the term which is used in the geometric portion of the vertical transmissibility (DX * DY)/DZ is 500x500/0.01 or 2.5E6. The normal pressure and saturation shifts in the simulator can cause a very large change in flow and hence very slow convergence. Use of smaller pressure and saturation shifts overcome this. In this case p_shift of 5.0e-6 psi and s_shift of 1.0e-6 can be tried.
2. The well head pressure being calculated is close to an atmosphere or the results are very sensitive to the well head pressure. The default whp_shift can be reduced from its default value of 0.0145 psi to a value on the order of 1.0e-5 psi.

Modelling Non Darcy Flow in Fractured Shale Gas Reservoirs

This work extends the work done with IMEX to model non Darcy flow in hydraulically fractured reservoirs using LGR grids. In that work, it was found that the fracture could be represented by a one foot thick pseudo fracture if Darcy flow and the Forchheimer number were pseudoized to predict reasonable pressure drops in the fracture. In addition, the region around the fracture needed to be modelled using a finely gridded LGR in order to reproduce the pressure drop around the fracture accurately.

In the modelling of non Darcy flow in fractured shale gas reservoirs we extend the previous work by using LGR grids to model flow into networks of fractures connected to either vertical or horizontal wells. We investigate the effect that the low matrix permeability (100 nano-Darcies) has on the models and recommend the use of a simpler Dual Permeability LGR grid. We also investigate simple approaches to add shale-gas desorption to IMEX and compare results with and without desorption affects.

LGR modelling of Fractured Shale Networks

Fracture Width Pseudoization

The basic LGR grid used to model vertical hydraulic fractures was slightly modified to create a template which could be used for a rectangular network of connected fractures in a very low permeability reservoir. In this example we assume that the fracture length is 200 feet (1/2 length = 100 feet), the fracture permeability is 4000 mD, the reservoir permeability is 0.0001 mD, and the fracture thickness is 0.001 feet.

We first refine the rectangular region of the reservoir we believe contains a rectangular network of fractures as shown below.

```
REFINE 47:55 47:55 1 INTO 21 21 1
```

We then define a fine scale LGR grid within the blocks to be refined. The fine grid is normally required to obtain accurate pressure drop modelling in the regions adjacent to the fractures. We will see that for low permeability reservoirs this is not the case.

```
DI IVAR  
101*200.0  
DI RG 47:55 47:55 1 IVAR  
33.0 25.0 16.0 8.0 6.0 4.0 3.0 2.0 1.5 1.0 1.0 1.0 1.5 2.0 3.0  
** 4.0 6.0 8.0 16.0 25.0 33.0
```

```
DJ JVAR  
101*200  
DJ RG 47:55 47:55 1 JVAR  
33.0 25.0 16.0 8.0 6.0 4.0 3.0 2.0 1.5 1.0 1.0 1.0 1.5 2.0 3.0  
** 4.0 6.0 8.0 16.0 25.0 33.0
```

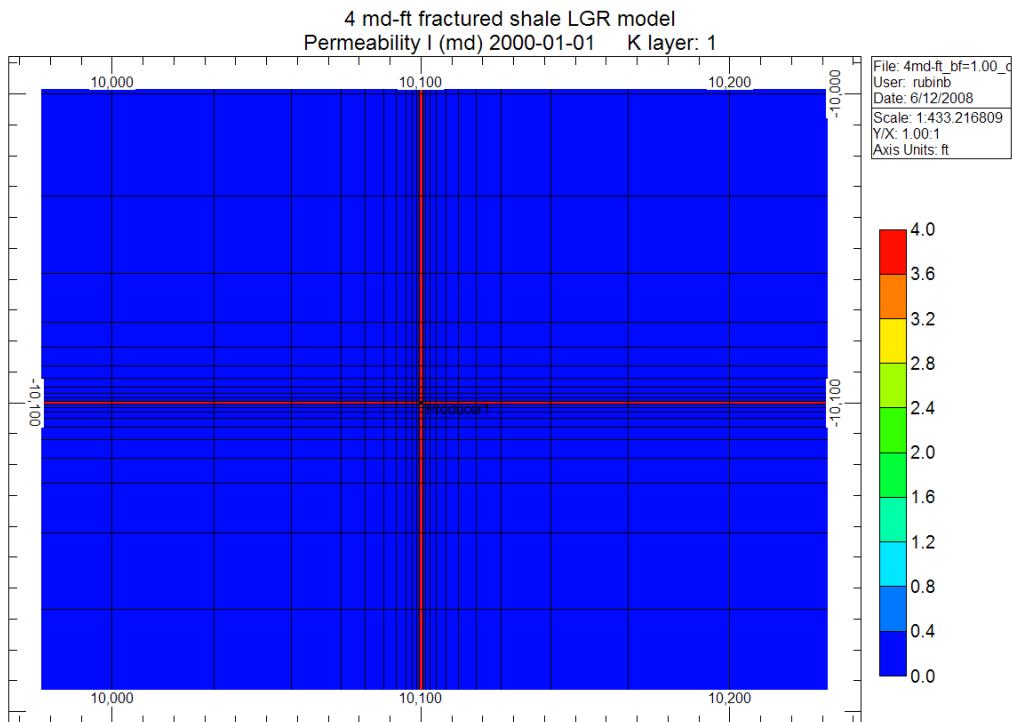
We then define the matrix permeability in the shale and the fracture permeability for fractures which extend in both areal directions. As with the simpler single plane hydraulic fracture models, the fracture permeability and fracture Forchheimer number are pseudoized in order to be able to use a computationally efficient 1 foot fracture width.

```

PERMI CON    1e-4
PERMI RG 47:55 47:55 1 ALL
10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4
10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4
10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4
10*1e-4 4.000 10*1e-4
21*4.000
10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4
10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4
10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4    10*1e-4 4.000 10*1e-4
10*1e-4 4.000 10*1e-4

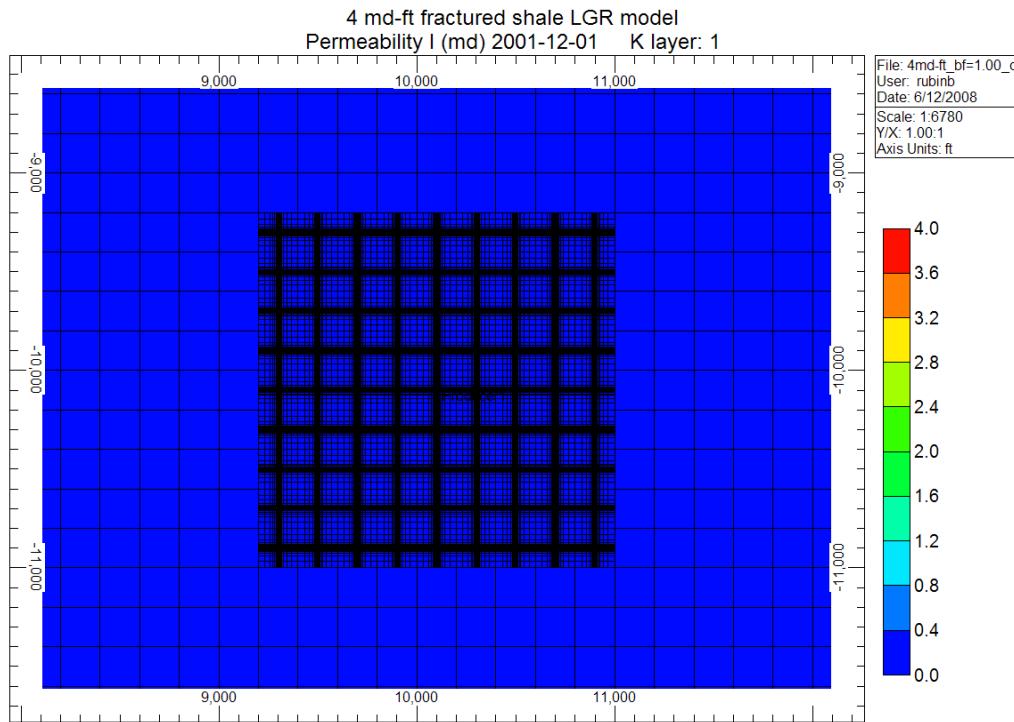
```

The fracture set up in each of these LGR's is pictured below



The fracture is indicated by the “red” colored region. The remainder of the LGR, the “blue” region is matrix.

When a rectangular region of these grids is defined as in a fractured shale model the connected LGR regions take on the appearance shown below. At the resolution presented below it is impossible to see the “red” fracture blocks as can be seen in the figure above. However, the clustering of grid blocks near the fracture makes the fracture locations obvious.

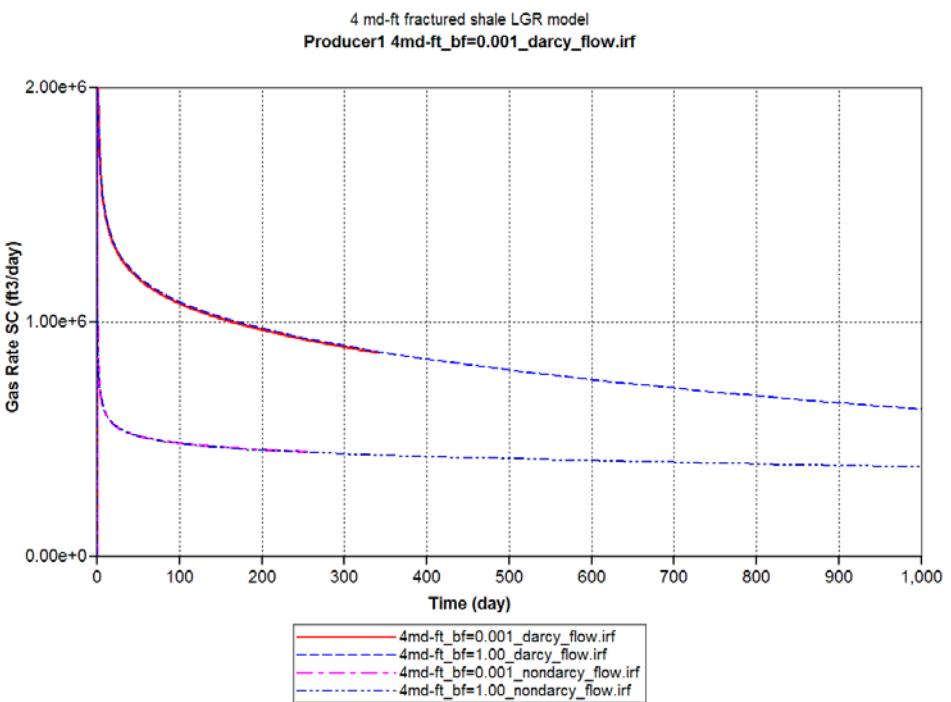


The first effort in this work was determining if the 1 foot wide fractures could be used to adequately model the flow in the extremely narrow fractures. In this model the fracture conductivity and fracture widths were assumed to be quite small. The fracture width was set at 0.001 foot (0.012 inch) and the fracture permeability was varied from 4 mD-ft to 12 mD-ft. Fracture conductivity as low as 0.5 mD-ft were used as the permeability of fractured shale and width of fracture are relatively small due to the small amounts of fracturing media used.

Models of the type above were set up to model Darcy and non Darcy flow from LGR grids using 0.001 ft. fracture thicknesses and 1 ft. thicknesses. In this pseudoization test model, we only used a vertical well configuration. An extra 8 grid blocks were used in the fine grid LGR regions so better resolution could be used capturing the flow near the 0.001 ft. wide fractures. So in the 0.001 ft. wide fracture models, the LGR consisted of a 29 x 29 block grid. In the 1 ft. wide fracture model the LGR used was the 21 x 21 block grid defined in the data above.

In the 0.001 ft. wide fracture model, neither the fracture permeability nor the Forchheimer number was pseudoized. There was no need as the “actual” fracture dimensions were used.

The results of the first 300 days of production were compared. The running time of the 0.001 ft. wide fracture model made it impractical to produce for the full 10 years of the simulation; however the first 300 days of production are enough to determine the validity of the pseudoization technique.

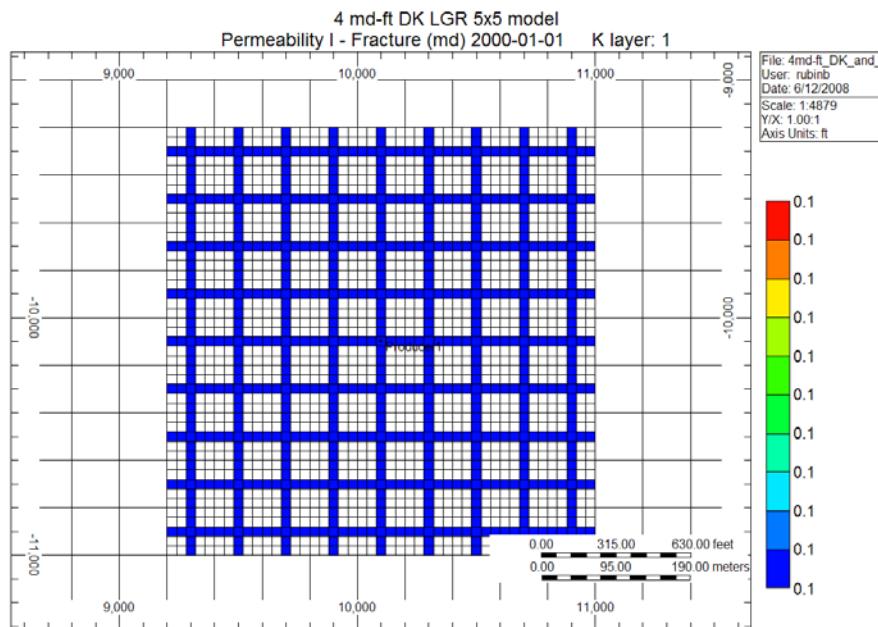


The important finding in this figure is that the non Darcy correction could be used to model a pseudo fracture 1000 times wider than the actual fracture size.

This simulation was used as validation of the 1 ft. wide fracture width pseudoization results. In all other simulations models, one foot fracture width pseudoizations will be the reference solutions when comparing simpler model fracture network models

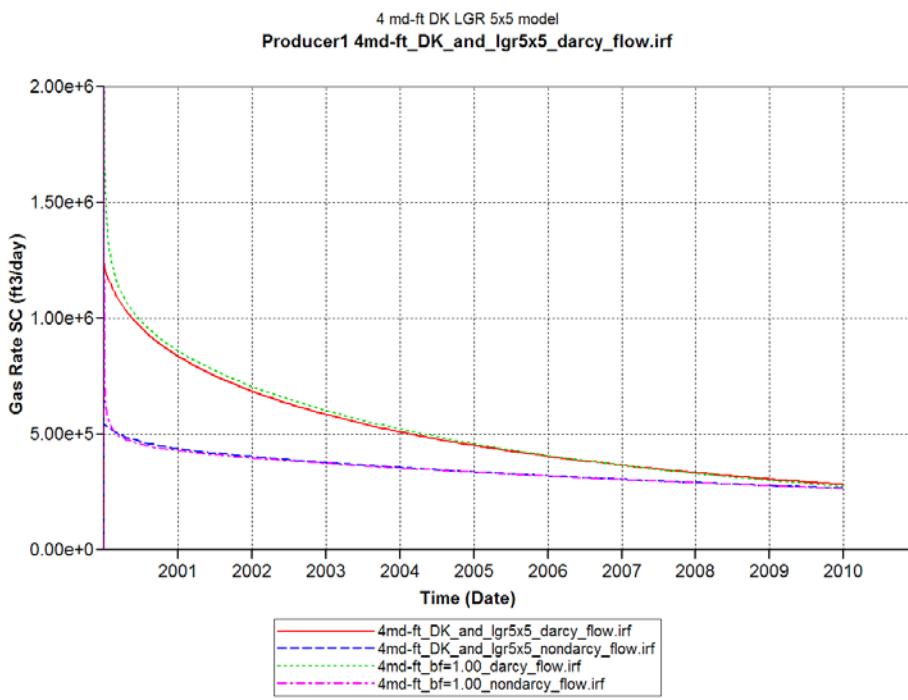
Low Permeability Shale and Dual Permeability LGR Fracture Network Models

When investigating possible simplifications to the LGR grids used in these models it became evident that the very low permeabilities in the matrix (100 nano-Darcies) allowed the use of far simpler grids. The most interesting of the grids used is a Dual Permeability representation within an LGR grid (DK-LGR). Instead of using a 1 foot wide grid cell to model a fracture, a dual permeability fracture “fairway” was used. This grid modelled the fracture flow path to the well as a relatively low permeability 40 ft. wide region (fairway). The non Darcy flow Forchheimer correction factor was used in this model to correct a 0.001 ft wide fracture to a represent the non Darcy flow in a 40 ft wide fairway. The Forchheimer correction value was over 13,000.0 to achieve this. An example of the grid is shown below.



Only the fractured part of the dual permeability grid is shown for clarity. A dual permeability model was used with equally sized 5 x 5 LGR refinements. The fracture permeability and porosity were chosen to represent the effective permeability and porosity of a 0.001 ft wide fracture in a 200 foot block. Note that in this grid the fracture porosity is zero in the unfractured portion of the grid and in the region not designated as a fracture fairway. In these regions only matrix porosity is defined. Additionally in the fracture fairway, no matrix porosity is defined. The matrix fracture transfer terms were calculated assuming a horizontal fracture spacing of 200 ft and assuming no horizontal fractures. However since there are no co-located fracture and matrix blocks, the fracture spacing (and matrix-fracture transfer term) is not used. The CMG grid module makes the appropriate connections between matrix-only and fracture-only regions.

For the same 4 mD-ft fracture network as was previously used, we compare the DK-LGR model with the standard LGR model for both Darcy and non Darcy flow. The results are presented below.



An excellent result is observed using dual permeability 5 x 5 LGR grids which produces similar results to far more refined 21 x 21 single porosity LGR grids.

We will show later in this work that these excellent results cannot be duplicated with more permeable matrices, when the matrices are more permeable good pressure drop resolution becomes critical, but we have been able to produce good matches with matrix permeabilities as high as 0.01 mD.

We have run the DK-LGR model on several variants of the basic fractured shale reservoir.

- The fracture conductivity is increased to 12 mD-ft
- The fracture conductivity is decreased to 0.5 mD-ft
- The fracture spacing is decreased from 200 ft to 25 ft
- Horizontal well in the 4 mD-ft fracture conductivity model
- Horizontal well in the 12 mD-ft fracture conductivity model
- Horizontal well in the 25 ft fracture spacing model
- Three dimensional 4 mD-ft model (10 layers – only perfed in layer 2)
- 12 mD-ft model in a 0.01 mD matrix permeability shale

In all of these models the DK LGR model produces results virtually identical to the standard LGR based models for both Darcy and non Darcy flow.

Dual Permeability LGR GRID Permeability and Porosity Pseudoization

The distance between vertical fractures in both the X and Y directions is 200 ft. The 200 ft x 200 ft blocks are broken in to 25, 40 ft x 40 ft blocks in an LGR. The LGR blocks in the row j=3 and column i=3 are set to have fracture porosity and so represent the fracture fairway. The LGR blocks in the row j=3 and column i=3 also have zero matrix porosity. In each of these cells, a single 0.001 ft wide fracture is assumed to exist which runs for the entire length of the 40 ft long LGR block, hence the average fracture porosity (assuming the porosity in the 0.001 ft wide fracture is 100%) is $(0.001 \times 40) / (40 \times 40)$.

This is 0.000025.

The permeability of the fracture is adjusted so that $K_{frac} \times Area_{frac} = K_{fraceff} \times Area_{lgrblock}$.

In the case with a 4 mD-ft fracture this gives us an effective fracture permeability of $(0.001/40) \times 4000$ mD which equals 0.10 mD

Non Darcy Correction Factor Set Up (Single Porosity LGR Grid)

For the single porosity reference grids (21 x 21 block LGR). The non Darcy correction set up is virtually identical to the set up for a hydraulically fractured well. The only difference is the two dimensional nature of the fracture. This is reflected in how the NDARCYCOR parameter is defined.

The value used in the NDARCYCOR factor is $(K_{frac}/K_{fraceff})^{(2-1.1045)}$ or 485.847 where k_{frac} equals 4000 mD and $K_{fraceff}$ equals 4 mD. $K_{fraceff}$ is 4 mD due to the use of a 1 foot wide fracture to model a 0.001 ft wide fracture.

The value 1.1045 in the NDARCYCOR equation is the exponent which multiples permeability in the Beta factor term of the Forchheimer number. See the tutorial section concerning non Darcy flow in fractures for a discussion of the non Darcy correction factor.

In this example, the Beta A-term = 27.3e9 and the B-term = 1.1045 were taken from an example in the textbook by Dake (Fundamentals of Reservoir Engineering).

```
NONDARCY GENERAL 0.5
**Beta A Beta B
 27.3e9    1.1045 0
 27.3e9    1.1045 0
 27.3e9    1.1045 0

NDARCYCOR CON      0
NDARCYCOR RG 47:55 47:55 1 ALL
10*0 485.847 10*0    10*0 485.847 10*0    10*0 485.847 10*0    10*0
485.847 10*0
10*0 485.847 10*0    10*0 485.847 10*0    10*0 485.847 10*0    10*0
485.847 10*0
10*0 485.847 10*0    10*0 485.847 10*0
21*485.847
10*0 485.847 10*0    10*0 485.847 10*0    10*0 485.847 10*0    10*0
485.847 10*0
10*0 485.847 10*0    10*0 485.847 10*0    10*0 485.847 10*0    10*0
485.847 10*0
10*0 485.847 10*0    10*0 485.847 10*0    10*0 485.847 10*0    10*0
485.847 10*0
10*0 485.847 10*0    10*0 485.847 10*0    10*0 485.847 10*0    10*0
```

Fracture Porosity and Non Darcy Correction Factor Set Up (Dual Perm. LGR Grid)

In the Dual Permeability grid, we set up the non Darcy correction factor in all fracture blocks which are defined by the keyword POR FRACTURE.

How the values of fracture permeability and porosity were determined were discussed in the previous section.

Initially define matrix (shale) porosity everywhere at 3%

```
POR MATRIX CON 0.03
```

Initially define no fractures in the Dual Permeability model

```
POR FRACTURE CON 0.0
```

Define fractures in every block in the hydraulically fractured region

```
MOD 47:55 47:55 1 = 0.000025
```

The fractures in the LGR region are limited to fracture “fairways”

```
POR FRACTURE RG 47:55 47:55 1 ALL
0 0 0.000025 0 0
0 0 0.000025 0 0
0.000025 0.000025 0.000025 0.000025 0.000025
0 0 0.000025 0 0
0 0 0.000025 0 0
```

The matrix porosity is limited to the non fractured region in the refined area.

```
POR MATRIX RG 47:55 47:55 ALL
0.03 0.03 0.0000 0.03 0.03
0.03 0.03 0.0000 0.03 0.03
0.00 0.00 0.0000 0.00 0.00
0.03 0.03 0.0000 0.03 0.03
0.03 0.03 0.0000 0.03 0.03
```

The volume lost by nulling out the matrix which is co-located with the fracture fairway is made up by increasing the volume of the remaining matrix blocks in the refined region:

```
VOLMOD MATRIX RG 47:55 47:55 1 CON 1.5626 ** = 1/(4/5x4/5)
```

In the statement above, the 16 remaining matrix blocks are given the volume of the original 25.

The non Darcy correction factor is set to zero in the matrix and to a non-zero value in the fracture as shown below.

```
NONDARCY GENERAL 0.5
**Beta A Beta B
27.3e9 1.1045 0
27.3e9 1.1045 0
27.3e9 1.1045 0
NDARCYCOR MATRIX CON 0
NDARCYCOR FRACTURE CON 13217.36
```

The value of the non Darcy correction is based on the ratio of the actual fracture permeability (4000 mD) to the effective Dual Permeability fracture permeability (0.10 mD) and equals $(4000/0.10)^{1.1045}$ or 13,217.36. How the Dual Permeability model fracture permeability of 0.1 mD was arrived at has been previously discussed.

The value 1.1045 in the NDARCYCOR equation is the exponent which multiples permeability in the Beta factor term of the Forchheimer number. See the tutorial section concerning non Darcy flow in fractures for a discussion of the non Darcy correction factor.

Use of the Dual Permeability Grids in models with Higher Shale Permeabilities

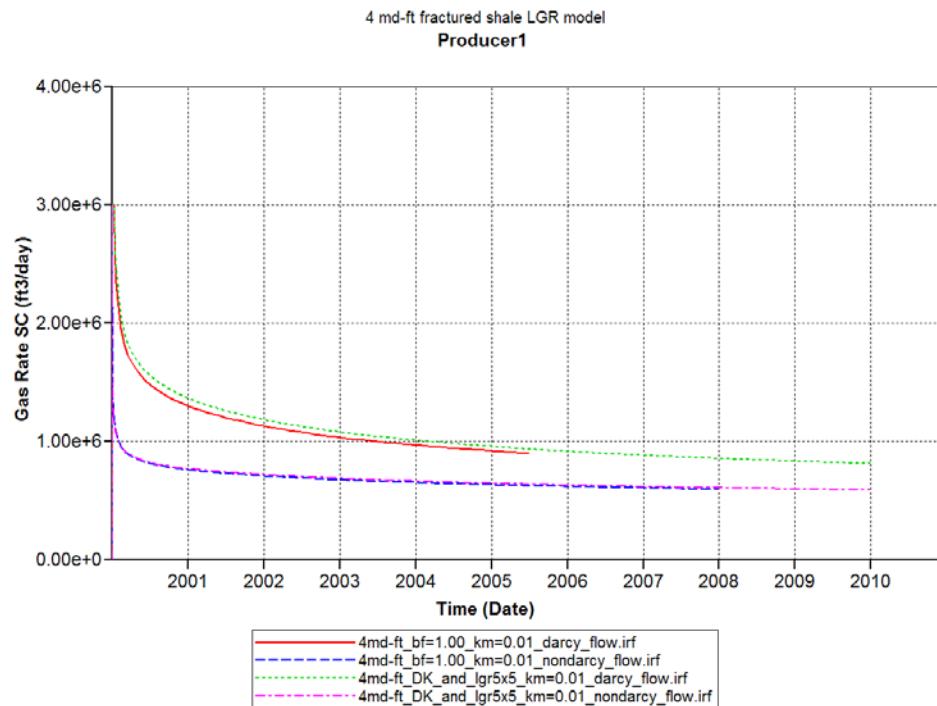
The DK-LGR model is not significantly different from any other coarsely defined grid we could have used to model this process. What makes the use of coarse grids possible in these shale models is the very low permeability and hence low pressure drop in the matrix. In order to determine the usability of this option we investigated how the match changed when the matrix permeability was increased from 0.0001 mD to 0.01 mD.

For this study we investigated both the 4 mD-ft and 12 mD-ft conductivity fracture cases.

Note that the 4 mD-ft base case models were not run to completion but were run far enough to indicate how well the DK-LGR models match the more finely gridded single porosity LGR models

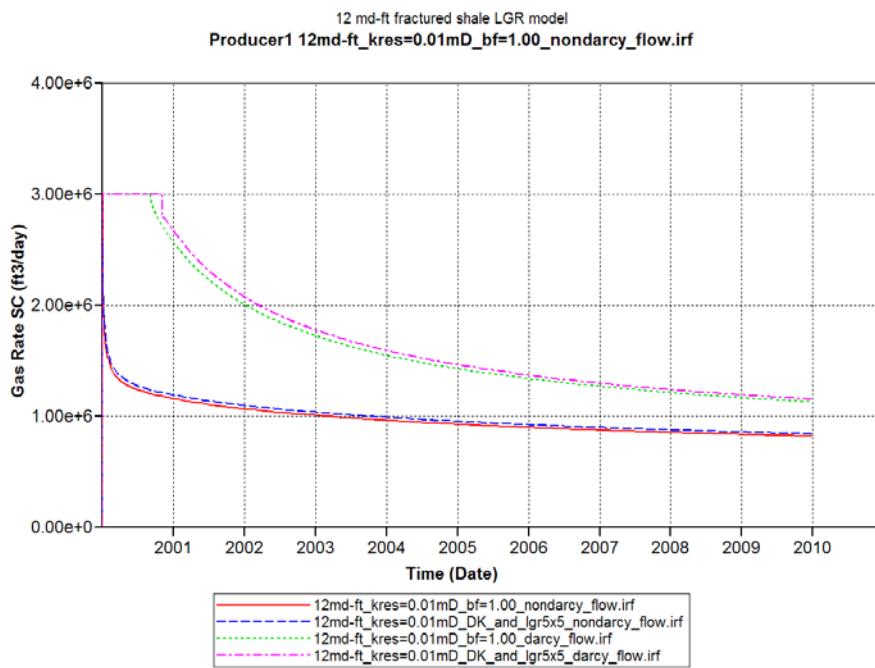
Vertical Well, 4 mD-ft Fracture Conductivity, 200 foot fracture spacing, Kshale=0.01 mD

21x21 LGR 1 foot wide fracture pseudoization compared to dual perm 5x5 LGR



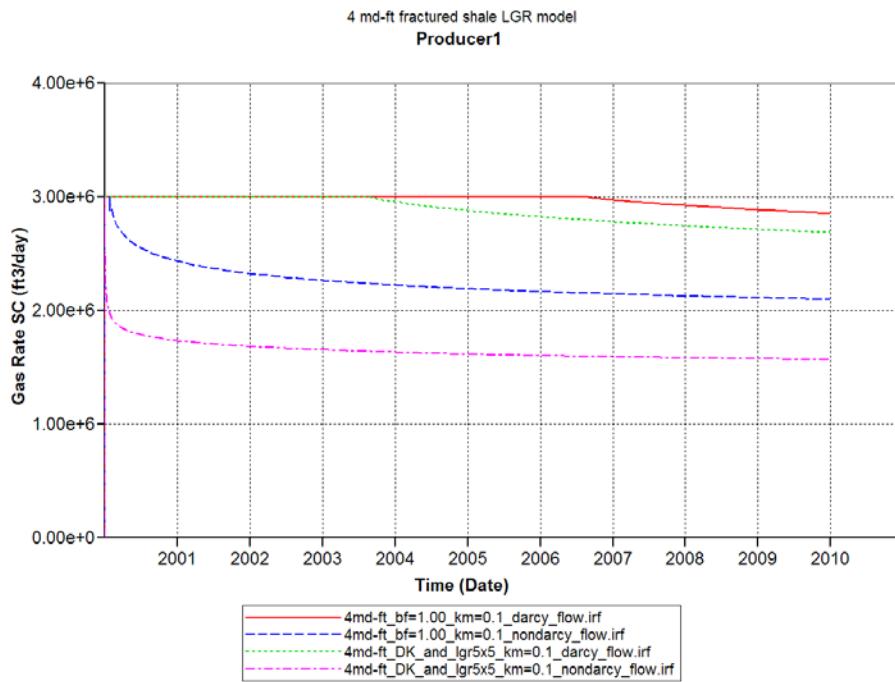
Vertical Well, 12 mD-ft Fracture Conductivity, 200 foot fracture spacing. Kshale=0.01 mD

21x21 LGR 1 foot wide fracture pseudoization compared to dual perm 5x5 LGR



The above figures indicate that this simple Dual Permeability representation is valid for shale permeabilities as high as 0.01 mD.

When the 4 mD-ft model was run again with a shale permeability of 0.10 mD, the results of the simplified DK-LGR model and the detailed LGR model differed significantly.



The figure above models a vertical well in 0.10 mD shale with a 4 mD-ft conductivity fracture network. The 5x5 block LGR resolution does not properly model gas flow in the matrix.

Both the Darcy flow and non Darcy flow models exhibit the same behavior which is due to the poor representation of the pressure distribution in the more permeable matrix.

This result just emphasizes the importance of proper discretization around the fractures to be able to properly model the rapidly changing pressure in and near the fractures.

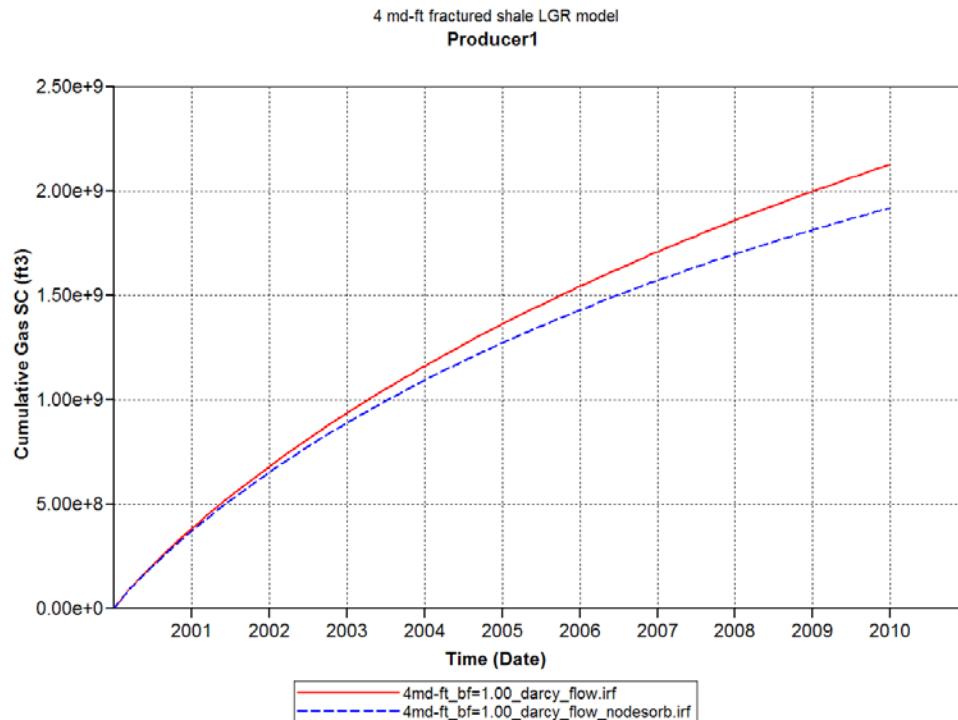
From these results we conclude that a simplified 5x5 grid is usable for shale permeabilities of 0.01 mD and less which should be sufficiently high for most shale gas reservoirs.

Gas Adsorption Effects

All of the above studies assume that the primary source of gas for a fractured shale reservoir is from evolved gas flowing in the matrix. They do not incorporate desorbed gas from organics in the shale to the overall production.

However, it is relatively easy to model gas desorption from a shale gas reservoir in a black oil model using a technique developed by Seidle and Arri (Pet. Soc. CIM/SPE 90-118). This procedure uses a black oil model's solution gas ratio to replicate a Langmuir Isotherm and was used in IMEX to include gas desorption effects in the base model.

The 4 mD-ft vertical well base model was rerun using the single porosity LGR grid with desorption included for the Darcy flow case and compared to the case without desorption. The results are presented below.



The Isotherm parameters used for the simulation above are derived for 10% total organic Carbon New Albany Shale (from Zuber et al, SPE 77469):

Langmuir Volume constant = 104.2 SCF/ton

Langmuir Pressure = 142.5 psia

Shale Density = 2.4 gm/cm³

Using the Langmuir parameters listed above the run with desorption produces about 10% more gas over the 10 years simulated.

Setting up the Black Oil Model to Simulate Desorption

The use of a black oil model to simulate gas desorption assumes diffusion is fast compared to desorption and so does not limit desorption.

We need to use a black oil model rather than a gas water model to simulate this process. As it is necessary to define a small amount of oil in the model (Som); this immobile pseudo oil represents the organic material which is the source of desorbed gas.

We recommend Som = 0.0001, a value small enough so that it is unnecessary to adjust initial saturation or porosity conditions.

In each PVT region, we must assume constant matrix porosity as we need to calculate a Langmuir equation which uses pseudo oil properties and constant matrix porosity.

In the PVT table, pseudo oil properties need to be set up to model an immobile oil phase which produces gas in response to pressure changes.

Oil Viscosity = 1.0E+12 cP (makes oil immobile)

Oil FVF = 1.0E-03 Bbl/STB

Any Reasonable Oil Density can be used (say 40 lb/ft³)

Co = Cvo = 0.0

Water and gas properties can be entered normally

The high oil viscosity ensures that the oil does not flow.

The oil FVF of 1.0E-03 need not be realistic; the use of a small value allows Som to be made small enough to have no affect on initial conditions.

The Langmuir isotherm is represented by a solution gas ratio (Rs) using an equation of the form

$$Rs = 5.615 * Lvol * Bo / (Som * Porm) * P/Plang / (1+P/Plang)$$

Rs in Field units (SCF/Bbl)

Plang = Langmuir Pressure (psi)

Lvol = Langmuir Volume (SCF gas/ CF Coal)

Som = 1e-4

Porm = Porosity of matrix

It is necessary to ensure the PVT table has enough resolution to capture the curvature of the Langmuir Equation.

Use DEPTH_AVE initialization and note that Bubble Point pressure is the desorption pressure.

The Gas Liquid relative permeability table endpoint (Slcon) needs to be increased to Swc +0.0001 to account for the pseudo oil. P_{cog} = 0.0.

The oil relative permeability in the black oil tables should always remain small. A straight line curve varying from 0.0 to 1.0E-04 is sufficient

Assuming Porm	=	0.03;
Lvol	=	104.2 SCF/ton;
Density of shale matrix	=	2.4 gm/cm ³ ;
Langmuir Pressure	=	142.5 psia;
C	=	0.031185 converts the units of Lvol * Shale Density ((SCF/ton)*(gm/cm ³)) to SCF/CF.

The pseudo Rs (SCF/Bbl)

$$= [5.615 * C * 104.2 * 2.4 * 1.00E-03 / (1.00E-04*0.03)] * (P/Plang) / (1+P/Plang)$$

or

$$Rs = 14596.6 * (P/Plang) / (1+P/Plang)$$

The resulting Rs vs. Pressure table is presented below.

P	Rs
14.7	1364.94
200	8523.56
400	10762.47
800	12389.68
1200	13047.24
2000	13625.76
4000	14094.48

The WATER_FIX Keyword

In the numerical section the following keyword may be entered:

WATER_FIX 2

Normally the IMEX formulation and variable switching algorithms are robust enough to handle even the most difficult problems.

However under certain circumstances, when oil attempts to enter a block which is completely empty of any hydrocarbon, numerical issues may occur. These can be identified as periods of a simulation where timestep sizes fails to build up significantly and where timestep repeats are common.

Using *WPRN *ITER *MATRIX to turn on more timestep convergence information, these periods are fairly obvious as times when oil and gas residuals fail to reduce in blocks at the border/interface of hydrocarbon and completely water filled blocks. The maximum residuals of oil and gas fail to converge at these border blocks and appear to be stuck at values above the convergence tolerances. Occasionally the message “Newton Method is Stuck” will appear.

We have developed an optional algorithm to improve our formulation under these circumstances. The algorithm may be started by specifying *WATER_FIX 2 in the numerical data section or by adding the –water_fix 2 command line option.

Additional water_fix options have been added to further improve issues which occur when the variable update sends primary outside expected ranges (such as saturation > 1.0).

These may be of use and can be enabled by specifying “water_fix 4”

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.

Relative Permeability and Capillary Pressure Hysteresis in Oil-Water Systems

IMEX can model both relative permeability and capillary pressure hysteresis in oil-water systems using an entirely consistent model. The model includes the hysteresis of oil-water capillary pressure P_{cow} , oil relative permeability in water K_{row} , and water relative permeability K_{rw} . The model can represent both hysteresis due to oil trapping and hysteresis due to contact angle (wettability) change.

Modelling Trapped Oil Hysteresis

Fig.1 shows the input curves for modeling a water-wet rock type that traps oil. The imbibition capillary pressure curve is provided with an end-point, S_{ormax} . S_{ormax} is the maximum residual oil saturation which stops the boundary imbibition process of both capillary pressure and oil relative permeability. S_{ormax} must be greater than residual oil saturation S_{orw} .

For the imbibition process, the end point of the imbibition scanning curve is the imbibition residual oil saturation, S_{orh} . S_{orh} lies between S_{orw} and S_{ormax} , and is a function of S_{ormax} , S_{orw} , the maximum oil saturation $1-S_{wcon}$, and the historical maximum attained oil saturation S_{ohy} .

Imbibition scanning begins at the point where the rock changes from drainage to imbibition. The water saturation at the turning point is $1-S_{ohy}$. The function for determining S_{orh} from S_{ohy} is the oil-trapping relationship. The difference, $S_{orh} - S_{orw}$, is the trapped oil saturation. See Fig. 3.

In IMEX, the oil-trapping relationship is assigned by keyword *OILTRAP. Land's (1968) correlation, both linear- and table-defined relationships, can be assigned to model the oil trapping relationship. Eq.1 gives the expression for Land's correlation, which is the default relationship. Fig.2 graphically shows the Land correlation together with the linear relationship.

$$S_{orh} = S_{orw} + \frac{1}{1/(S_{ormax} - S_{orw}) - 1/(1 - S_{wcon} - S_{orw}) + 1/(S_{ohy} - S_{orw})} \quad (\text{Eq.1})$$

S_{ormax} and K_{row} Hysteresis

In the *SWT table, if the boundary imbibition capillary pressure P_{cow} is given as in Fig.1, the curve provides the end-point $1-S_{ormax}$. To define S_{ormax} in the *SWT table, the input boundary imbibition P_{cow} values after $S_w = 1-S_{ormax}$ *must* be kept constant.

S_{ormax} is also given by the *SWTI table, which is used to enter boundary imbibition oil relative permeability K_{row} and water relative permeability K_{rw} . The water saturation of the *SWTI table begins from the connate water saturation S_{wcon} and ends at $1-S_{ormax}$.

If S_{ormax} is defined to be greater than S_{orw} , the K_{row} hysteresis will be invoked to honor S_{ormax} . Keywords *HYSKRO and *HYSKRW can be used to assign hysteresis methods for relative permeabilities.

If imbibition K_{row} is table-defined, the Killough (1976) relative-permeability-interpolation method (Eq.2 & 3) and the Killough saturation-interpolation method (Eq.4) are available. The default is relative-permeability-interpolation if *HYSKRO is omitted. On the other hand, if imbibition K_{row} is not defined by the *SWTI table, Carlson's (1981) K_{row} hysteresis method (Eq.5 & 6) will be used as the default. The following equations are the available methods for

K_{rw} hysteresis. In all of the equations in this tutorial, Im , Dr and $Scan$ are used to denote the imbibition, drainage and scanning respectively.

Killough's permeability-interpolation:

$$K_{rw}^{Scan}(S_o) = K_{rw}^{Im}(S_o^{\text{norm}}) \frac{K_{rw}^{Dr}(S_{ohy})}{K_{rw}^{Dr}(1-S_{wcon})} \quad (\text{Eq.2})$$

where the normalized oil saturation S_o^{norm} is calculated by:

$$S_o^{\text{norm}} = S_{orh} + \frac{(S_o - S_{orh})(1 - S_{wcon} - S_{ormax})}{S_{ohy} - S_{orh}} \quad (\text{Eq.3})$$

Killough's saturation-interpolation:

$$K_{rw}^{Scan}(S_o) = K_{rw}^{Dr}(S_{ohy}) \left(\frac{S_o - S_{orh}}{S_{ohy} - S_{orh}} \right)^{hyexo} \quad (\text{Eq.4})$$

where the exponent $hyexo$ is defined by sub-keyword *HYEXO *KILLOUGH.

Carlson K_{rw} hysteresis:

$$K_{rw}^{Scan}(S_w) = K_{rw}^{\text{Drainage}}(S_w(\text{shifted})) \quad (\text{Eq. 5})$$

where:

$$S_w(\text{shifted}) = 1 - S_{orw} - (1 - S_w - S_{orh}) \frac{(S_{ohy} - S_{orw})}{(S_{ohy} - S_{orh})} \quad (\text{Eq.6})$$

K_{rw} Hysteresis

Water relative permeability hysteresis is controlled by either the original Killough (1976) K_{rw} method or its modification. The scanning K_{rw} at water saturation S_w by Killough is:

$$K_{rw}^{Scan}(S_w) = K_{rw}^{Dr}(1 - S_{ohy}) + \frac{K_{rw}^{Im}(1 - S_o^{\text{norm}})}{K_{rw}^{Im}(1 - S_{ormax})} [K_{rw}^{Scan}(1 - S_{orh}) - K_{rw}^{Dr}(1 - S_{ohy})] \quad (\text{Eq.7})$$

In Eq.7, the maximum value of K_{rw} at its scanning end, $K_{rw}^{Scan}(1 - S_{orh})$, is calculated by:

$$K_{rw}^{Scan}(1 - S_{orh}) = K_{rw}^{Dr}(1 - S_{orh}) + [K_{rw}^{Im}(1 - S_{ormax}) - K_{rw}^{Dr}(1 - S_{ormax})] \left(\frac{S_{orh} - S_{orw}}{S_{ormax} - S_{orw}} \right)^{hyexw} \quad (\text{Eq.8})$$

Eq.7 may result in the scanning curve crossing the drainage K_{rw} curve. The modified Killough method is introduced to improve this situation:

$$K_{rw}^{Scan}(S_w) = K_{rw}^{Dr}(S_w) + [K_{rw}^{Scan}(1 - S_{orh}) - K_{rw}^{Dr}(1 - S_{orh})] \left(\frac{K_{rw}^{Im}(1 - S_o^{\text{norm}}) - K_{rw}^{Dr}(1 - S_o^{\text{norm}})}{K_{rw}^{Im}(1 - S_{ormax}) - K_{rw}^{Dr}(1 - S_{ormax})} \right) \quad (\text{Eq.9})$$

With the Killough or modified Killough method, if water saturation exceeds $S_w = 1 - S_{orh}$, K_{rw} will scan the Locus Maximum curve (see Fig. 3). The Locus Maximum is determined by Eq.8 by substituting S_w for $1 - S_{orh}$.

Fig.3 displays the P_{cow} , K_{rw} and K_{rw} scanning curves for hysteresis with trapped oil.

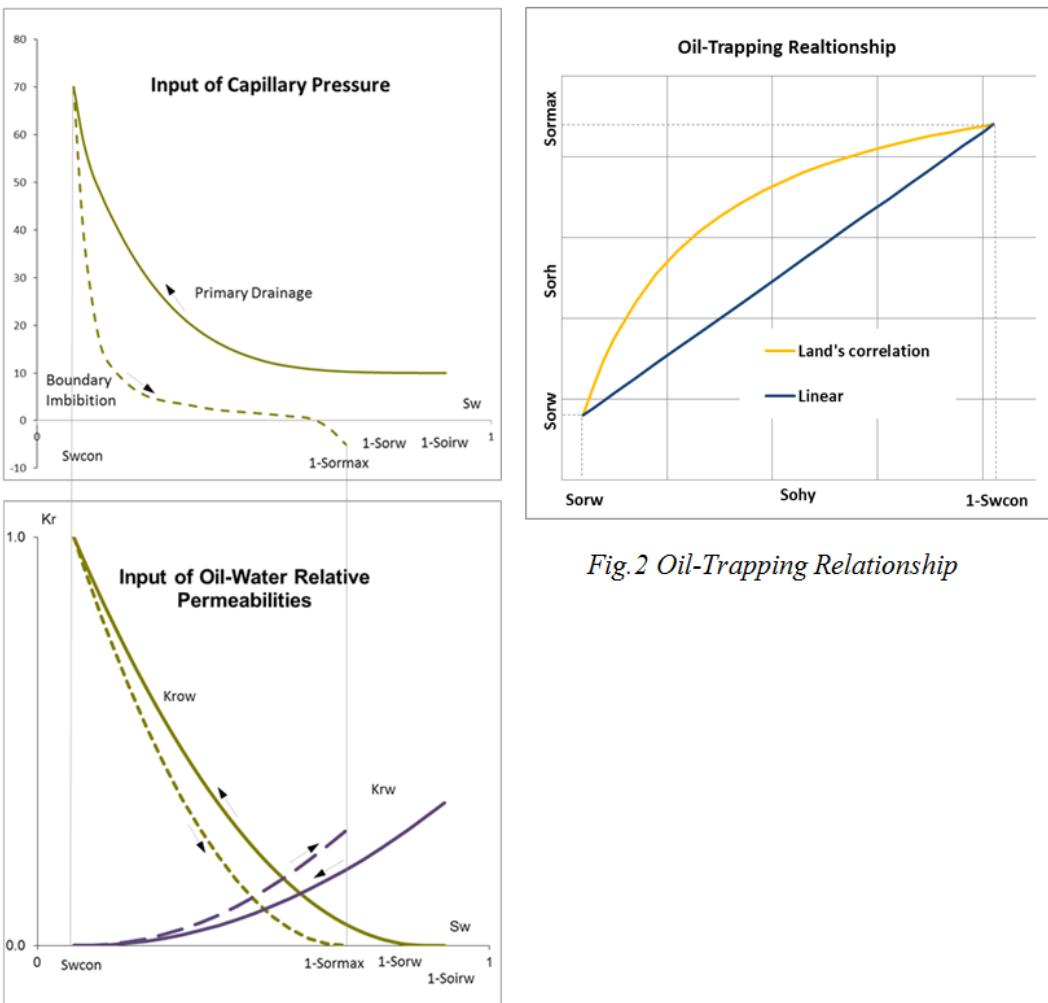


Fig.1 Inputs for Hysteresis with Trapped Oil

P_{cow} Hysteresis

The P_{cow} hysteresis processes are illustrated by Fig.3. Initially, all reservoir blocks are on the drainage curve. Suppose the imbibition of a block begins at $S_w=1-S_{ohy}$. The S_{ohy} saturation will be recorded as the historical maximum attained oil saturation. The imbibition process scans the curve (d) towards its end-point $1-S_{oh}$.

The imbibition process may reverse at any saturation (i.e. if water saturation decreases) and begin the drainage (imbibition to drainage) process, scanning curve (e) or curve (f) in Fig.3. Both curves (e) and (f) will drain back to the water saturation of $1-S_{ohy}$, where the original imbibition process began. The drainage process on curve (e) or (f) may continue and pass through the point $1-S_{ohy}$ to lower water saturation. In that case, the S_{ohy} of the block increases. On the other hand, if the process on (e) or (f) reverses before the $1-S_{ohy}$ point is reached (i.e. imbibition begins before reaching $1-S_{ohy}$), the same curve (e) or (f) is scanned back as shown by the arrows in Fig.3.

Fig.2 Oil-Trapping Relationship

The calculation of the scanning curves uses the Killough (1976) method. The drainage to imbibition scanning curve (d), and imbibition to drainage scanning curve (e), are calculated by:

$$P_{cow}(d) = f_i \times P_{cow}(b) + (1 - f_i) \times P_{cow}(a) \quad (\text{Eq.10})$$

$$P_{cow}(e) = f_d \times P_{cow}(a) + (1 - f_d) \times P_{cow}(b) \quad (\text{Eq.11})$$

where f_i and f_d are given by:

$$f_i = \left(1 + \frac{\text{epspc}}{S_{ohy} - S_{orh}} \right) / \left(1 + \frac{\text{epspc}}{S_w - (1 - S_{ohy})} \right) \quad (\text{Eq.12})$$

$$f_d = \left(1 + \frac{\text{epspc}}{S_{ohy} - S_{orh}} \right) / \left(1 + \frac{\text{epspc}}{(1 - S_{orh}) - S_w} \right) \quad (\text{Eq.13})$$

In the above equations, curvature parameter *epspc* is defined by keyword *EPSPC or is set by default to a value of 0.1.

If the imbibition process, on curve (d), is reversed at $S_{w,rev}$, another imbibition to drainage curve (f) will be scanned towards $1 - S_{ohy}$. P_{cow} along scanning curve (f) is calculated using curve (e) and (d):

$$P_{cow}(f) = f_d \times P_{cow}(e) + (1 - f_d) \times P_{cow}(d) \quad (\text{Eq.14})$$

where:

$$f_d = \left(1 + \frac{\text{epspc}}{S_{w,rev} - (1 - S_{ohy})} \right) / \left(1 + \frac{\text{epspc}}{S_{w,rev} - S_w} \right) \quad (\text{Eq.15})$$

Damped P_{cow} Scanning Curves

By default, the P_{cow} value at $1 - S_{orh}$ of the scanning curves is treated as the constant that is the P_{cow} value at $1 - S_{orh}$ of the boundary imbibition curve (see Fig.4, default cycle). When the historical maximum attained oil, S_{ohy} , of a block is close enough to S_{orw} , the created P_{cow} scanning curves will tend towards a vertical line between $P_{cow}(a)$ and $P_{cow}(b)$ at $1 - S_{orw}$. This discontinuity could be a source of numerical instability, depending on the extent of the difference between $P_{cow}(a)$ and $P_{cow}(b)$.

Optional keyword *DAMP_PCOW_TROIL in Rock-Fluid data can be used to reduce the vertical extent of the P_{cow} cycle. As shown by Fig.4, the “damped” P_{cow} is formed by reducing the default P_{cow} with a constant ratio and placing the $P_{cow}(1 - S_{orh})$ onto the straight line between $P_{cow}(b, 1 - S_{orh})$ and $P_{cow}(a, 1 - S_{orh})$.

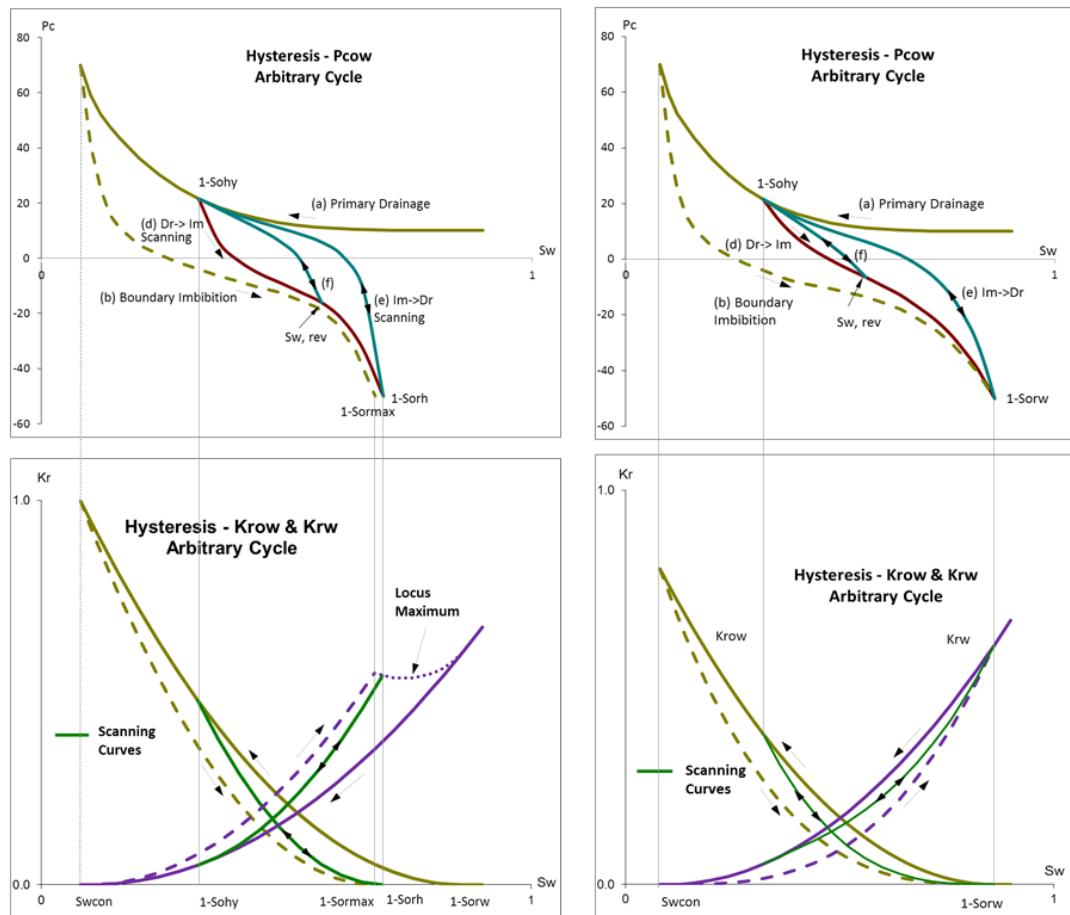


Fig.3 Hysteresis with Oil trapping

Fig.5 Hysteresis without Oil Trapping (contact angle)

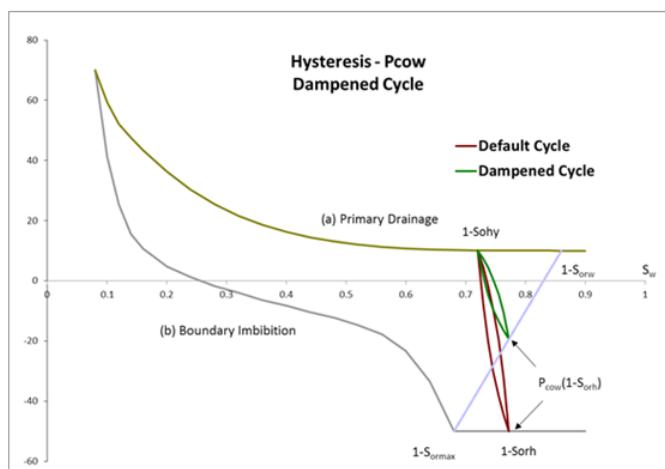


Fig.4 Damped P_{cow} Curves

S_{ormax} for Grid Blocks and Well Layers

Maximum residual oil saturation can be assigned to each grid block by using keyword *SORMAX. *SORMAX can be entered in Rock-Fluid data and the recurrent data section. Note that the assignment of *SORMAX to a block is effective only when the rock type traps oil.

*SORMAX is only used to adjust an existing value of S_{ormax} (from the rock table). It cannot be used to assign a S_{ormax} to a table that does not already have a defined value of S_{ormax}.

For well layers, S_{ormax} can also be applied as an end point if the rock type of the well has trapped oil hysteresis activated. For a well layer, sub-keyword *SORMAX can be used in the *KRPERF keyword, for example:

```
*KRPERF  
10      10      1:20  *SORMAX  0.35
```

Note that the *SORMAX defined in *KRPERF only effects K_{rw} hysteresis. Capillary pressure is not included in the well equations.

Model Hysteresis without Oil Trapping

Without oil trapping, oil-water hysteresis in relative permeability and capillary pressure may occur due to change of contact angle on pore surfaces due to water-advancing or water-receding processes. Fig.5 gives graphs of hysteresis of a mixed-wet rock type.

When a rock type does not trap oil, oil saturation end-points S_{orw}, S_{ormax} and S_{orh} overlap, which means all imbibition processes advance towards S_{orw}. In this case, the definition of the oil-trapping relationship is not required. Carlson's method is no longer applicable for K_{rw} hysteresis since it is based on the difference between S_{orh} and S_{orw}.

The Killough methods still hold for capillary pressure and relative permeability calculations if, in those equations, S_{orw} is substituted for S_{orh} and S_{ormax}. K_{rw} hysteresis differences should, however, be noted. When S_{ormax} becomes identical to S_{orw}, Eq.8 simplifies to a constant value, K_{rw} at 1-S_{orw}. The original Killough method (Eq.7) is thus simplified to:

$$K_{rw}^{Scan}(S_w) = K_{rw}^{Dr}(1 - S_{ohy}) \left(1 - \frac{K_{rw}^{Im}(1 - S_o^{\text{norm}})}{K_{rw}^{Im}(1 - S_{orw})} \right) + K_{rw}^{Im}(1 - S_o^{\text{norm}}) \quad (\text{Eq.7a})$$

When boundary imbibition K_{rw} is defined to be less than the drainage K_{rw}, such as the curves provided by Fig.5, Eq.7a will tend to be even lower than the boundary imbibition curve. On the other hand, when using the modified Killough method (the default), Eq.9 is simplified to:

$$K_{rw}^{Scan}(S_w) = K_{rw}^{Dr}(S_w) + krwco \times [K_{rw}^{Im}(1 - S_o^{\text{norm}}) - K_{rw}^{Dr}(1 - S_o^{\text{norm}})] \quad (\text{Eq.9a})$$

Eq.9a is applicable for imbibition K_{rw} either greater or less than the drainage K_{rw}. The krwco coefficient in Eq.9a, which has a default value of 0.5, can be adjusted from 0 to 1 using keyword *KILLOUGH_MOD_KRWCO.

Remarks and Restrictions

The consistently coupled hysteresis methods described in this document are based on the consideration that water and oil are not substituted for each other. Option *OILWET following keyword *RPT is generally not applicable with these methods. The hysteresis described herein illustrates the ability of IMEX to model reservoirs with wettability

conditions varying from water-wet to strongly mixed-wet. This coincides with the widely accepted view that most oil reservoirs are mixed-wet or water-wet.

The mixed-wet or trapped oil hysteresis option assumes the input primary drainage curve is used to calculate the initial equilibrium P_{cow} . The option that specifies the use of the boundary imbibition P_{cow} as the initial equilibrium capillary pressure is inconsistent with this approach, hence, the use of keyword *RPT “set number” *IMBIBITION will be flagged as an error.

If gas exists in the block, the gas is counted as the non-wetting phase. K_{row} is looked up with respect to the wetting phase saturation; therefore, it will be inconsistent if the 3-phase K_{ro} is looked up using Stone’s method with the *SO option enabled (see*KROIL). For the same reason, the *KROIL *SEGREGATED option does not work with the hysteresis option.

Although allowed, altering the rock type of grid blocks in recurrent data is not recommended when simulation with the trapped oil option is enabled. The historical maximum attained oil saturation of a block may not be consistent with the residual oil saturation of imbibition if rock types are altered during a run.

Output of Sormax and Sorh Distributions

Block $S_{or_{max}}$ and S_{or_h} can be output to simulation text-based output as well as to Results SR2 files, if any part of the reservoir is modeled using trapped oil hysteresis. Subkeywords *SORMAX and *SORH can follow after either *OUTPRN *GRID or *OUTSRF *GRID keyword in the IO data section.

References

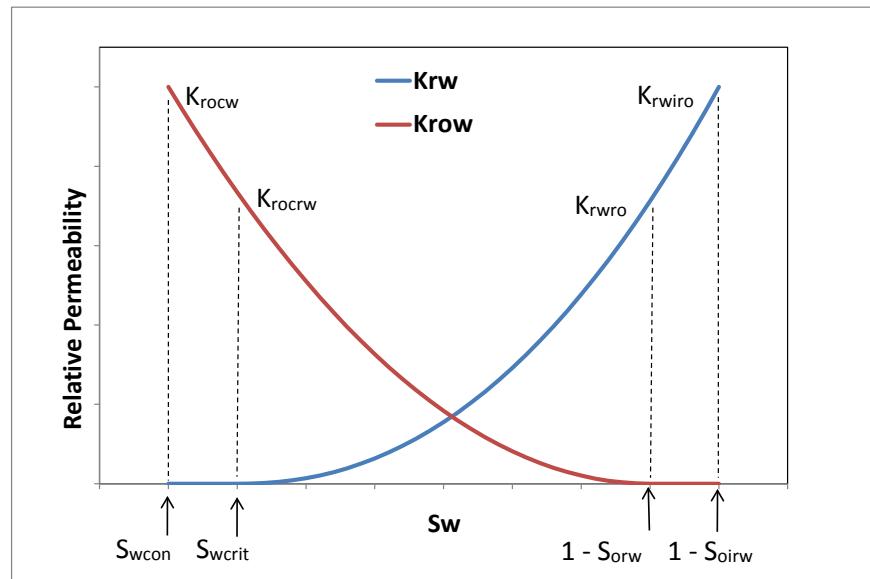
Land, C. E. “Calculation of Imbibition Relative Permeability for Two- and Three-Phase Flow from Rock Properties,” paper SPE 1942, Society of Petroleum Engineers Journal (1968), Vol. 8, No. 2, 149-156.

Killough, J. E. “Reservoir Simulation with History-dependent Saturation Functions,” paper SPE 5106, Society of Petroleum Engineers Journal (1976), Vol. 16, No. 1, 37-48.

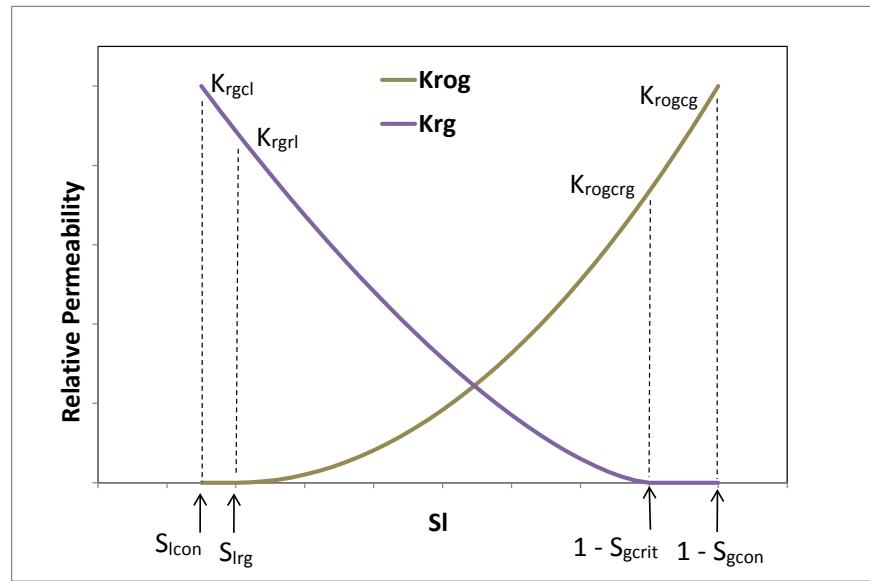
Carlson, F. M. “Simulation of Relative Permeability Hysteresis to the Non-Wetting Phase,” paper SPE 10157, presented at the SPE Annual Technical Conference & Exhibition, San Antonio, Texas, USA (October 5-7, 1981).

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 K_{rw} , K_{row} , K_{rog} and K_{rg} 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 K_{rw} to describe the scaling method. Other relative permeability values are scaled in a similar fashion.

There are two types of K_{rw} value scaling, which depend on how endpoints are defined for a grid block.

a) Two-Point Scaling

A water relative permeability curve, K_{rw} , always has two endpoints, one at each end of the curve, i.e. K_{rw} at critical water saturation ($K_{rw}@S_{wcrit}=0$) and K_{rw} at irreducible oil ($K_{rwiro} = K_{rw}@S_{oilw}$). Thus if a grid block is only assigned a K_{rwiro} value (using *KRWIRO), the block K_{rw} 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 K_{rw} 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 further assigned $K_{rwro}(b)$, using the *KRWRO keyword AND the corresponding oil-water Kr table contains a residual oil saturation endpoint, $S_{orw}(t)$ which is different from $S_{oilwt}(t)$, the block K_{rw} value will be scaled using three-point scaling, where:

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

or

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

If the table's K_{rw} curve is flat between $1-S_{orw}$ and $1-S_{oilw}$, which means the term $(K_{rwro}(t) - K_{rwiro}(t))$ is close to zero, an alternate linear interpolation will be applied when $S_w > 1 - S_{orw}$:

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

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

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 and in the *SWT table associated with that block, the table value of Sorw must be different from the table value of Soirw.

Krow:

*KROCRW array value for that block must be defined and in the *SWT table associated with that block, the table value of Swcrit must be different from the table value of Swcon.

Krog:

*KROGCRG array value for that block must be defined and in the *SLT table associated with that block, the table value of Sgcrit must be different from the table value of Sgcon.

Krg:

*KRGRL array value for that block must be defined and in the *SLT table associated with that block, the table value of Slrg must be different from the table value of Slcon.

See the *SWCON keyword for a complete definitions of the endpoints referred to above.

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 value scaling is independent of three-point saturation scaling as defined by the *3PTSCALING keyword. Both options can be used independently. Using the three-point saturation scaling option (using *3PTSCALING) does not turn on the three-point relative permeability endpoint value scaling option. Only defining the critical relative permeability endpoints (as discussed above) activates this option.

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 1.0
*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 angle
    1000.0      .4        .002      230.0     1.0
*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

Any line in a data file has a maximum length of 512 characters. Any characters after number 512 are ignored. This maximum input length may be changed by changing the parameter MDLINE in the source code file KWCOM1.INC, and recompiling the simulator.

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'). Embedded single quotes are not allowed; however, double quotes ("") may be used inside single quotes, e.g., "This is the "right" way".

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). P_{cow} 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 ($P_{cow} = 0$), 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	(P_{cow})
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	

However, use of the *INT keyword (See *INT) to interpolate omitted entries allows tables such as the one above to be used. For example:

*SWT			
**Sw	krw	krow	(P_{cow})
0.2	0.0	1.0	45.0
0.2899	0.022	0.6769	19.03
0.3778	*INT	0.4153	10.07
0.4667	0.061	0.2178	4.09
0.5556	0.143	0.0835	1.80
0.6782	0.289	*INT	.50
0.7561	*INT	0.0	.10
0.8325	0.780	0.0	.0
0.9222	1.000	0.0	*INT
1.0000	1.000	0.0	*INT

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, timestep calculations will be done.

CONDITIONS:

This keyword must appear in the Input/Output Control section.

It is also possible to specify a checkonly run from the command line by entering the "-checkonly" flag. This use of the checkonly mode has the additional benefit of not requiring an IMEX run license to function.

Example:

```
C:\CMG\IMEX\98.00\exe\dyn\mx9800.exe -checkonly
```

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.

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. By default only the first 20 lines of input data is listed unless the keyword *NOLISTLIM is found in the Input/Output Control section (See *NOLISTLIM below)

*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. When large arrays are listed under *LIST, only the first 20 lines of data are echoed, in order to echo all lines of data (this can result in large files) the *NOLISTLIM keyword in the Input/Output Control section is required. 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.

Controlling Data File Listing Using NOLISTLIM (Optional)

***NOLISTLIM**

PURPOSE:

*NOLISTLIM specifies that the default 20 lines of keyword echoing under the *LIST keyword be extended so that all values of the keyword are echoed regardless of the number of lines required to do so. This can create large files if used, but allows the user to view all of the data.

FORMAT:

*NOLISTLIM

DEFAULTS:

Optional keywords. Defaults to a list limit of 20 lines per keyword

CONDITIONS:

*NOLISTLIM must appear on a line by itself in the Input/Output Control Section.

EXPLANATION:

By default, the entire data file is listed to the output print file prior to the start of the simulation run. When keywords containing large amounts of data are listed under *LIST, only the first 20 lines of data are echoed. In order to echo all lines of data (this can result in large files) the *NOLISTLIM keyword in the Input/Output Control section is required.

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'
```

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 elements of the grid block (e.g., matrix and fracture). The *array_qualifier* is optional. Choices are

*MATRIX

*FRACTURE

*ALLELEM

followed by optional grid qualifier

*RG *uba_range* (*FZ | *NFZ)

*BG '*block_group*' (*FZ | *NFZ)

*EQUALSI

If no *array_qualifier* is present then *MATRIX is assumed. These *array_qualifier* keywords are described separately.

If grid qualifier is absent then the fundamental grid is assumed.

Each of the above array reading qualifiers will access only the element indicated. 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, FR, WB and TU. A UBA range is not allowed with *read_option* *EQUALSI or *array_modifier* *MOD.

'block_group'

Quoted block group name defined by keyword *BLOCKGROUP. See manual page for *BG in this chapter. See *read_option* *FZ and *NFZ.

read_option

The read options are

- *CON
- *IVAR
- *JVAR
- *KVAR
- *ALL
- *IJK
- *FZ
- *NFZ

These *read_option* keywords are described separately.

All these *read_options* 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*; for *IJK it depend also on whether the context is recurrent data or not.

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.

Entering Matrix Grid Properties

*MATRIX

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 *i j k*

DEFINITIONS:

i

I direction index of the fundamental grid block containing the refined grid.

j

J direction index of the fundamental grid block containing the refined grid.

k

K direction index 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 $n_i = 3$, $n_j = 2$ and $n_k = 1$. 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:

<i>array_qualifier</i>	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:

<i>read_option</i>	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 are 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 $3 \times 2 \times 2$ 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 $\frac{1}{4}$ 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
```

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.
```

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 i₂

Initial and final I direction grid block indices.

j₁ j₂

Initial and final J direction grid block indices.

k₁ k₂

Initial and final K direction grid block indices.

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 $i_1=1$, $i_2=3$, $j_1=3$, $j_2=4$, and $k_1=k_2=2$. 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 $n_i=20$, $n_j=25$, and $n_k=6$ 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(n_i)*

DEFINITIONS:

value(1)

Value assigned to all grid blocks with an I direction index of 1.

n_i

Number of grid blocks in the I direction.

EXPLANATION:

Enter n_i values separated by spaces or commas.

Example: I direction block sizes where $n_i = 10$:

```
*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 $n_i = 3$:

```
*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(n_j)*

DEFINITIONS:

value(1)

Value assigned to all grid blocks with a J direction index of 1.

n_j

Number of grid blocks in the J direction.

EXPLANATION:

Enter n_j values separated by spaces or commas.

Example: The J direction increments for a problem where $n_j = 10$ 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(n_k)*

DEFINITIONS:

value(1)

Value assigned to all grid blocks with a K direction index of 1.

n_k

Number of grid blocks in the K direction.

EXPLANATION:

Enter n_k 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 $n_k = 5$, 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($n_i \times n_j \times n_k$)*

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: $n_i = 10$, $n_j = 3$, $n_k = 2$

```
*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
```

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.

*FZ *fz_centre fz_tip* may be used only by a fracture grid created with *PLNRFRAC.

EXPLANATION:

For a fracture-type local grid created by *REFINE *INNERWIDTH 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 *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
```

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 )
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, *MATRIX, *FRACTURE).

<u>operator</u>	<u>Action</u>
+	Add <i>value</i> to the existing property value
-	Subtract <i>value</i> from the existing property value
*	Multiply the existing property value by <i>value</i>
/	Divide the existing property value by <i>value</i>
=	Replace the existing property value by <i>value</i>

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 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 $(i_2 - i_1 + 1)(j_2 - j_1 + 1)(k_2 - k_1 + 1)$ 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).

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).

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 10×6×1 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 2×2×1 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 10×6×1 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)

```

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.

The *INT option can be used for extrapolation, however the use of *INT may violate saturation table end point checks (due to the extrapolation). The use of *INT for extrapolation on saturation tables is not advised.

The input of only one non *INT entry is handled as a special case to allow the user to set an entire column to the same value.

Input/Output Control

Command Line Input/Output Control

Input/Output Control

Command Line Options:

IMEX has several useful command line options.

-f input_data	Defines the input-data-file.
-r input_restart	Defines the restart irf file.
-checkonly	Runs the model in check only mode. An IMEX license is not required for a check only run specified from the command line.
-dimsum	Prints detailed information on model dimensioning at run time.
-onestep	Runs the model for one timestep only. See the *MAXSTEPS description in the Numerical Methods section for more details.
-maxsteps	Sets the maximum timestep number to n. See the *MAXSTEPS description in the Numerical Methods section for more details.
-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	Output files will be written to the directory that contains the data file. This option is intended to be used when an absolute pathname has been supplied via the “-f” argument.
-wd <i>path</i>	Output files will be written to the directory given by <i>path</i> . This option is useful in an environment where the “current directory” may not be defined.

-log (<i>log_path</i>)	Specifies that consol “diary” output will be redirected to a file. If <i>log_path</i> is present, this file has pathname <i>log_path</i> and extension “.log” will be added if it is not already present. If <i>log_path</i> is absent, this file’s name has the same base as the output files but extension “.log”. This file will not contain error or status messages from the operating system. 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. In addition, the user *INTERRUPT *INTERACTIVE choice is also entered in the ".in" file when the –log option is used. A carriage return in the ".in" file where the simulator is expecting to read the interrupt option defaults to *RESTART-STOP.
-doms	Used to enable Jacobian building parallelization from the command-line, its use requires that you have a license with “Parallel IMEX” enabled (see *DPLANES in the Numerical Methods section for more details). “-jacpar” is now recommended.
-parasol n	Used to enable Solver parallelization from the command-line for a target of n logical CPU's, its use requires that you have a license with “Parallel IMEX” enabled (see *SOLVER and *PPATTERN in the Numerical Methods section for more details).
-aimsol	Overrides numerical keyword specifying that PARASOL is to be used as the solver.
-jacdoms n	Used to enable *JACDOMS Jacobian building parallelization from the command-line for a target of n logical CPU's, its use requires that you have a license with “Parallel IMEX” enabled (see *JACDOMS in the Numerical Methods section for more details). “-jacpar” is now recommended.
-jacpar	Used to enable *JACPAR *ON Jacobian building parallelization from the command line. This option uses Jacobian domains based on the Parasol classes. It uses the same number of threads as specified by *PARASOL or *PNTHRDS options. Its use requires that you have a license for “Parallel IMEX”.
-pnthrds n	Used to enable *PNTHRDS n, where n is the desired number of threads. This option may be used to override the default number of threads specified using other options.
-cputime	This command line option enables non-intrusive clock and CPU time profiling of IMEX equivalent to the keywords *DEBUG *CPUTIME *SOLVER.

-partition_info	Output detailed information about Parasol partitioning
-old_autopslab	Revert to old AUTOPSLAB option
-autop2d IJ	To enable automatic parallel partitioning similar to AUTOPSLAB, in 2 directions (any two of the x,y,z coordinates) instead. I/J/K represents x,y,z coordinates. IJ gives partitioning order in I direction first and then J direction, while JI works as partitioning J direction first and I the next. Two of -autop2di n, -autop2dj m and -autop2dk l are used in consistency with -autop2d I/J/K to assign n-1 and m-1 separator planes in each direction. The default value is 1 separator plane. (see *PPATTERN *AUTOP2D in the Numerical Methods section for full details.)
-autop2di n	Revert to old reordering option
-autop2dj m	Enables cache alignment to ibytes bytes. Use ibytes=0 to turn it off as done using the corresponding keyword CACHE_ALIGN. See CACHE_ALIGN in the Numerical Methods section for details.
-autop2dk l	Enables or disables fully implicit alignment. Without on or off defaults to on as done using the corresponding keyword FI_ALIGN if MDIMPL is 100 otherwise defaults to off. See FI_ALIGN in the Numerical Methods section for details.
-psreor_new	Enables or disables fully implicit alignment. Without on or off defaults to on as done using the corresponding keyword FI_ALIGN if MDIMPL is 100 otherwise defaults to off. See FI_ALIGN in the Numerical Methods section for details.
-cache_align ibytes	Changes the initial targeted maximum percent of fully implicit blocks for dimensioning of adaptive implicit based arrays used for Jacobian building and the linear solver.
-fi_align (on off)	Enable/override WATER_FIX “n” option from the command line
-ai_align (on off)	Enable command-line restart from timestep step. If the <i>step</i> is left blank (-restart without argument), the run will restart from the last timestep.
-mdimpl mdimpl	Enable command-line restart from time <i>time</i> , where <i>time</i> is a decimal number of time.
-water_fix “n”	Enable command-line restart from the <i>date</i> . The <i>date</i> takes the format YYYY_MM_DD.DDDD....
-restart <i>step</i>	Example:
-restime <i>time</i>	-resdate 2001_03_01
-resdate <i>date</i>	-resdate 2001_03_12.345678

-restart_sr2 main/separate	<p>Command-line option to indicate the sr2 files are combined or separated, for both input and output. If ‘-restart_sr2 main’ is assigned on the command line, sr2 will be considered to have restart and results information combined in one sr2 file set. Otherwise if ‘-restart_sr2 separate’ is used, restart and results will be treated separately in two different sr2 file sets.</p> <p>Combined or separated sr2 files should keep consist between initial and restart run. No mixed uses of sr2 files are supported. Currently, IMEX default is to use the separate sr2.</p> <p>By default, when separate files are used, graphics data is written out in single precision while restart data is written out in double precision.</p>
----------------------------	---

Please note that LAUNCHER enables the parallel option by using command line options when submitting batch jobs. The command line options will override selected parallel keywords in the data set.

If you want to ensure that the parallel control keywords in the data set are always honored, it is necessary to run the Simulator from LAUNCHER as if in serial mode (specify that you want to use only one processor).

If this is done, all parallel keywords in the NUMERICAL section of the data set will be honored. Obviously when doing this ensure the NUMERICAL section completely defines the parallel options to be used.

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 data set 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 SR2 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 SR2 result files 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

** Initial solution gas from primary production
*MFRAC_OIL 'SOLN GAS' *BINARY_DATA
  *MOD 4 5 1:9 = 0.3 ** Enriched zone around well
```

ASCII Characters

IMEX Data File and Directories

It is crucial that IMEX data files and data/output directories do not include non-ASCII characters (such as accents, Greek symbols, etc). The IMEX data line interpreter does not handle such characters properly.

Please ensure that IMEX data set names and the complete data and output directory paths (from the Drive Letter to the data/output directories) only include standard ASCII text.

Input/Output File Names (Optional)

***FILENAME, *FILENAMES**

PURPOSE:

Specify names for input and output files.

FORMAT:

***FILENAME(S) file_types name_options**

where file_types are one or more of:

- *OUTPUT**
- *INDEX-OUT**
- *MAIN-RESULTS-OUT**
- *REWIND-OUT**
- *INDEX-IN**
- *MAIN-RESULTS-IN**
- *REWIND-IN**

For file_types ***INDEX-OUT**, ***MAIN-RESULTS-OUT**, ***REWIND-OUT**, ***MAIN-RESULTS-IN** and ***REWIND-IN** one of these name_options may be specified.

- "
- 'filename'
- *PROMPT**

For file_type ***OUTPUT** one of these name_options may be specified.

- "
- 'filename'
- *PROMPT**
- *SCREEN**

For file_type ***INDEX-IN** one of these name_options may be specified.

- 'filename'
- *PROMPT**

DEFINITIONS:

***FILENAME(S)**

File name keyword. The trailing 'S' is optional.

***OUTPUT**

Indicates the output file to which formatted simulation results will be written.

***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.

***REWIND-OUT**
Indicates the rewritable-results-file (rrf) to which the restart data is written when the *REWIND option is used.

***INDEX-IN**
Indicates the index-results-file from which simulation results and restart records are read. This file is only necessary for restart runs.

***MAIN-RESULTS-IN**
Indicates the main-results-file from which the simulation results and restart records (binary) are read. This file is only necessary for restart runs.

***REWIND-IN**
Indicates the rewritable-results-file from which the rewound restart records (binary) are read. This file is only necessary for restart runs.

..

Empty string, denoting that the default file name will be used. This is the default if the *FILENAMES keyword is missing from the input-data-file.

'filename'
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.

***SCREEN**
Indicates that this file will go to the screen (standard output device).

DEFAULTS:

If any required input/output file name has not been specified via *FILENAME, then *PROMPT is assumed. For example, *FILENAME implies *FILENAME *OUTPUT *PROMPT.

If *FILENAME file_type is used to assign a file name but the name_options list is absent then " is used, triggering the default file names, below. For example, *FILENAME *OUTPUT implies *FILENAME *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 input-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, 'irf' for SR2

index and 'mrf' for SR2 main-results-file. 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
- c) discard everything after column 8.

For example, the UNIX 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
*MAIN-RESULTS-OUT	*OUTPUT	.mrf
*REWIND-OUT	*OUTPUT	.rrf
*INDEX-IN	(no default)	.irf
*MAIN-RESULTS-IN	*INDEX-IN	.mrf
*REWIND-IN	*INDEX-IN	.rrf

With this defaulting system, the user is able to perform a series of 'bootstrapped' restart runs by changing only the data file name and *INDEX-IN file names for each run.

Example #1: 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'.
```

Example #2: Data file is 'cycle.dat'. The first run segment has

```
*FILENAME *OUTPUT 'trial1'.
*INDEX-OUT default is 'trial1.irf'
*MAIN-RESULTS-OUT default is
'trial1.srf'
```

For the second run segment, modify 'cycle.dat' to add the *RESTART keyword, and use

```
*FILENAME *OUTPUT 'trial2.out', and
*FILENAME *INDEX-IN 'trial1.irf'.
```

Do not rename or delete files.

```
*INDEX-OUT default is 'trial2.irf'
*MAIN-RESULTS-OUT default is
'trial2.mrf'
*MAIN-RESULTS-IN default is
'trial1.mrf'
```

CONDITIONS:

*FILENAMES must be the first keyword(s) in the input-data-file. If not, the user will be prompted for the file names.

File names must be the first keywords in the Input/Output Control section of an input-data-file.

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 new Simulation Results File System (Generation 2) is also referred to as SR2. The SR2 file system consists of three files that work together. These are the index-results-file (irf), the main-results-file (mrf) and the rewindable-results-file (rrf). It is necessary for the graphics post-processor "RESULTS" that two of these files exist. These are the irf and the mrf files. These files are also required for restart runs. If the *REWIND option was used to write restart records, then the rewindable-results-file (rrf) is also required for restart runs.

PLEASE NOTE:

The SR2 file system is incompatible with the previous SRF file system. In order to create an SRF file from the SR2 file system please use the SR2 to SRF conversion program provided with this release.

Example:

```
** use default file name for index-out-file
** 'root.irf'.
** Prompt for input mrf file *MAIN-RESULTS-IN by
** leaving it out of the input-data-file.
*FILENAMES      *OUTPUT                  'root.out'
                 *INDEX-OUT
                 *MAIN-RESULTS-OUT    'name.mrf'
                 *INDEX-IN       'input.irf'
```

Important Notes (Run Time Dimensioning)

DEFUALTS:

IMEX does not assume any defaults for problem dimensions, however; there are minimum values which override the automatic allocation procedure. These minimums in turn can be overridden if the user explicitly specifies small dimensions using the *DIM card.

IMEX is always run time dimensioned. The absence of Dynamic Dimensioning section in the data file does NOT mean static dimensioning is employed.

In order to maintain backward compatibility, this entire section has been developed as optional. Users need not enter any of the keywords for the Run Time Dimensioning section, most existing data sets should work without any difficulty.

The *DIM section will normally be omitted.

Problem Type Where IMEX Fails to Allocate Adequate/ Appropriate Storage

One type of model causes IMEX to allocate too much memory. This is data which has pinched out blocks.

Data sets which have blocks pinched out by the grid module cannot have the number of active blocks accurately determined. IMEX will print a warning and set the number of active blocks equal to the number of total blocks and continue on with initialization.

After Grid Module initialization IMEX prints out the number of active blocks used (in the pinched out case --- total blocks) and the minimum value of active blocks in the correct card format to be able to be pasted directly into the data set directly.

Scan the output file for

*DIM *ACTIVE_BLOCKS

and place this card in your data file if the dimensions used are significantly different from the minimum dimensions calculated.

A timestep need not be completed for the minimum dimensions to be determined. It is only necessary to complete the Grid Initialization section of IMEX.

One type of model may cause IMEX to allocate too little temporary storage. These are problems with a small (less than 20%) percentage of active blocks compared to total blocks (i.e. NX*NY*NZ).

When IMEX warns that insufficient temporary storage is available, increase the number of active blocks dimensioned for, using the *DIM *ACTIVE_BLOCKS keyword.

If you need to enter run time dimensioning control data, place this data before the Input/Output Control section. If there are any *FILENAME keywords that are being used, the *FILENAME keywords should be placed above the Run Time Dimensioning section.

Run Time Dimensioning (Optional)

*DIM

PURPOSE:

Specify a dimensioning variable with a value.

FORMAT:

*DIM dim_variable dim_value

where dim_variable are one or more of:

- *ACTIVE_BLOCKS
- *NULL_BLOCKS
- *MAXFAULTS
- *MAX_AQUIFERS
- *MAX_SECTORS
- *MAXLEASELINES
- *MAXBLKS_INLEASELINES
- *MAX_PVT_REGIONS
- *MAX_ROCK_TYPES
- *MAXENTRIES_IN_KR_TABLE
- *MAXENTRIES_IN_PVT_TABLE
- *MAX_SPECIALVARS_IN_SRFOUTPUT
- *MAXPERCENT_OF_FULLYIMPLICITBLOCKS or *MDIMPL
- *SOLVER_DIMENSIONING

The *SOLVER_DIMENSIONING card must be read before the following can be read

- *MAXORTHOGONALIZATIONS
- *MAXINTERBLOCK_CONNECTIONS
- *MAXOFFDIAGONAL_ENTRIES
- *MAX_BLOCKENTRIES_IN_LU_FACTORS
- *MAXSIZE_SOLVER_VECTOR
- *MAX_SOLVER_DIAG_ENTRIES
- *MAX_NONZERO_L_OR_U_ENTRIES

DEFINITIONS:

*DIM

DIM keyword. Required to dimension variables.

*ACTIVE_BLOCKS

The number of active blocks in a given run. Active blocks are all the active blocks including dual porosity, hybrid or refined grids, excluding null (zero porosity) blocks.

Default is to obtain this information directly by reading the data file accounting for any NULL blocks.

Example: *DIM ACTIVE_BLOCKS 2000

*NULL_BLOCKS

The number of null blocks in a given run. The default would be no NULL blocks. All blocks would be considered ACTIVE.

Example: *DIM NULL_BLOCKS 500

*MAX_FAULTS

Maximum number of faults

Example: *DIM MAX_FAULTS 15

Default value of MAX_FAULTS is set to 15.

*MAX_AQUIFERS

Maximum number of aquifers

Example: *DIM MAX_AQUIFERS 15

Default number of MAX_AQUIFERS is 15

*MAX_SECTORS

Maximum number of sectors

Example: *DIM MAX_SECTORS 50

Default number of MAX_SECTORS is 50

*MAXLEASELINES

Maximum number of lease-lines

Example: *DIM MAXLEASELINES 12

Default number of MAXLEASELINES is 15

*MAXBLKS_INLEASELINES

Maximum number of blocks in a lease-line

Example: *DIM MAXBLKS_INLEASELINES 5000

Default number of MAXBLKS_INLEASELINES is 150

*MAX_PVT_REGIONS

Maximum number of PVT regions allowed.

Example: *DIM MAX_PVT_REGIONS 50

Default number of MAX_PVT_REGIONS is 15.

***MAX_ROCK_TYPES**

Maximum number of rock types allowed.

Example: *DIM MAX_ROCK_TYPES 100

Default number of MAX_ROCK_TYPES is 30.

***MAXENTRIES_IN_KR_TABLE**

Maximum number of entries in a rock-fluid table

Example: *DIM MAXENTRIES_IN_KR_TABLE 500

Default number of MAXENTRIES_IN_KR_TABLE is 101.

***MAXENTRIES_IN_PVT_TABLE**

Maximum number of entries per PVT table

Example: *DIM MAXENTRIES_IN_PVT_TABLE 51.

Default number of MAXENTRIES_IN_PVT_TABLE is 51.

***MAX_SPECIALVARS_IN_SRFOUTPUT**

Maximum number of *SPECIAL variables in SRF output allowed.

Example: *DIM MAX_SPECIALVARS_IN_SRFOUTPUT 500

Default number of MAX_SPECIALVARS_IN_SRFOUTPUT is 100.

***MAXPERCENT_OF_FULLYIMPLICITBLOCKS or *MDIMPL**

Maximum percent of fully implicit blocks. Chosen between 1 and 100. If fully implicit, choose 100.

Example: *DIM MDIMPL 100.

Default number of MDIMPL is 1 if the exe is WIN32 otherwise the default number is 100.

If more than the initial estimated storage for the target number of fully implicit blocks is required then the corresponding adaptive implicit arrays are deallocated, a new target estimate is calculated and the corresponding adaptive implicit arrays are re-allocated based on the new target estimate. The new target mdimpl is based on the minimum of (twice the current mdimpl, the current mdimpl + 20, 100) or the actual current percentage implicit if it is larger.

***SOLVER_DIMENSIONING**

Should solver arrays be custom dimensioned. Requires a 'ON' or 'OFF' (in quotes) as a response.

Example: *DIM SOLVER_DIMENSIONING 'ON' to turn on custom dimensioning

Example: *DIM SOLVER_DIMENSIONING 'OFF' to turn off custom dimensioning.

By default, SOLVER_DIMENSIONING is set to 'OFF'.

*MAX_ORTHOGONALIZATIONS

Number of orthogonalizations in the linear solver. Reducing the number of orthogonalizations will save storage, but will degrade performance.

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_ORTHOGONALIZATIONS 5

The default value of MAX_ORTHOGONALIZATIONS is set to 30.

*MAX_INTERBLOCK_CONNECTIONS

Maximum number of interblock connections. Interblock connections include connections between grid blocks and those due to wells.

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_INTERBLOCK_CONNECTIONS 5

The default value of MAX_INTERBLOCK_CONNECTIONS is estimated based on the number of fundamental grid blocks, NI, NJ and NK, the number of null blocks and the problem types to be run.

*MAX_OFFDIAGONAL_ENTRIES

Maximum number of entries in off-diagonal Jacobian matrix.

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_OFFDIAGONAL_ENTRIES 5

The default value of MAX_OFFDIAGONAL_ENTRIES is estimated based on the number of fundamental grid blocks, NI, NJ and NK, the number of null blocks and the problem types to be run.

*MAX_BLOCKENTRIES_IN_LU_FACTORS

Maximum number of block entries in LU Factors.

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_BLOCKENTRIES_IN_LU_FACTORS 5

The default value of MAX_BLOCKENTRIES_IN_LU_FACTORS is estimated based on the number of fundamental grid blocks, NI, NJ and NK, the number of null blocks and the problem types to be run.

*MAXSIZE_SOLVER_VECTOR

Maximum size of solver solution vector

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_SOLVER_VECTOR 5

The default value of MAXSIZE_SOLVER_VECTOR is estimated based on the number of fundamental grid blocks, NI, NJ and NK, the number of null blocks and the problem types to be run.

***MAX_SOLVER_DIAG_ENTRIES**

Maximum number of solver diagonal entries

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_SOLVER_DIAG_ENTRIES 5

The default value of MAX_SOLVER_DIAG_ENTRIES is estimated based on the number of fundamental grid blocks, NI, NJ and NK, the number of null blocks and the problem types to be run.

***MAX_NONZERO_L_OR_U_ENTRIES**

Maximum number of nonzero L or U entries

Conditional input valid only when SOLVER_DIMENSIONING is set to 'ON'.

Example: *DIM MAX_NONZERO_L_OR_U_ENTRIES 5

The default value of MAX_NONZERO_L_OR_U_ENTRIES is estimated based on the number of fundamental grid blocks, NI, NJ and NK, the number of null blocks and the problem types to be run.

Background (Run Time Dimensioning)

During every run of the simulator, certain amount of computer system's memory resources are required to store the variables used by the simulator. Depending on how a variable is declared in the program, this storage could be allocated to it when the program is being compiled or when the program is being run. For example, the following dimension statement in the program code would cause the array A to be allocated memory at compile time

DIMENSION A(5)

Note that A has been dimensioned to 5 units in the above code. The memory to store array A has been allocated at compile time and this way of dimensioning A could be called "compile time dimensioning (CtD)". The advantage of CtD is its simplicity and merely declaring A and its size is sufficient to allocate the required memory. The main disadvantage of CtD is that we may not know in advance the maximum sizes of many dimensioned variables. For example, what if in a particular application we need 6 values of A to be represented. With CtD we have to change A(5) in the above DIMENSION statement to A(6), recompile the entire program and run it.

IMEX 96.00 and older versions of IMEX have followed CtD. They have consistent sets of maximum dimensioning values for all its dimensioned variables that result in different sized executables when compiled.

The current version of IMEX uses a different approach to dimensioning variables. It uses a pointer based approach to dimension its variables at run time of the simulation and not at compile time. To illustrate run time dimensioning, let us reconsider the same DIMENSION statement mentioned above. This statement is now written as,

```
pointer(p_A, A)
p_A = malloc((size of A in bytes)*5)
```

In the above lines, p_A is the pointer variable and is associated with A through the pointer statement. In the following statement, the exact memory required to store A(5) is computed in bytes and allocated using a special function by name malloc. The pointer variable, p_A, holds the address of this allocated space and makes it available to store the values of A in the program. This is the scheme of memory allocation followed by IMEX.

Advantages of Run Time Dimensioned IMEX

Run time dimensioning confers several advantages to IMEX. The principal advantages are,

1. IMEX comes in only one executable. There is no need for any custom sized executables. This results in saving some disk space.
2. IMEX uses only the memory it requires for a given problem. With the same hardware, this allows the solution of bigger (sized) problems than before.
3. Only hardware limitations would dictate the maximum size of the problems that could be solved using IMEX. There would be no such limitations on the software side.

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 input-data-file.

EXPLANATION:

This keyword identifies an alphanumeric character string used for project identification. It will appear both in the output file and in the index-results-file.

Any character string with embedded blanks or commas must be enclosed with single quotes.

Examples:

```
*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 input-data-file.

EXPLANATION:

This keyword identifies an alphanumeric character string used for project identification. It will appear both in the output file and in the index-results-file.

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 F.R. Sangiovanni, December 16, 1988. IBM '
```

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 input-data-file.

EXPLANATION:

This keyword identifies an alphanumeric character string used for project identification. It will appear both in the output file and in the index-results-file.

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 F.R. Sangiovanni, December 16, 1988. IBM '
*TITLE3
'IMEX 4.0; 4200 grid blocks; variable thickness'
```

Case Identification (Optional)

***CASEID**

PURPOSE:

*CASEID is used to identify specific case runs.

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 input-data-file.

EXPLANATION:

This keyword identifies an alphanumeric character string used to name specific case runs. It will appear both in the output file and in the index-results-file.

Examples:

*CASEID 'Case 21c'

Check Only (Optional)

***CHECKONLY**

PURPOSE:

*CHECKONLY indicates that the well data is to be checked for format errors, but the simulation will not be run. Also, please note that on the command line "-checkonly" enables this option without requiring the *CHECKONLY Card. In addition using the command line version of checkonly disables License Management for the "CHECKONLY" run. This allows the user to run a model and check data simultaneously using a single IMEX License.

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.

Trap UNIX Signal 2 Interrupt (Optional)

*INTERRUPT

PURPOSE:

*INTERRUPT indicates that when the user interrupts the simulation run by simultaneously pressing the "CTRL-C" keys or by typing "kill -2" followed by the process identification number (PID), one of the following user specified actions can take place:

1. Continue the simulation run.
2. Finish the current timestep, write a restart record and stop the simulation.
3. Flush buffers, close files and stop the simulation run immediately.
4. Write an impromptu restart and continue.
5. For PC only: Stop cold (immediately and unconditionally). Interactively, this action has to be requested with a Control-Break.

Use of this keyword (1-3) prevents corruption of the irf and mrf files when aborting a simulation run and optionally writes a restart record before stopping. It (4) also allows writing of an impromptu restart record.

FORMAT:

```
*INTERRUPT (*RESTART-STOP)      (Default)
              (*WRST-CONT)
              (*STOP)
              (*STOP-COLD)
              (*INTERACTIVE)
```

DEFINITIONS:

*RESTART-STOP

This subkeyword specifies that the current timestep be completed, all specified output data be written to the output and SR2 files and a restart record also be written to the SR2 files.

*WRST-CONT

This subkeyword specifies that an impromptu restart be written, i.e. all specified output data be written to the output and SR2 files and a restart record also be written to the SR2 files. The run then continues.

*STOP

This subkeyword specifies that the simulation run be terminated immediately. The current timestep is not to be completed. However, prior to termination, all buffers will be flushed and files closed to prevent file corruption.

***STOP-COLD**

***** for the PC only *****

This subkeyword specifies that the simulation run be stopped cold, immediately and unconditionally. The action is that of a raw Control-C without any trapping. This is the equivalent of Control-Break in interactive mode. This option is needed as the action on any of the other options is not instantaneous on the PC, hence may take too long.

***INTERACTIVE**

This subkeyword specifies that the user will be prompted to enter the action to be taken when a Unix signal 2 is detected.

Signal file:

If the user places a file in the data directory with the extension “.signal” and the same name (without the extension) as the data set, the simulator running that data set will act as if it has received an interrupt. If the signal file is empty, the action the simulator will take will default to the last defined action. If the action was not previously defined, the simulator will treat the existence of the file as a *RESTART-STOP signal (the default).

If the file contains a single integer in the range 1 to 4, then the simulator will

Integer	Action
1	Continue Simulation Run
2	Write Restart and Quit
3	Tidy up and Quit Now
4	Write impromptu Restart

DEFUALTS:

Optional keyword. If it is not specified, then the user is prompted interactively which of the above four actions is to be taken. If *INTERRUPT is specified without an action, the default action is *RESTART-STOP.

CONDITIONS:

This keyword must appear in the Input/Output Control keyword group.

When a Control-C is encountered by the simulator during a run with an action of either *RESTART-STOP or *WRST-CONT, all required restart information is written out. In addition, grid, well and sector output will be written to both the output and SR2 files.

When this additional output information is written out, user defined output frequency controls remain unchanged and will behave as if the Control-C was not encountered.

ASCII Form of SR2 (Optional)

***SR2ASCII**

PURPOSE:

*SR2ASCII indicates that an ASCII copy of the main-results-file (mrf) will be created in addition to the binary file. The extension for this file is .asc

FORMAT:

*SR2ASCII

DEFAULTS:

Optional keyword. No Defaults

CONDITIONS:

This keyword must appear in the Input/Output Control keyword group.

Precision of SR2 (Optional)

*SR2PREC

PURPOSE

*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 of *SINGLE will cause the result of a restart run to vary slightly from the same timesteps of the first run, due to the fact that the lower order digits of the starting values have been truncated.

FORMAT:

*SR2PREC (*SINGLE | *DOUBLE)

DEFAULTS:

If *SR2PREC is absent, then *SR2PREC *DOUBLE is assumed.

CONDITIONS:

This keyword must appear in the Input/Output Control keyword group.

Controlling Data File Listing Using NOLISTLIM (Optional)

***NOLISTLIM**

PURPOSE:

*NOLISTLIM specifies that the default 20 lines of keyword echoing under the *LIST keyword be extended so that all values of the keyword are echoed regardless of the number of lines required to do so. This can create large files if used, but allows the user to view all of the data.

FORMAT:

*NOLISTLIM

DEFAULTS:

Optional keywords. Defaults to a list limit of 20 lines per keyword

CONDITIONS:

*NOLISTLIM must appear on a line by itself in the Input/Output Control Section.

EXPLANATION:

By default, the entire data file is listed to the output print file prior to the start of the simulation run. When keywords containing large amounts of data are listed under *LIST, only the first 20 lines of data are echoed. In order to echo all lines of data (this can result in large files) the *NOLISTLIM keyword in the Input/Output Control section is required.

See *LIST and *NOLIST in the Keyword Data Entry System section.

Input Data Units (Optional)

***INUNIT**

PURPOSE:

*INUNIT specifies the input data units.

FORMAT:

*INUNIT	(*SI) (*FIELD) (*LAB) (*MODSI)
----------------	---

DEFINITIONS:

***SI**

This option specifies SI units for input data.

***FIELD**

This option specifies FIELD units for input data.

***LAB**

This option specifies LAB units for input data.

Please note: Do not use the analytical aquifer model when using *LAB units. The use of minutes rather than days as a time unit makes the aquifer model invalid.

***MODSI**

This option specifies modified SI 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 input-data-file.

EXPLANATION:

Input data may be entered using one of the four options shown in the following Units Table.

Examples:

*INUNIT *FIELD

UNITS TABLE

	*SI	*FIELD	*LAB	*MODSI
Time	days	days	mins	days
Liquid Volumes	m ³	bbl	cm ³	m ³
Gas Volume	m ³	ft ³	cm ³	m ³
Pressure	kPa	psi	kPa	kg/cm ²
Absolute Permeability	md	md	md	md
Viscosity	mPa-s	cp	mPa-s	mPa-s
Mass Density	kg/m ³	lbm/ft ³	g/cm ³	kg/m ³
Length, Distance	m	ft	cm	m
Solution Gas/Oil Ratio	m ³ /m ³	scf/STB	cm ³ /cm ³	m ³ /m ³
Polymer Concentration	kg/m ³	lb/STB	g/cm ³	kg/m ³
Porosity	fraction	fraction	fraction	fraction
Saturation	fraction	fraction	fraction	fraction
Temperature	deg C	deg F	deg C	deg C
Oil, Water Formation				
Volume Factors	m ³ /m ³	RB/STB	cm ³ /cm ³	m ³ /m ³
Gas Expansion Factor	m ³ /m ³	scf/RB	m ³ /m ³	m ³ /m ³
Gas Formation				
Volume Factor	m ³ /m ³	RB/scf	cm ³ /cm ³	m ³ /m ³
Surface Tension	dynes/cm	dynes/cm	dynes/cm	dynes/cm

Here is a list of conversion units used by IMEX. To convert user input units to SI units multiply by the numbers given below the unit in the table. To convert from SI, multiply by the inverse of the numbers given in the table.

Liquid Volume				
User Unit	m^3	bbl	cm^3	m^3
Conversion Factor	1.0	0.1589873	1.0E-06	1.0
Gas Volume				
User Unit	m^3	ft^3	cm^3	m^3
Conversion Factor	1.0	0.02831685	1.0E-06	1.0
Pressure				
User Unit	kPa	psi	kPa	kg/cm^2
Conversion Factor	1.0	6.894757	1.0	98.0665
Viscosity				
User Unit	$\text{mPa}\cdot\text{s}$	cp	$\text{mPa}\cdot\text{s}$	$\text{mPa}\cdot\text{s}$
Conversion Factor	1.0	1.0	1.0	1.0
Permeability				
User Unit	md	md	md	md
Conversion Factor	1.0	1.0	1.0	1.0
Density				
User Unit	kg/m^3	lbm/ft^3	g/cm^3	kg/m^3
Conversion Factor	1.0	16.01846	1000.0	1.0
Length				
User Unit	m	ft	cm	m
Conversion Factor	1.0	0.3048	.01	1.0
Mass				
User Unit	kg	lbm	g	kg
Conversion Factor	1.0	0.4535924	.001	1.0
Time				
User Unit	days	days	minutes	days
Conversion Factor	1.0	1.0	6.944E-04	1.0
Solution Gas/Oil Ratio				
User Unit	m^3/m^3	scf/STB	cm^3/cm^3	m^3/m^3
Conversion Factor	1.0	0.1780444	1.0	1.0
Formation Vol. Factors (oil, water)				
User Unit	m^3/m^3	RB/STB	cm^3/cm^3	m^3/m^3
Conversion Factor	1.0	1.0	1.0	1.0
Gas Expansion Factor				
User Unit	m^3/m^3	scf/RB	cm^3/cm^3	m^3/m^3
Conversion Factor	1.0	0.178044	1.0	1.0

Output Data Units (Optional)

***OUTUNIT**

PURPOSE:

*OUTUNIT specifies the output data units.

FORMAT:

*OUTUNIT (*SI)
 (*FIELD)
 (*LAB)
 (*MODSI)

DEFINITIONS:

*SI

This option specifies SI units for output data.

*FIELD

This option specifies FIELD units for output data.

*LAB

This option specifies LAB units for output data.

*MODSI

This option specifies modified SI units for input data.

DEFAULTS:

Optional keyword. Default is *INUNIT.

CONDITIONS:

This keyword must appear in the Input/Output Control keyword group, at the start of the data file.

EXPLANATION:

Output data may be displayed using one of the four options shown in the Units Table (see *INUNIT)

Examples:

*OUTUNIT *FIELD

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 input-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 noncritical data range checking.

*RANGECHECK *OFF also will suppress the printing of all "warning" messages. Error messages will always be printed.

Examples:

*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 input-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 syntactical 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.

Examples:

```
** Stop simulation if 50 errors encountered.  
*MAXERROR 50
```

Restart Timestep (Optional)

***RESTART, *RESTIME, *RESDATE**

PURPOSE:

*RESTART, *RESTIME or *RESDATE indicates that this is a restart run.

FORMAT:

*RESTART num
*RESTIME time
*RESDATE date

DEFINITIONS:

num

An integer to specify the timestep number after which to restart the simulation.

time

A real number to specify the time from which the simulation run should restart from. Here the time defined is the same as on the *TIME keyword in the recurrent data you want to restart from.

date

A date from which the simulation run should restart. The date takes the form of

YYYY MM DD.DDDD

which is the same as on the *DATE keyword in the recurrent data you want to restart from.

DEFAULTS:

Optional keywords. If these keywords are not present in the input-data-file, then no restart records are read and the simulation begins at time zero.

If *RESTART is present, but no timestep number, the default is to restart at the last timestep in the input index-results-file. *RESTIME and *RESDATE must be followed by a corresponding argument.

CONDITIONS:

Restart keywords must appear in the Input/Output Control keyword group.

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 mid-point 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.

Examples:

```
** Restart at the 10th timestep.  
*RESTART 10  
  
** Restart at the last timestep in the restart file.  
*RESTART  
  
** Restart at simulation time 730.0 (time start from 0.0) in  
the restart file.  
*RESTIME 730.0  
  
** Restart at date June 25, 2010 in the restart file.  
*RESDATE 2010 06 25
```

Separate or Combined Restart/Graphics (Optional)

***RESTART_SR2**

PURPOSE

*Keyword to indicate whether graphics and restart data are combined into a main SR2 file or separated into two files (one for graphics and one for the restart record).

FORMAT:

***RESTART_SR2 (*SEPARATE | *MAIN)**

DEFINITIONS:

***SEPARATE**

Send graphics data into BASENAME.irf/BASENAME.mrf and restart data into BASENAME.rstr.irf/BASENAME.rstr.mrf.

***MAIN**

Send graphics and restart data into BASENAME.irf/BASENAME.mrf.

DEFAULTS:

If *RESTART_SR2 is not found in the data, *RESTART_SR2 *SEPARATE is assumed. By default, when separate files are used, graphics data is written out in single precision while restart data is written out in double precision.

CONDITIONS:

This keyword must appear in the Input/Output Control keyword group. Combined (*MAIN) or separated (*SEPARATE) sr2 files must be kept consistent between the initial and restart runs. The formats cannot be mixed between runs.

EXPLANATION:

The SR2 files contain both graphics and restart records. If the *MAIN option is used, both restart records and graphics information are stored in a single binary file (BASENAME.mrf) and its accompanying index file (BASENAME.irf). If the default, *SEPARATE option is used, the graphics files are separated from the restart records. The graphics are stored in BASENAME.irf and BASENAME.mrf while the restart records are stored in BASENAME.rstr.irf and BASENAME.rstr.mrf.

When the *SEPARATE option is used, by default, the graphics files are stored in single precision, while the restart records are stored in double precision. This was done to reduce both CPU time and storage. *SR2PREC can be used to override this behavior (i.e. to make both graphics and restart records double precision). The *SEPARATE restart SR2 file set rewinds its main data file (e.g., "BASENAME.rstr.mrf") when *REWIND is enabled; there is no .rff file (rewindable restart file) required.

When *SEPARATE is used, graphics data are read into RESULTS 3D or RESULTD GRAPH by choosing the BASENAME.irf files, exactly like when the *MAIN option is used. When restarting, the file BASENAME.rstr.irf is used in place of the BASENAME.irf file (as in *MAIN option), which no longer contains restart information.

Restart Record Writing Frequency (Optional)

*WRST, *REWIND

PURPOSE:

*WRST controls the frequency of writing restart records to the index-results-file and to either the main-results-file or the rewindable-results-file.

*REWIND controls the frequency of rewinding the rewindable-results-file.

FORMAT:

*WRST	(*FREQ-TIME freq) (*FREQ freq) (*TIME) (*TNEXT)
*REWIND	(num)

DEFINITIONS:

*FREQ-TIME

Subkeyword indicating that restart records are to be written at both the timestep frequency specified by 'freq' and the subsequent recurrent *TIME or *DATE keywords.

*FREQ

Subkeyword indicating that restart records are to be written at only the timestep frequency specified by 'freq'.

freq

An integer to specify the frequency (timesteps). Data is written at the given frequency. If 'freq' is missing after *FREQ or *FREQ-TIME then *TIME is the default action. If both *FREQ-TIME and *FREQ are missing and 'freq' appears by itself after *WRST then *FREQ-TIME is the default.

*TIME

Write data to the two (or three) results files every time specified by subsequent recurrent *TIME or *DATE keywords.

*TNEXT

Write data to the two (or three) results files on the next time specified by the next recurrent *TIME or *DATE keywords. No further output is written until the next *WRST card is read.

num

Frequency of rewinding the rewindable-restart-file. num is the maximum number of restart records allowed to accumulate in the rewindable-restart-file before it is rewound. num must be greater than 0. If num is 1, then only the last restart is available.

DEFAULTS:

Optional keywords. If the *WRST keyword is not present in the input-data-file, the default is no restart record written; if *WRST is present, the default is to write a restart at every time specified by subsequent recurrent *TIME or *DATE keywords in the input-data-file.

If *REWIND is not present the rewritable-restart-file is not used. 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.

EXPLANATION:

If *REWIND (num) is used, restart records are written not to the main-results-file but to the rewritable-results-file. This file is rewound after 'num' restart records are written into it. Thus at the end of the run it can contain up to 'num' restart records from which the simulation can be restarted.

Examples:

```
** Write restart record every 10 timesteps.  
*WRST 10  
** Write restart at the every time change.  
*WRST  
*REWIND 5 ** rewind the rewritable-results-file  
          ** after five restarts have been written.  
          ** Write restart on date Jan 01 2001  
*WRST *TNEXT  
*DATE 2001 01 01
```

Output Printing Frequency (Optional)

*WPRN

PURPOSE:

*WPRN controls the frequency of writing data to the output file.

FORMAT:

*WPRN	(*WELL)	(freq)
		(*TIME)
		(*TNEXT)
	(*GRID)	(freq)
		(*TIME)
		(*TNEXT)
	(*SECTOR)	(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-data-file.

*WELL *TNEXT

Write well results to the output file on the next time specified by the next recurrent *TIME or *DATE keywords in the input-data-file. No further output is written until the next *WPRN *WELL card is read.

*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-data-file.

***GRID *TNEXT**

Write grid results to the output file on the next time specified by the next recurrent *TIME or *DATE keywords in the input-data-file. No further output is written until the next *WPRN *GRID card is read.

***SECTOR freq**

Write sector, flux sector and lease line results to the output file every freq timesteps, where freq is an integer. If freq is zero, no sector and lease plane results are written.

***SECTOR *TIME**

Write sector, flux sector and lease line results to the output file at every time specified by subsequent recurrent *TIME or *DATE keywords in the input-data-file.

***SECTOR*TNEXT**

Write sector, flux sector and lease line results to the output file on the next time specified by the next recurrent *TIME or *DATE keywords in the input-data-file. No further output is written until the next *WPRN
*SECTOR card is read.

***ITER *ALL**

Provide full details of matrix solution and Newton iterations at every timestep.

***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:

Well results are written every timestep. Grid results are written every *TIME or *DATE card. Sector and lease plane results are written every *TIME or *DATE card. Timestep summary output defaults to *WPRN *ITER *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 output file may be changed during the simulation.

P/Z Sector Output:

In gas-water, black-oil and condensate models which specify a PVT region Temperature (*TRES) for each PVT region defined, average P/Z is output on a sector and field basis. Average P/Z is available weighted in each Sector by both pore volume and Hydrocarbon pore volume.

EXPLANATION:

Examples:

```
** Write well results at every time change.  
*WPRN *WELL *TIME  
  
** Write grid results every 10 timesteps.  
*WPRN *GRID 10  
  
** Write sector and lease plane results every time  
** change.  
*WPRN *SECTOR *TIME  
  
** Write details of Newton iterations  
** at every timestep.  
*WPRN *ITER *NEWTON  
  
** Write well results only on the next  
** time change or date card.  
*WPRN WELL *TNEXT
```

Items in Output Print File (Optional)

*OUTPRN

PURPOSE:

*OUTPRN identifies what information is written to the output file.

FORMAT:

*OUTPRN	(*WELL)	(*ALL) (*RESERVOIR) (*LAYER) (*BRIEF)
	(*GRID)	(*ALL) (*NONE) (*EXCEPT) (grid_list) (grid_list)
	(*TABLES)	(*ALL) (*NONE)
	(*RES)	(*ALL) (*NONE) (*EXCEPT) (res_list) (res_list)
	(*WELL-SECTOR)	(*SORT-WELL-NAME) (*SORT-WELL-NUM) (*NONE)
	(*FLUX-SECTOR)	(*ALL) (sc-rc selector) (ijk-con selector) (*INFLUX) (*INTERSEC) (*NONE)

DEFINITIONS:

*WELL

This subkeyword specifies that well results will be written to the output file.

*GRID

This subkeyword specifies that grid results will be written to the output file.

*TABLES

This subkeyword specifies that input property tables will be printed at the start of the simulation run.

*RES

This subkeyword specifies that input reservoir properties will be printed at the start of the simulation run. Original volumes in place will always be printed after input reservoir properties unless *RES *NONE is specified.

***WELL-SECTOR**

This keyword specifies that well in sector information will be printed out.

Well in sector information consists of two tables. The first is ordered by sectors (sector table) and lists all active wells within a sector (an active well has at least one perforation open). The second table is ordered by well (well table) and lists all sectors associated with each active perforation of every well.

Wells which have all perforations closed or have not had any perforations defined will not be listed in either table. Individual perforations that are no longer active will not be listed in the well table. Perforations which have had their production/injection zeroed due to use of the *XFLOW-MODEL *ZERO-FLOW keyword will not be listed in the well table for as long as their production/injection is zeroed. Active perforations of wells that are “shut in” will be listed in the well table, but “shut in” wells are not listed in the sector table.

The table information is printed out every timestep to a separate file with extension “sct”, the file will be created if when scanning the data IMEX finds the *OUTPRN *WELL-SECTOR keywords. The amount of printout can be controlled by using the *NONE subkeyword in recurrent data.

For example, the user can initially define *OUTPRN *WELL-SECTOR *NONE and have no printout until within the recurrent data a *OUTPRN *WELL-SECTOR keyword is found with subkeyword *SORT-WELL-NAME or *SORT-WELL-NUM. Output is again turned off at the next time/date the *OUTPRN *WELL-SECTOR *NONE keyword is found.

If *OUTPRN *WELL-SECTOR is not found, no tables are written. If *OUTPRN *WELL-SECTOR is found without subkeywords *SORT-WELL-NAME or *SORT-WELL-NUM, *SORT-WELL-NAME is the default.

This output option will normally be used only on selected runs to help debug well placement within sectors.

***FLUX-SECTOR**

This subkeyword specifies that flux sector summary will be printed out. The output frequency is controlled by *WPRN *SECTOR

***ALL**

Write all possible variables to the output file. For the *WELL option this means all of *BRIEF, *RESERVOIR and *LAYER. For *FLUX-SECTOR option, it means both influx and inter-sector flux will be printed out.

***RESERVOIR**

Write a summary of well variables at reservoir conditions to the output file. If available this will also cause voidage replacement information to be printed out at the field and group level

***LAYER**

Write a summary of layer variables at surface conditions to the output file.

***BRIEF**

Write a summary of well variables to the output file at surface conditions only. No layer information is printed.

***NONE**

No variables will be written to the output file.

***EXCEPT**

This subkeyword identifies an exception list of variables to be written to the output file (valid for *GRID only).

***SORT-WELL-NAME**

Sorts wells in both the sector and well tables in the “sct” file by well name (*OUTPRN *WELL-SECTOR *SORT-WELL-NAME)

***SORT-WELL-NUM**

Sorts wells in both the sector and well tables in the “sct” file by well number (*OUTPRN *WELL-SECTOR *SORT-WELL-NUM)

grid_list

A series of character strings to identify which variables to write (or not to write when used with *EXCEPT) to the output file (valid for *GRID only).

The grid_list for *GRID is:

*SO	Oil saturation
*SG	Gas saturation
*SW	Water saturation
*SS	Solvent saturation
*PCONC	Polymer concentration
*PADSORP	Polymer adsorption
*PRES	Pressure
*DATUMPRES	Datum pressure
*OILPOT	Oil potential
*BPP	Bubble point pressure
*SSPRES	Solvent saturation pressure
*WINFLUX	Aquifer water influx
*IMEXMAP	Implicit / explicit block map
*KRO	Oil relative permeability
*KRW	Water relative permeability
*KRG	Gas relative permeability
*KRS	Solvent relative permeability

*VISO	Oil viscosity
*VISW	Water viscosity
*VISG	Gas viscosity
*VISS	Solvent viscosity
*MASDENO	Oil density
*MASDENW	Water density
*MASDENG	Gas density
*MASDENS	Solvent density
*TRMOI	Oil phase transmissibility in the I-, J-, and K-direction respectively
*TRMOJ	
*TRMOK	
*SWCON	Connate water saturation S_{wcon}
*SWCRIT	Critical water saturation S_{wcrit}
*SOIRW	Irreducible oil saturation S_{oirw} (oil-water table)
*SORW	Residual oil saturation after water flood (S_{orw})
*SGCON	Connate gas saturation S_{gcon}
*SGCRIT	Critical gas saturation S_{gcrit}
*SLCON	Connate liquid saturation S_{lcon} (gas-liquid table)
*SOIRG	Irreducible oil saturation $S_{oирg}$ (gas-liquid table)
*SORG	Residual oil saturation after gas flood (S_{org})
*APIGRAV	Light oil volume fraction (STC) (API tracking option)
*POROS	Porosity at current pressure (includes compressibility/ compaction effects)
*KRSETN	Relative Permeability Set Number Map
*RFG	Gas resistance factor for non-Darcy flow (1+ Forchheimer number)
*RFO	Oil resistance factor for non-Darcy flow (1+ Forchheimer number)
*RFW	Water resistance factor for non-Darcy flow (1+ Forchheimer number)
*RFS	Solvent resistance factor for non-Darcy flow (1+ Forchheimer number)
*PCOW	Water-oil capillary pressure (water-gas capillary pressure when using *MODEL *GASWATER)
*PCOG	Oil-gas capillary pressure
*SBDZ	Subsidence
*EG	Gas expansion factor
*BO	Oil formation volume factor
*RS	Solution gas ratio
*RV	Condensate oil content
*DPP	Dew point pressure
*SEAWF	Seawater Volume Fraction (Seawater Injection Model Only)

The following keywords are not output automatically when *GRID *ALL or *GRID *EXCEPT is used. The keywords must appear in a grid_list.

*KRW SCL	Krw at irreducible oil (Soirw)
*KROWSCL	Krow at connate water (Swcon)
*KRG SCL	Krg at connate liquid (Slcon)
*KROGSCL	Krog at connate gas (Sgcon)
*PCWSCL or	Pcow, Pcwg or Pclg (or J Function equivalent) at connate
*PCWMAX	water (Swcon)
*PCWMIN	Pcow, Pcwg or Pclg (or J Function equivalent) at maximum
	water (1-Soirw)
*PCWIMIN	The minimum Pcow of boundary imbibition; for hysteresis
	with oil-trapping, the Pcow at the maximum residual oil
	(Sw=1-Sormax)
*PCGSCL or	Pcog (or J Function equivalent) at connate liquid (Slcon)
*PCGMAX	
*PCGMIN	Pcog (or J Function equivalent) at maximum liquid (1-Sgcon)
*PCGIMIN	The minimum Pcog of boundary imbibition
*SORMAX	Maximum residual oil saturation of the imbibition process
	(oil-water table, Krow hysteresis model or trapped oil
	hysteresis only)
*SORH	Current residual oil saturation of the imbibition process (oil-
	water table, Krow hysteresis model or trapped oil hysteresis
	only)
*GADSORB	Gas Adsorbed on Rock (Standard Vol/Bulk Rock Volume)
	Used in Gas Adsorption Model (Langmuir Isotherm)

res_list

A series of character strings to identify which variables to write (or not to write when used with *EXCEPT) to the output file (valid for *RES only).

The res_list for *RES is:

*GRNUM	Grid block numbers
*SECTOR	Sector numbers
*DI	I-direction grid block length
*DJ	J-direction grid block length
*DK	K-direction grid block length
*NTG	Net-to-gross ratio
*PAYDEPTH	Block center depth
*POR	Gross porosity
*PV	Pore volume
*HCPV	Hydrocarbon pore volume
*PERMI	Permeability
*PERMJ	in the I-, J-, and K-direction
*PERMK	respectively

*TRANSI	Transmissibility multipliers in the I-, J-, and K-direction respectively
*TRANSJ	
*TRANSK	
*DIFRAC	Fracture spacing for dual porosity or dual permeability system in the I-, J-, and K-direction respectively
*DJFRAC	
*DKFRAC	
*TRSMIS	Transmissibility
*RS	Solution gas R _s
*SWCON	Connate water saturation S _{wcon}
*SWCRIT	Critical water saturation S _{wcrit}
*SOIRW	Irreducible oil saturation S _{oirw} (oil-water table)
*SORW	Residual oil saturation after water flood (S _{orw})
*SORMAX	Maximum residual oil saturation of the imbibition process (oil-water table, Krow hysteresis model or trapped oil hysteresis only)
*SORH	Current residual oil saturation of the imbibition process (oil-water table, Krow hysteresis model or trapped oil hysteresis only)
*SGCON	Connate gas saturation S _{gcon}
*SGCRIT	Critical gas saturation S _{gcrit}
*SLCON	Connate liquid saturation S _{lcon} (gas-liquid table)
*SOIRG	Irreducible oil saturation S _{oirg} (gas-liquid table)
*SORG	Residual oil saturation after gas flood (S _{org})
*PVTSETN	PVT set number map
*DPO	Offsets for oil, water, and
*DPW	gas phase equilibrium with DEPTH_ave
*DPG	respectively
*NDBETCR	Non-Darcy correction factor
*WSRFTN	Water-oil (water-gas) surface tension term
*GSRFTN	Gas-oil surface tension term
*CRSETN	Compaction rock type (set number) map
*KRWCL	Krw at irreducible oil (Soirw)
*KROWSCL	Krow at connate water (Swcon)
*KRGCL	Krg at connate liquid (Slcon)
*KROGSCL	Krog at connate gas (Sgcon)
*PCWSCL	Pcow, Pcwg or Pclg (or J Function equivalent) at connate water (Swcon)
*PCGSCL	Pcog (or J Function equivalent) at connate liquid (Slcon)
*PCWSHF	Pcow (or Pcgw or Pclg) shift
*JFWSHF	Water J function shift
*FRWIDTHCOR	Fracture width correction for velocity dependent polymer-water viscosity table
*PTHRESHI	Pressure gradient threshold in the I direction
*PTHRESHJ	Pressure gradient threshold in the J direction
*PTHRESHK	Pressure gradient threshold in the K direction
*ROCKDEN	Intrinsic rock density used in gas adsorption model

*ADGMAXV	Langmuir Isotherm model: the maximum volume of adsorbed gas per unit mass of rock
*ADGCSTV	Langmuir Isotherm model: inverse pressure constant
*ADGPCRIT	Langmuir Isotherm model: critical adsorption pressure
sc-rc selector	
*SC	Flux sector summary contains values in surface condition
*RC	Flux sector summary contains values in reservoir condition

If neither present, both *SC and *RC are selected.

ijk-con selector

*SUM	Flux sector summary contains summation of connections in all three directions
*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 it is not present in the input-data-file, the defaults are:

*OUTPRN	*WELL	*BRIEF
*OUTPRN	*GRID	*SO *SG *SW *SS *PCONC *PADSORP *PRES *OILPOT *BPP *SSPRES *WINFLUX *IMEXMAP
*OUTPRN	*TABLES	*ALL
*OUTPRN	*RES	*SECT *DI *DJ *DK *NTG *PAYDEPTH *POR *PERMI *PERMJ *PERMK *TRANSI *TRANSJ *TRANSK
*OUTPRN	*WELL-SECTOR	*NONE
*OUTPRN	*FLUX-SECTOR	*NONE

CONDITIONS:

The keywords *OUTPRN *WELL, *OUTPRN *GRID *OUTPRN *WELL-SECTOR and *OUTPRN *FLUX-SECTOR 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 files may be changed during the simulation.

EXPLANATION:

An example of *OUTPRN *GRID, when using the list option is:

```
*OUTPRN *GRID *SO *SG *SW *PRES *OILPOT *BPP *IMEXMAP
*OUTPRN *WELL *BRIEF will only print the surface information.
*OUTPRN *WELL *LAYER will print both the surface and the layer
information.
*OUTPRN *WELL *RESERVOIR will print both the surface and reservoir
information.
*OUTPRN *WELL *ALL will print the surface, the reservoir and
the layer information.
```

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.

Simulation Results File Writing Frequency (Optional) *WSRF

PURPOSE:

*WSRF controls the writing of well and/or grid information to the index-results-file and the main-results-file (the SR2 file system).

FORMAT:

```
*WSRF (*WELL) (freq)
              (*TIME)
              (*TNEXT)
(*GRID)      (freq)
              (*TIME)
              (*TNEXT)
(*SECTOR)    (freq)
              (*TIME)
              (*TNEXT)
```

DEFINITIONS:

*WELL freq

Write well results to the SR2 file system every freq timesteps, where freq is an integer. If freq is zero the well results are not written to the SR2 file system.

*WELL *TIME

Write well results to the SR2 file system at every time specified by subsequent recurrent *TIME or *DATE keywords in the input-data-file.

*WELL *TNEXT

Write well results to the SR2 file system on the next time specified by the next recurrent *TIME or *DATE keywords in the input-data-file. No further output is written until the next *WSRF *WELL card is read.

*GRID freq

Write grid results to the SR2 file system every freq timesteps, where freq is an integer. If freq is zero the grid results are not written to the SR2 file system.

*GRID *TIME

Write grid results to the SR2 file system at every time specified by subsequent recurrent *TIME or *DATE keywords in the input file.

*GRID *TNEXT

Write grid results to the SR2 file system on the next time specified by the next recurrent *TIME or *DATE keywords in the input-data-file. No further output is written until the next *WSRF *GRID card is read.

***SECTOR freq**

Write sector, flux sector and lease line results to the SR2 file system every freq timesteps, where freq is an integer. If freq is zero, sector results are written only at time 0.

***SECTOR *TIME**

Write sector, flux sector and lease line results to the SR2 file system at every time specified by subsequent recurrent *TIME or *DATE keywords in the input file.

***SECTOR *TNEXT**

Write sector, flux sector and lease line results to the SR2 file system on the next time specified by the next recurrent *TIME or *DATE keywords in the input-data-file. No further output is written until the next *WSRF *SECTOR card is read.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the defaults are:

*WSRF	*WELL	1
*WSRF	*GRID	*TIME
*WSRF	*SECTOR	*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 SR2 file system may be changed during the simulation.

EXPLANATION:

*WSRF controls the writing of well, grid, sector and lease plane information to the SR2 file system. The SR2 files system consists of up two three files: the index-results-file, the main-results-file and the rewritable-results-file. These files are required for restart runs and by the graphics post-processor "RESULTS".

The SR2 files also contain the grid information related to sectors defined on the *SECTOR or *SECTORARRAY cards. Thus it is possible to use "RESULTS" to probe which sectors a block is in. In addition it is possible to use "RESULTS" to only view blocks belonging to a specified sector. The blocks not in the specified sector would be invisible.

P/Z Sector Output:

In gas-water, black-oil and condensate models which specify a PVT region Temperature (*TRES) for each PVT region defined, average P/Z is output on a sector and field basis. Average P/Z is available weighted in each Sector by both pore volume and Hydrocarbon pore volume.

Examples:

```
*WSRF *WELL 1
*WSRF *GRID *TIME
*WSRF *SECTOR *TNEXT
```

Items in Simulation Results File (Optional)

*XDR, *OUTSRF

PURPOSE:

*OUTSRF identifies what information is written to the index-results-file and the main-results-file (the SR2 file system).

FORMAT:

*XDR (*ON *OFF)	(*DOWNHOLE)
*OUTSRF (*WELL)	(*BLOCKP) (*ON *OFF)
	(*LAYER) (*ALL) (*DOWNHOLE)
	(*NONE)
	*WELLIPI
	*PDRAIN
*OUTSRF *WELLRATES	(*ON *OFF)
	(*ALL)
	(*NONE)
	(*EXCEPT) (grid_list)
	(grid_list)
(*RES)	(*ALL *NONE)
(*SPECIAL	i j k grid_list)
(*FLUX-SECTOR)	(*ALL *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 SR2 is in XDR format, then the keyword "XDR" will appear near the top of the index file (irf).

*GRID

This subkeyword specifies that grid results will be written to the SR2 file system.

*WELL

This subkeyword specifies that well results will be written to the SR2 file system.

*RES

This subkeyword specifies that input reservoir properties will be printed at the start of the simulation run.

***DOWNHOLE**

This subkeyword specifies that well performance values will be written to the SR2 file system at reservoir conditions in addition to surface conditions. If available voidage replacement ratios will be recorded to the SR2 file system for display using Results Graph.

***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.

***PDRAIN**

Specification of *OUTSRF *WELL *PDRAIN will turn on a more involved calculation of well productivity / injectivity index (*WELLPI) at surface conditions for oil, gas and water streams for all wells based on a selected drainage area estimation technique (see *PDRAIN-METHOD in the WELL AND RECURRENT DATA keyword group). Without *PDRAIN, *WELLPI is based on the pressure of the blocks the well passes through, hence the drainage radius is the “Peaceman” effective radius.

Generally, the well PI for a given surface stream is the ratio of the stream's surface rate divided by the pressure drop ($P_d - P_{wf}$). The *PDRAIN option will calculate P_d (a proxy for drainage pressure) by a spatial average pressure of (1) blocks which the well passes through and (2) blocks which are areally adjacent to the well. The user has the option of choosing either the well transmissibility or the grid block pore volumes for weighting the spatial average (See explanation in *PDRAIN-METHOD located in the WELL AND RECURRENT DATA keyword group). The values are written to the RESULTS output files and replace the normally output *WELLPI productivity indexes in the SR2.

***WELLRATES**

*WELLRATES determines if well rates are written directly to the SR2, the default is to write both cumulatives and rates to the SR2. If *WELLRATES *OFF is specified, rates are derived from cumulatives.

***LAYER**

This subkeyword specifies that well performance values will be written for all layers of wells specified to the SR2 file system. The default is *NONE (see DEFAULTS below). Using *ALL specifies that layer information will be written for all wells to the SR2 file system.

Use of this option, especially with *ALL, can increase the size of the SR2 file substantially.

***BLOCKP**

The pressure of the block containing the bottomhole layer as defined in the description for keyword *PERF. If *LAYER is enabled, then block pressure of each layer is written as well.

***SPECIAL**

This subkeyword specifies a grid block location and a variable. The grid variable value at the given location is written to the SR2 file system.

***FLUX-SECTOR**

This subkeyword specifies that flow values of flux sectors will be written out. The output frequency is controlled by *WSRF *SECTOR. The output can be found under the LEASE origin in RESULTS GRAPH.

If *FLUX-SECTOR *ALL is specified, both influx and inter-sector flux will be written out. *FLUX-SECTOR *NONE indicates nothing will be written to the SR2 file except if there is *LEASE keyword defined. Please see CONDITIONS below for details.

***ALL**

Write all possible variables to the SR2 file system.

***NONE**

No variables will be written to the SR2 file system.

***EXCEPT**

This subkeyword identifies an exception list of variables to be written to the SR2 file system (valid for *WELL and *GRID only).

i j k

Grid block location of *SPECIAL variable.

grid_list

A series of keywords to identify which grid variables to write (or not to write when used with *EXCEPT) to the SR2 file system (valid for *GRID only). For *OUTSRF *SPECIAL the grid_list can contain only one variable.

Valid keywords in the grid_list for *GRID include:

*SO	Oil saturation
*SG	Gas saturation
*SW	Water saturation
*SS	Solvent saturation
*PCONC	Polymer concentration
*PADSORP	Polymer adsorption
*PRES	Pressure
*DATUMPRES	Datum pressure
*OILPOT	Oil potential
*BPP	Bubble point pressure
*SSPRES	Solvent saturation pressure
*WINFLUX	Aquifer water influx
*KRO	Oil relative permeability
*KRW	Water relative permeability
*KRG	Gas relative permeability
*KRS	Solvent relative permeability
*VISO	Viscosity
*VISW	Water viscosity
*VISG	Gas viscosity
*VISS	Solvent viscosity
*MASDENO	Oil density
*MASDENW	Water density
*MASDENG	Gas density
*MASDENS	Solvent density
*SWCON	Connate water saturation S_{wcon}
*SWCRIT	Critical water saturation S_{wcrit}
*SOIRW	Irreducible oil saturation S_{oirw} (oil-water table)
*SORW	Residual oil saturation after water flood (S_{orw})
*SGCON	Connate gas saturation S_{gcon}
*SGCRIT	Critical gas saturation S_{gcrit}
*SLCON	Connate liquid saturation S_{lcon} (gas-liquid table)
*SOIRG	Irreducible oil saturation $S_{oирg}$ (gas-liquid table)
*SORG	Residual oil saturation after gas flood (S_{org})
*APIGRAV	Light oil volume fraction (STC) (API tracking option)
*POROS	Porosity at current pressure (includes compressibility compaction effects)
*KRSETN	Relative permeability set number map
*PERM	Permeability in each direction. This option is only available when compaction is used and permeability multipliers vary with pressure or when permeability is entered in the recurrent data section
(*PERMEABILITY)	

FLOW VECTORS

*FLUXSC Flux and velocity vectors of oil, water and gas at surface conditions and/or reservoir conditions.

(*FLUX)
*FLUXRC

*VELOCSC
(*VELOC or *VELOSC)
*VELOCRC
(*VELORC)

Due to the large amount of data required to construct flow vectors in Results, it is recommended that only a single flux or velocity at either surface or reservoir conditions be output. To reduce file size, flow output is not, by default, included when *GRID *ALL (or EXCEPT) is chosen. Flow Vector output must be explicitly selected.

Due to the large number of calculations required to recalculate flow vectors it is also recommended that the *WSRF *TNEXT option be used to explicitly select (and hopefully reduce) the timesteps when Flow Vectors are written to the SR2.

STREAMLINES

*STRMLN

Triggers the output of Block Face based phase velocities to be used to construct streamlines in Results3D.

To reduce SR2 file size, face velocity output is not, by default, included when *GRID *ALL (or EXCEPT) is chosen.

Streamline output must be explicitly selected.

Due to the large number of calculations required to calculate face velocities it is also recommended that the *WSRF *TNEXT option be used to explicitly select (and hopefully reduce) the timesteps when Streamline information is written to the SR2. Gas resistance factor for non-Darcy flow (1+ Forchheimer number)

*RFG

*PCOW

*PCOG
*SBDZ
*EG
*BO
*RS
*RV
*DPP
*SEAWF
*FLUXCON

Water-oil capillary pressure (water-gas capillary pressure when using *MODEL *GASWATER)

Oil-gas capillary pressure

Subsidence (per block and summed over layers)

Gas expansion factor

Oil formation volume factor

Solution gas ratio

Condensate oil content

Dew point pressure

Seawater Volume Fraction (Seawater Injection Model Only)
Output Connection Based Flux to SR2. This information can be accessed using ReportWriter using the CONNECTION-FOR keyword. Oil flux is obtained using

Connection-for 'FLUXCONO' "Output-Time". Water and gas flux is obtained using the 'FLUXCONW' and 'FLUXCONG' keywords respectively. Due to the large number of calculations required to calculate face velocities it is also recommended that the *WSRF *TNEXT option be used to explicitly select (and hopefully reduce) the timesteps when FLUXCON information is written to the SR2.

The following keywords are not output automatically when *GRID *ALL or *GRID *EXCEPT is used. The keywords must appear in a grid_list.

*KRWSDL	Krw at irreducible oil (Soirw)
*KROWSCL	Krow at connate water (Swcon)
*KRGSCL	Krg at connate liquid (Slcon)
*KROGSCL	Krog at connate gas (Sgcon)
*PCWSCL or *PCWMAX	Pcow, Pcwg or Pclg (or J Function equivalent) at connate water (Swcon)
*PCWMIN	Pcow, Pcwg or Pclg (or J Function equivalent) at maximum water (1-Soirw)
*PCWIMIN	The minimum Pcow of boundary imbibition; for hysteresis with oil-trapping, the Pcow at the maximum residual oil (Sw=1-Sormax)
*PCGSCL or *PCGMAX	Pcog (or J Function equivalent) at connate liquid (Slcon)
*PCGMIN	cog (or J Function equivalent) at maximum liquid (1-Sgcon)
*PCGIMIN	The minimum Pcog of boundary imbibition
*SORMAX	Maximum residual oil saturation of the imbibition process (oil-water table, Krow hysteresis model or trapped oil hysteresis only)
*SORH	Current residual oil saturation of the imbibition process (oil-water table, Krow hysteresis model or trapped oil hysteresis only)

sc-rc selector

*SC	Flux sector summary contains values in surface condition
*RC	Flux sector summary contains values in reservoir condition

If neither present, both *SC and *RC are selected.

ijk-con selector

*SUM	Flux sector summary contains summation of connections in all three directions
*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 it is not present in the input-data-file, the defaults are:

```
*OUTSRF    *GRID    *SO *SG *SW *SS *PCONC *PADSORP
           *PRES *OILPOT *BPP *SSPRES
           *WINFLUX
*OUTSRF *WELL *LAYER *NONE
```

```
*OUTSRF *RES *ALL  
*XDR *ON  
*OUTSRF *FLUX-SECTOR *ALL *SC *SUM
```

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. *OUTSRF *RES, *OUTSRF *SPECIAL and *OUTSRF *FLUX-SECTOR cannot appear in the Recurrent Data section of the input-data-file.

If *LAYER or *BLOCKP well information is requested then mobility weighted datum pressure is also output.

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.

*TRMOI, *TRMOJ, and *TRMOK are not available for output to the SR2 file using the *OUTSRF keyword. *TRMOI, *TRMOJ, and *TRMOK are only available for display in the output file using the *OUTPRN keyword

The *PERM and all of the flow vector keywords cannot be used as subkeywords of the *SPECIAL option. *PERM is used to view permeabilities which vary with pressure (see the compaction keyword *CROCKTAB and *CROCKTABH) or with time if permeability is redefined in the recurrent data section. If permeability does not vary with pressure or has not been redefined in the recurrent data section *PERM will produce no output. *PERMI, *PERMJ and *PERMK in the res_list can be used to view constant permeabilities.

*APIGRAV, *RFG, oil transmissibility arrays (*TRMOI, *TRMOJ, *TRMOK), saturation endpoint arrays (*KRSETN, *SWCON, *SWCRIT, *SGCON, *SGCRIT, *SOIRW, *SORW, *SLCON, *SORG, *SOIRG) and relative permeability endpoint arrays (*KRWSC, *KROWSCL, *KRGSC, *KROGSCL, *PCWSCL and *PCGSCL) cannot be used as subkeywords of the *SPECIAL option.

Because flux sectors and lease lines use the same LEASE origin in the SR2, the output control for flux sector affects both. If a lease line is defined in the data set, it automatically switches on *OUTSRF *FLUX-SECTOR *ALL with *SC *SUM being selected. On other hand, if *RC, *ICON, *JCON and *KCON are selected for flux sectors, corresponding values for lease lines will be output as zero.

EXPLANATION:

An example of *OUTSRF, when using the list option is:

```
*OUTSRF *GRID *SO *SG *SW *PRES *OILPOT *BPP
```

Items in Diary File (Optional)

*OUTDIARY

PURPOSE:

*OUTDIARY controls some of the information written to the output-diary-file.

FORMAT:

*OUTDIARY (*BRIEF | *MAXLOC | *WELLSTATUS)
(*PRESAQ | *PRESNOAQ)
(*HEADER freq)

DEFINITIONS:

*BRIEF

This subkeyword specifies that the grid block location (indices) where maximum saturation or pressure changes occur will not be written to the output-diary-file. The oil/gas/water production/injection rates will be written with up to 8 significant figures. The water-cut will be written with up to 5 significant figures.

*MAXLOC

This subkeyword specifies that the grid block location (indices) where maximum saturation or pressure changes occur will be written to the output-diary-file. The oil/gas/water production/injection rates will be written with up to 5 significant figures. The water-cut will be written with up to 3 significant figures.

*WELLSTATUS

This subkeyword specifies that the number of producing wells, injecting wells, and shut in wells and the maximum changes of saturations and pressure will be written to the output-diary-file. The oil/gas/water production/injection rates will be written with up to 6 significant figures. The water-cut will be written with up to 3 significant figures.

*PRESAQ

This subkeyword specifies that the total pore volume average pressure, including the aquifer blocks, will be written to the output-diary-file.

*PRESNOAQ

This subkeyword specifies that the total pore volume average pressure, excluding the aquifer blocks (blocks where water saturation equals 1), will be written to the output-diary-file.

*HEADER

This subkeyword writes the table header to the output-diary-file every freq timesteps. If *HEADER is not present the table header will be written every 20 timesteps.

freq

This integer specifies the number of timesteps after which the table header will be written to the output-diary-file. If this integer is missing then the table header will be written every 20 timesteps.

DEFAULTS:

Optional keywords. If not present in the input-data-file, the defaults are:

*OUTDIARY *BRIEF *PRESAQ *HEADER 20

CONDITIONS:

This keyword may appear in the Input/Output Control keyword group and the Recurrent Data section.

EXPLANATION:

When the keyword *MAXLOC is present the grid blocks locations where maximum saturation and pressure changes occur are written to the diary file. Then, due to space limitations, the oil, gas and water production and injection rates and the water-cut have to be written with less significant figures.

Only one of the three keywords *BRIEF, *MAXLOC and *WELLSTATUS should be used for a given timestep. Likewise, *PRESAQ and *NOPRESAQ are exclusive.

The table header is written to the output-diary-file every "freq" timesteps or whenever the subkeywords *BRIEF *MAXLOC *WELLSTATUS *PRESAQ or *PRESNOAQ are changed.

Examples:

```
** Write grid block locations with maximum changes
** to the output-diary-file.
*OUTDIARY *MAXLOC

** Write table header after 50 timesteps.
*OUTDIARY *HEADER 50

** Average pressure does not include aquifer blocks.
*OUTDIARY *PRESNOAQ

** Write well status of the entire field
*OUTDIARY *WELLSTATUS
```

Production Split Output (Optional)

***PSPLIT**

PURPOSE:

*PSPLIT Turns on limited production split output.

FORMAT:

*PSPLIT (*ON)
(*OFF)

DEFINITIONS:

***PSPLIT**

Turns on the output of surface production split information to the output and SR2 files.

***PSPLIT *ON**

Explicitly turns on the output of surface production split information to the output and SR2 files.

***PSPLIT *OFF**

Explicitly turns off the output of surface production split information to the output and SR2 files. (Default)

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is not to print the split information.

CONDITIONS:

This keyword may appear in the Input/Output Control keyword group, but not in the recurrent data section, as it must not be changed during a simulation.

EXPLANATION:

The use of surface production streams divides up oil and gas phase information into surface split based information. The splits are defined below.

1. Black Oil Split - Liquid at surface, Liquid at reservoir conditions
2. Condensate Split - Liquid at surface, Gas at reservoir conditions
3. Free Gas Split - Gas at surface, Gas at reservoir conditions
4. Solution Gas Split - Gas at surface, Liquid at reservoir conditions

For a black oil model there are two splits reported: the solution gas split and the free gas split (which sum to the total gas produced at surface). For the condensate option, all four of the splits listed above are reported.

Production splits from wells and from sectors (sum of wells in sector) are output to the ascii and SR2 output files.

Debug Output (Optional)

*DEBUG

PURPOSE:

*DEBUG identifies what debug information is written to the output file.

FORMAT:

```
*DEBUG  (*JDUMP ts_range iter_range block_range)
        (*VDUMP ts_range)
        (*WELMAN *ON | *OFF | *WELLBORE)
        (*PDUMP *ON | *OFF)
        (*ECHO-RESTART *ON | *OFF)
        (*SMOOTHKRPC*ON | *OFF)
        (*XFLOW *ON | *OFF)
        (*PARAMS)
        (LSOLVER *ON | *OFF)
```

DEFINITIONS:

*JDUMP

Outputs the entire Jacobian Matrix in a format that the AIMSOL driver can read.

ts_range

Range of timesteps in the format i1:i2, where the Jacobian or grid map is to be printed.

iter_range

Range of Newtonian iterations in the format i1:i2 where the Jacobian matrix is to be printed.

block_range

Range of grid blocks in the format i1:i2, for which the Jacobian is to be printed.

*VDUMP

This keyword dumps the grid map from subroutine OUTIN after each Newtonian iteration. This is very useful for tracking down convergence problems.

*PDUMP

This keyword dumps pointer information form the Grid Module. Connection transmissibility, connection number and indices of connected blocks are dumped along with other block related information.

*WELMAN

This keyword activates the debug information from the well-management module.

***ON**

Turn debug information on.

***OFF**

Turn debug information off.

***WELLBORE**

Print detailed information on wellbore hydraulic calculations.

***ECHO-RESTART**

This keyword activates the keyword echo from the index-results-file during restart.

***ON**

Turn debug information on.

***OFF**

Turn debug information off.

***SMOOTHKRPC**

This keyword activates the debug information for *SMOOTH options (*SWT, *SLT, and *SGT keywords). If this is enabled, then three files are written:

*SMOOTHKRPC-IN, *SMOOTHKRPC-OUT and *SMOOTHKRPC-SUM.

These three files have default endings of .kri, .kro and .krs. The .kri file contains the input relative permeability and capillary pressure tables. The .kro file contains the internally calculated tables. The .krs file contains a summary of the *SMOOTH calculations. If *OUTPRN *TABLES *ALL is enabled (the default) and *SMOOTHKRPC is enabled then both the input tables and the internally calculated tables are written to the output file. If *OUTPRN *TABLES *ALL is enabled (the default) and *SMOOTHKRPC is not enabled then only the internally calculated tables are written to the output file.

***ON**

Turn debug information on.

***OFF**

Turn debug information off.

***XFLOW**

Activated debug information for the *XFLOW-MODEL keyword. Cycle by cycle information on layer crossflow status is printed out if debug information is turned on

***ON**

Turn debug information on.

***OFF**

Turn debug information off.

***PARAMS**

When using run time dimensioning, print out the 'Virtual' parameter file generated for IMEX. These values are calculated and used to run time dimension IMEX and determine common block size. Use of this keyword and use of the *DIM keyword can allow the user to tune the dimensioning of the IMEX simulator.

***LSOLVER**

Turn on summary output of the number of solver failures.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the defaults are 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, when using the *JDUMP option is:

```
** Dump derivatives at timestep 7 for iterations
** 2 and 3; blocks 244, 245 and 246
*DEBUG *JDUMP 7 2:3 244:246
```

This keyword must be in the Input/Output Control section or the Recurrent Data section. There may be more than one *DEBUG *JDUMP keywords.

An example of *DEBUG, when using the *VDUMP option is:

```
** Dump primary variables at timesteps 7, 8 and 9
** during Newtonian iterations.
*DEBUG *VDUMP 7:9
```

This keyword must be in the Input/Output Control section or the Recurrent Data section. There may be more than one *DEBUG *VDUMP keywords.

Reservoir Description

Fundamental Grid Definition (Required)

*GRID

PURPOSE:

*GRID defines the fundamental (main) grid, and marks the beginning of the reservoir description.

FORMAT:

*GRID	(*CART)	$n_i n_j n_k$
	(*VARI)	$n_i n_j n_k$
	(*CORNER)	$n_i n_j n_k$
	(*RADIAL)	$n_i n_j n_k$ (*RW rw)

DEFINITIONS:

*CART

Keyword indicating a rectangular Cartesian grid, with uniform depth/uniform thickness layers.

*VARI

Keyword indicating a rectangular grid allowing variable depth/variable thickness layers.

*CORNER

Keyword indicating a corner point grid, as described in the following.

*RADIAL

Keyword indicating a radial-angular cylindrical grid.

$n_i n_j n_k$

Number of grid blocks in the I, J and K directions.

*RW rw

The radius of the wellbore (m | ft | cm) located in the innermost radial grid block(s) for the *RADIAL grid. 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 $n_i \times n_j \times n_k$ 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.

A wellbore volume of radius rw is removed from the innermost ring in all layers of *RADIAL grids. This gives the innermost ring a circular inner boundary that connects to the wellbore. The wellbore volume is not included in the innermost rings.

I, J, and K indices are used to identify the blocks, where I runs in the range of 1 to n_i , J runs in the range of 1 to n_j and K runs in the range of 1 to n_k . The notation (I,J,K) will sometimes be used to denote a block. 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)

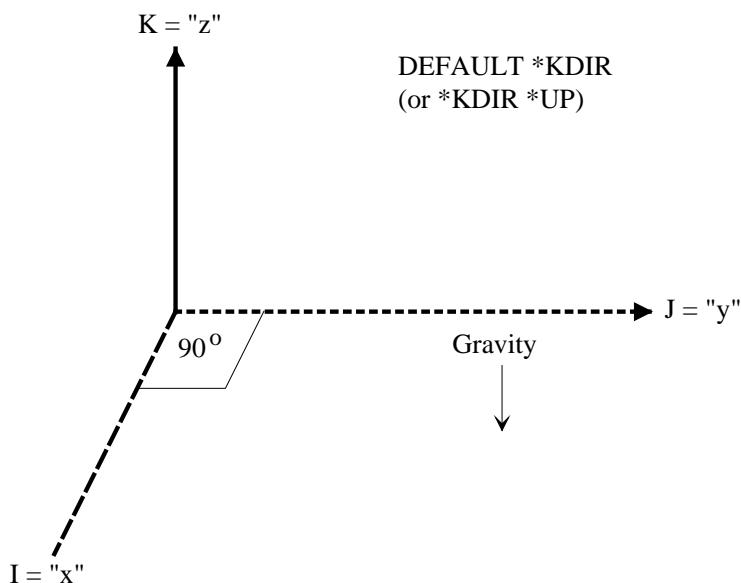
The default cases in the above table can be modified by use of the *KDIR *DOWN keywords.

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).

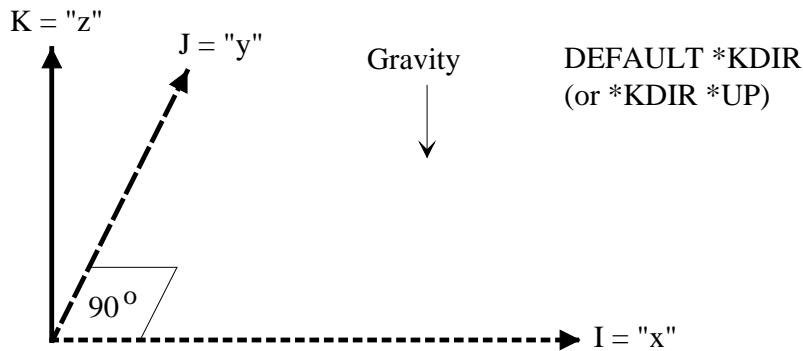
Rectangular:

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.

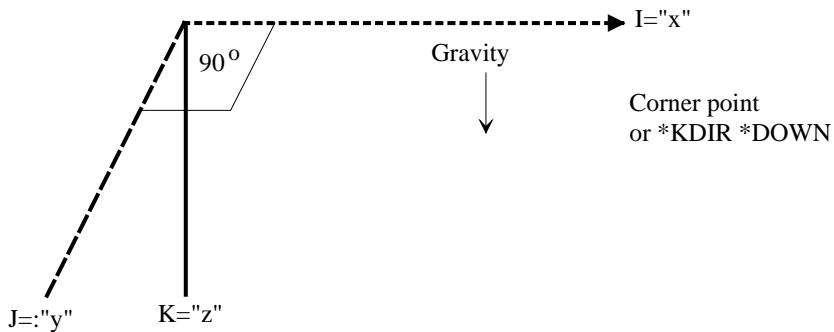
The coordinate system for a Cartesian or variable depth/variable thickness grid with the no KDIR specified (default), or *KDIR *UP specified, appears as follows:



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:

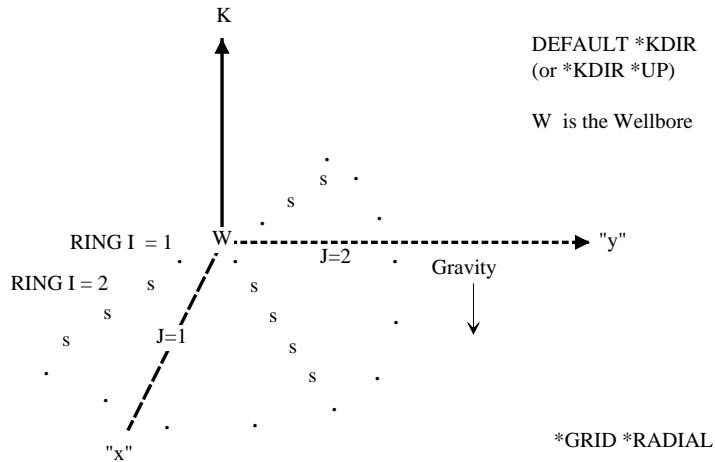


Radial:

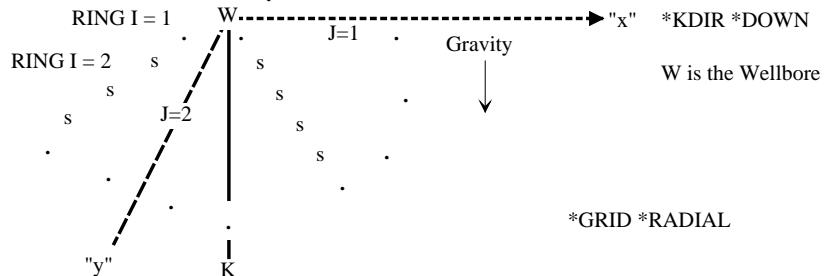
*RADIAL radial-angular cylindrical grids have rings indexed by I, counting from the innermost outwards. Each ring may be split into n_j sectors indexed by J, counting counterclockwise around the K axis with the first sector lying over the I axis. Layers are indexed by K, starting at the reservoir bottom 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 n_j 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:



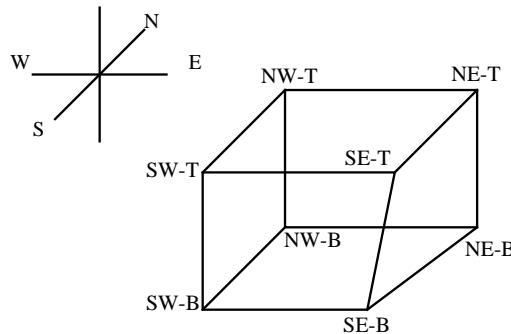
Grid block orderings are further described in Appendix D. FIGURE 1a shows Cartesian and variable depth/variable thickness grid systems that are NOT corner point grids, FIGURE 1b shows a corner point grid, and FIGURE 2 is for a *RADIAL radial-angular cylindrical system. The grid block nodes are the points where the pressures are evaluated. They are placed at the block centres for rectangular grids. The nodes for *RADIAL blocks are placed in accordance with an assumption made about the discretization of the accumulation terms in the simulator. The *RADIAL node is chosen as the point where the block pressure is equal to its volume-weighted average for the steady-state flow of a radially-directed incompressible fluid through the block. Details can be found in O.A. Pedrosa and K. Aziz, "Use of a Hybrid Grid in Reservoir Simulation", SPE, November 1986, pp. 611-621.

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:

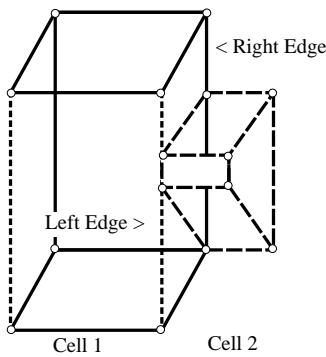


The block is the volume contained within the 6 faces, where each face is made by connecting the corner points with line segments as shown, and filling in the faces with a nonlinear (bilinear) interpolation. Faces need not be planar. The numerical results will be better if the cell shape is kept close to a rectangle however. The cell's barycentre is used as the pressure node.

IMEX 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 with the shared edges marked:



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:

- Rectangular Cartesian grid with ten blocks in the "x" direction, five blocks in the "y" direction, and four blocks in the "z" direction ($n_i = 10, n_j = 5, n_k = 4$). Enter:
`*GRID *CART 10 5 4`
- 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 ($n_i = 10, n_j = 3, n_k = 1$). Enter:
`*GRID *VARI 10 3 1`
- Corner point grid with 20 blocks in the "x" direction, 20 blocks in the "y" direction, and 5 layers ($n_i = 20, n_j = 20, n_k = 5$). Enter:
`*GRID *CORNER 20 20 5`
- Radial-angular cylindrical grid with 15 blocks in the radial direction, no radial subdivisions, and 5 layers ($n_i = 15, n_j = 1, n_k = 5$). Enter:
`*GRID *RADIAL 15 1 5`

K Direction Index (Optional)

*KDIR

PURPOSE:

*KDIR controls whether increasing K means going deeper or shallower in the reservoir.

FORMAT:

*KDIR (*UP)
(*DOWN)

DEFINITIONS:

*UP

Indicates that the K index increases upward, so larger K means shallower grid blocks.

*DOWN

Indicates that the K index increases downward, so larger K means deeper grid blocks.

DEFAULTS:

Optional keyword. Default is *KDIR *UP, except for corner point grids when *KDIR *DOWN is the default. (The corner point grid setting can not be changed.)

CONDITIONS:

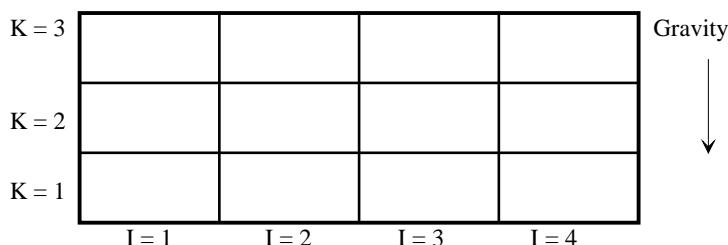
This keyword, if present, must be in the Reservoir Description keyword group.

EXPLANATION:

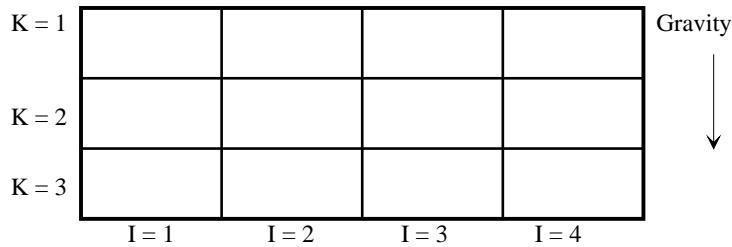
By default, the K index increases upward, except for corner point grids, when it increases downwards. See the *GRID keyword discussion for more details.

Examples:

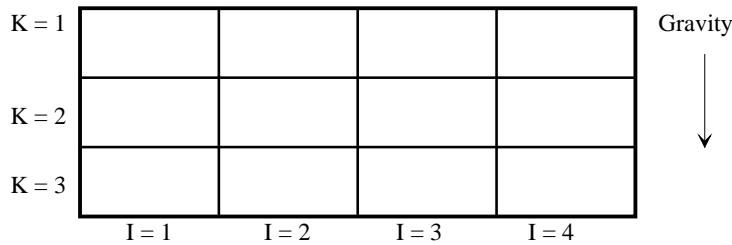
*KDIR *UP



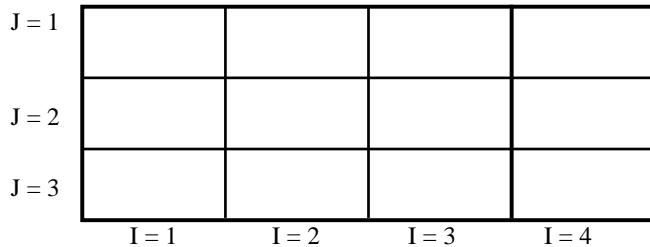
*KDIR *DOWN



*KDIR *DOWN (I-K View)



(I-J View)



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. The most commonly used array reading subkeywords are the *IVAR and *CON options.

EXPLANATION:

The keyword *DI defines the dimensions of the grid blocks in the I direction.

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:

- I direction grid increments for a Cartesian grid with $n_i = 10$ are:

1000,1000,1500,400,400,400,400,1000,1000

Use:

*DI *IVAR 2*1000 1500 5*400 2*1000

- I direction grid increments for a Cartesian grid are all 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 $n_i = 10$ 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 rw and its outer radius is $rw+2$.

The acceptable range of values for block lengths in the I direction is:

	SI m	Field ft	Lab cm
min	1.0E-4	3.28E-4	1.0E-2
max	1.0E+20	3.28E+20	1.0E+22

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:

The keyword *DJ defines the dimensions of the grid blocks in the J direction. Note that angular extents should be expressed in degrees.

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 $n_j=10$ 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 ($n_j = 1$). Use:

```
*DJ *CON 360
```

The acceptable range of values for block lengths in the J direction is:

	SI m	Field ft	Lab cm
min	1.0E-4	3.28E-4	1.0E-2
max	1.0E+20	3.28E+20	1.0E+22

The acceptable range of values for angles is:

	SI degrees	Field degrees	Lab degrees
min	0.0	0.0	0.0
max	360.0	360.0	360.0

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 keyword following).

EXPLANATION:

This keyword defines the dimensions of the grid blocks in the K direction. These dimensions are typically gross grid block thicknesses.

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:

- K direction grid increments for a Cartesian grid with $n_k=8$ are:

20,20,25,40,15,45,45,45

Use:

```
*DK *KVAR  
2*20 25 40 15 3*45
```

- 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
```

The acceptable range of values for block lengths in the K direction is:

	SI m	Field ft	Lab cm
min	1.0E-4	3.28E-4	1.0E-2
max	1.0E+20	3.28E+20	1.0E+22

Depth to the Centre of a Grid Block (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 j k

I, J and K direction indices of the reference block.

depth

Depth to the centre (or top if *TOP is used) of the reference block in the reservoir (m | ft | cm). 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, *PAYDEPTH or *DEPTH-TOP 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 *PAYDEPTH or *DEPTH-TOP keyword 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 of the grid block. The value may be positive or negative depending on the location of the reference surface, although positive values are most common.

At least one depth is required for any simulation. Note that corner point grids use special keywords to provide their depth information, or can obtain depth information from the actual corner point coordinates.

Depths are assigned to all blocks based on the depth provided by this keyword. This calculation is made based on the blocks' thicknesses (*DK keyword) and the dip angles provided by the *DIP keyword (see *DIP keyword description following).

Example:

```
*DEPTH 1 1 1 2000.0
```

The acceptable range of values for depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Depth to the Tops of Grid Blocks (Conditional)

*DTOP

PURPOSE:

*DTOP indicates input of a number of depths that provide the depth 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, *PAYDEPTH or *DEPTH-TOP must be specified for *GRID *CART, *GRID *VARI, or *GRID *RADIAL. Use of this keyword, or *PAYDEPTH, is recommended for *GRID *VARI. *PAYDEPTH or *DEPTH-TOP, 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 must all be the same.

No array qualifiers or array reading options are permitted for this particular array keyword. A fixed number of values ($n_i \times n_j$) 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 $n_i \times n_j$ 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 $K = n_k$ 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).

See FIGURE 5 in Appendix D.

Example:

A variable depth/variable thickness grid with $n_i=6$, $n_j=4$ and $n_k=2$ 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	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

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. (Net pay is assumed to be centered in the grid block.)

ARRAY:

*PAYDEPTH

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group. One of *DEPTH, *DTOP, *PAYDEPTH or *DEPTH-TOP must be specified for *GRID *CART, *GRID *VARI, or *GRID *RADIAL. Use of this keyword, or *DTOP, is recommended for *GRID *VARI.

*PAYDEPTH can be specified for corner point grids. The *PAYDEPTH 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 pay of each individual grid block. All array qualifiers and array reading options are allowed for specifying the $n_i \times n_j \times n_k$ 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 *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 *PAYDEPTH with corner point grids. Use of *PAYDEPTH with corner point grids works like a vertical position modifier for the cells.

Example:

A variable depth/variable thickness grid with $n_i=6$, $n_j=4$ and $n_k=2$ 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 m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

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 $n_i \times n_j \times n_k$ 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 $n_i=6$, $n_j=4$ and $n_k=2$ 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	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

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. Not used for *GRID *RADIAL.

DEFAULTS:

Conditional keyword. No defaults.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group. This keyword is optional with *GRID *CART and *GRID *RADIAL, and is not used with *GRID *VARI or *GRID CORNER. The *DIP keyword should not be specified when the *DTOP, *PAYDEPTH or *DEPTH-TOP keywords are used to input depths.

The *jdip* parameter is not required and should not be input for *GRID *RADIAL.

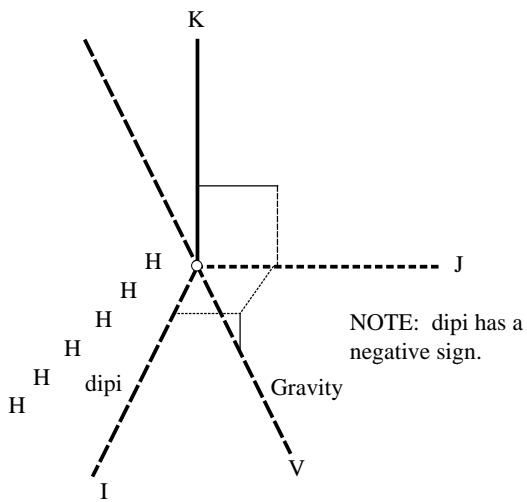
EXPLANATION:

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 that *DIP is not 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 *PAYDEPTH or *DEPTH-TOP keywords.)

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*.

See FIGURE 3 in Appendix D for further information regarding Cartesian grids, and FIGURE 4, also in Appendix D, for radial-angular cylindrical grids.

Example:

*DIP 1.0 0.0

The acceptable range of values for dip angles is:

	SI	Field	Lab
min	-90.0	-90.0	-90.0
max	90.0	90.0	90.0

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.

See section **Local Grid Refinement** for keyword *CORNERS.

The *ZCORN keyword causes the reading of all depths ("z"-coordinates) of the $8 \times n_i \times n_j \times n_k$ 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, ...,  $n_k$ : [
    Do the following for J = 1, ...,  $n_j$ : [
        Write NW-T and NE-T depths for block ( 1,J,K),
        ...
        Write NW-T and NE-T depths for block ( $n_i$ ,J,K).
        Write SW-T and SE-T depths for block ( 1,J,K),
        ...
        Write SW-T and SE-T depths for block ( $n_i$ ,J,K).
    ]
]
```

Do the following for $J = 1, \dots, n_j$: [

 Write NW-B and NE-B depths for block (1,J,K),

 ...

 Write NW-B and NE-B depths for block (n_i, J, K).

 Write SW-B and SE-B depths for block (1,J,K),

 ...

 Write SW-B and SE-B depths for block (n_i, 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 $n_i = 4$, $n_j = 2$, $n_k = 1$ *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	Lab cm
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

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.

See section **Local Grid Refinement** for keyword *CORNERS.

The *XCORN and *YCORN keywords each cause reading of all the $(n_i+1) \times (n_j+1) \times (n_k+1)$ "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, ..., ( $n_k + 1$ ): [  
  Do the following for J = 1, ..., ( $n_j + 1$ ): [  
    Do the following for I = 1, ..., ( $n_i + 1$ ): [  
      I, J, K are less than  $n_i$ ,  $n_j$ ,  $n_k$ , respectively:  
        write the "x"- (or "y"-) coordinate of the NW-T point;  
      J is less than  $n_j$ , K is less than  $n_k$ , and I =  $n_i$ :  
        write the "x"- (or "y"-) coordinate of the NE-T point;  
      I is less than  $n_i$ , K is less than  $n_k$ , and J =  $n_j$ :  
        write the "x"- (or "y"-) coordinate of the SW-T point;
```

I is less than n_i , J is less than n_j , and K = n_k :
 write the "x"- (or "y"-) coordinate of the NW-B point;
 I is less than n_i , and J = n_j , K = n_k :
 write the "x"- (or "y"-) coordinate of the SW-B point;
 J is less than n_j , and I = n_i , K = n_k :
 write the "x"- (or "y"-) coordinate of the NE-B point;
 K is less than n_k , and I = n_i , J = n_j :
 write the "x"- (or "y"-) coordinate of the SE-T point;
 I = n_i , J = n_j , K = n_k :
 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 $n_i = 4$, $n_j = 2$, $n_k = 1$ *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	Lab cm
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

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.

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 $(n_i+1) \times (n_j+1)$ 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 $2 \times 3 \times (n_i+1) \times (n_j+1)$ 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 $J = 1, \dots, (n_j + 1)$: [

Do the following for $I = 1, \dots, (n_i + 1)$: [

Firstly, ...

If I and J are less than n_i and n_j 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 $I = n_i$ and J is less than n_j , write the NE corner. If I is less than n_i and $J = n_j$, write the SW corner. If $I = n_i$ and $J = n_j$, write the SE corner.

Secondly, ...

If I and J are less than n_i and n_j 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 = n_i and J is less than n_j , write the NE corner. If I is less than n_i and J = n_j , write the SW corner. If I = n_i and J = n_j , 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 $n_i = 4$, $n_j = 2$, $n_k = 1$ *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 n_k)

```
*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	Lab cm
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

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 $3 \times (8 \times n_i \times n_j \times n_k)$ values, with the first group of $8 \times n_i \times n_j \times n_k$ values giving all the "x"-coordinates of all corner points, the second group giving all the "y"-coordinates, and the third group 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 $8 \times n_i \times n_j \times n_k$ 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, ..., n_k : [
 Do the following for J = 1, ..., n_j : [
 Write NW-T and NE-T values for block (1,J,K)
 ...
 Write NW-T and NE-T values for block (n_i ,J,K)
 Write SW-T and SE-T values for block (1,J,K)
 ...
 Write SW-T and SE-T values for block (n_i ,J,K)
]]
 Do the following for J = 1, ..., n_j : [
 Write NW-B and NE-B values for block (1,J,K)
 ...
 Write NW-B and NE-B values for block (n_i ,J,K)
 Write SW-B and SE-B values for block (1,J,K)
 ...
 Write SW-B and SE-B values for block (n_i ,J,K)
]]]]

This completes the algorithm.

This technique for corner point input will exhibit duplication in the first two groups of $8 \times n_i \times n_j \times n_k$ values, due to the fact that corner points must lie on vertical lines.

Examples:

Provide *CORNERS data for a $n_i = 4$, $n_j = 2$, $n_k = 1$ 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	Lab cm
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

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.

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 | cm), one value for each of *nnodes* nodes. The order of values is given by the **node list**.

***NODEY**

Array of Y coordinates (m | ft | cm), one value for each of *nnodes* nodes. The order of values is given by the **node list**.

***NODEZ**

Array of Z coordinates (m | ft | cm), 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 | cm), 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 $n_i \cdot n_j \cdot n_k$ blocks specified by *GRID *CORNER.

When *RG is present, this keyword refers to a locally refined grid which has $n_{ir} \cdot n_{jr} \cdot n_{kr}$ 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 $(n_i+1) \cdot (n_j+1) \cdot (n_k+1)$ 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 $(n_i+1) \cdot (n_j+1) \cdot (n_k+1)$ 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 $n_i = n_j = n_k = 2$ 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 $n_i = 3$, $n_j = 2$ and $n_k = 1$ 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 $(n_i+1) \cdot (n_j+1) \cdot (n_k+1) = 24$ unique nodes and $n_i \cdot n_j \cdot n_k = 6$ 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
*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
*NODES 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	Field	Lab
	m	ft	cm
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

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 switch ‘-crncon_node’ is available as an alternate to keyword
*CRNCON_NODE.

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 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.

Example

The following data defines a block group named 'Sand-1' consisting of a few cells in a $5 \times 5 \times 1$ grid and part of the locally refined $3 \times 3 \times 2$ 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
```

Local Refined Grid (Optional)

*REFINE, *RANGE

PURPOSE:

*REFINE defines specifications for local grid refinement.

(Note that refinement means that certain blocks in a range on the fundamental (main) grid, this range being described by the *RANGE keyword following, will each be replaced by a refined grid. Each refined grid will be made up of several small blocks that will together fill the space occupied previously by a parent fundamental block.)

The *REFINE keyword can be used with *GRID *CART, *GRID *VARI or *GRID *CORNER but not with *GRID *RADIAL.

*RANGE specifies the input of location(s) where the refinement(s) defined by the previous *REFINE keyword are to act. (*RANGE is not required if *INTO was used with the *REFINE keyword.)

FORMAT:

```
*REFINE *HYBRID nir njr nkr (*IDIR | *JDIR | *KDIR)
      *RW rw (*ALPHAI alphai) (*ALPHA alpha)
      -or-
*REFINE nir njr nkr
      -or-
*REFINE block_address *INTO nir njr nkr
      -or-
*REFINE block_address *INTO nir njr nkr *HYBRID (*IDIR | *JDIR | *KDIR)
      *RW rw (*ALPHAI alphai) (*ALPHA alpha)
      -or-
*REFINE *INNERWIDTH Winner
      block_address *INTO nir njr nkr
      (*WF2 Wf2) (*K2INT k2int)
*RANGE block_address
```

DEFINITIONS:

block_address

The address of the grid block(s) to which this refinement applies. Three forms are allowed:

1. Multi-level single-block UBA: $i_1 j_1 k_1 / i_2 j_2 k_2 \{ \dots \{ / i_n j_n k_n \} \}$;
2. Single-level range $i_1(:i_2) j_1(:j_2) k_1(:k_2)$; 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.

***HYBRID**

Indicates the use of a hybrid refined grid. This is a refined grid made up of stacked rings with optional angular subdivisions, much like a radial- angular cylindrical grid. This refinement is associated with a wellbore of radius *rw* (as specified by the *RW subkeyword following) assumed to be running through the centres of the innermost rings. The wellbore's volume is not included in any of the simulator's grid blocks.

*HYBRID rings will be circular if the parent block's permeabilities perpendicular to the well are the same (isotropic case). Otherwise, the rings will have ellipses for boundaries with aspect ratios related to the ratio of parent block permeabilities. If *HYBRID does not appear, the refined grid will be made up of rectangular blocks.

n_{ir}

If *HYBRID is specified, then *n_{ir}* is the number of rings to be used. The rings will have curved sides (circles or ellipses) with ring number 1 always having a circular inner boundary of radius *rw*. The last ring, ring number *n_{ir}*, will be curved on its inner boundary, but its outer boundary will be made up of four flat sides that match up with neighbouring fundamental grid blocks or other refined grids. These flat sides correspond to the four sides of the parent grid block that lie in the directions perpendicular to the well.

If the refinement is rectangular, than *n_{ir}* is the number of grid blocks to be inserted in the I direction in the refinement.

n_{jr}

If *HYBRID is specified, *n_{jr}* must be 1 or 4. If *n_{jr}* is equal to 4, each ring will be broken up into four pieces, each associated with 90 degrees of angular extent as measured in the appropriate angular coordinate. Note that ring number 1 is never quartered however, not even when *n_{jr}* is 4. If *n_{jr}* = 1, the rings are not subdivided.

If the refinement is rectangular, than *n_{jr}* is the number of grid blocks to be inserted in the J direction in the refinement.

n_{kr}

If *HYBRID is specified, then *n_{kr}* is the number of layers used in the stacks of rings; that is, *n_{kr}* is the number of layers used in the direction along the wellbore. If the refinement is rectangular, then *n_{kr}* is the number of grid blocks to be inserted in the K direction in the refinement.

***IDIR, *JDIR, *KDIR**

The wellbore is aligned with the I, J or K direction on the fundamental grid. Thus, the radial/elliptical rings used with the *HYBRID refined grid will be set up perpendicular to the I, J or K direction.

RW *rw

Define well radius (m | ft | cm), 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, 100 cm). 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_i/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 $R_i = rw \cdot alphai$. 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 $n_{ir}-1$, 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 | cm) 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 2/3 m 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 | cm). 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:

Optional keyword. The default is no refined grids.

If *REFINE *HYBRID appears with no values, the default for n_{ir} is 2 (two rings), the default for n_{jr} is 1 (no angular subdivisions), and the default for nkr is 1 (no layering).

If *REFINE appears with no values and no *HYBRID subkeyword, the default dimensions are $n_{ir} = n_{jr} = n_{kr} = 3$, meaning 3 rectangular subdivisions are used in each direction.

The default for rw is 3 inches (.25 ft or .0762 m) if *RW does not appear, or if *RW 0.0 appears.

The wellbore direction will default to *KDIR (vertical) if no *IDIR, *JDIR, or *KDIR appears for *REFINE *HYBRID.

If *ALPHAI is absent, $\alpha_{hai} = R_{parent}/[rw \cdot (nir+1)]$, 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 $R_1 = rw \cdot \alpha_{hai} = R_{parent}/(nir+1)$.

If *ALPHA is absent, $\alpha_{ha} = [R_{parent}/R_1]^{1/(nir-1)}$ 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 $\alpha_{ha} \cdot R_{nir-1} = R_{parent}$. 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:

The *REFINE keyword may be located in the Reservoir Description keyword group. It may also occur as part of recurrent data. Thus refined blocks may appear later in the simulation. Any well perf's which use the refined blocks or any of coarser blocks which contain them, must be defined with *REFINE before they are perforated with *PERF.

*RANGE keyword should follow immediately after the *REFINE keyword, or other *RANGE keywords. This offers an older, alternate form for the *INTO syntax of the *REFINE keyword.

Note that neighbouring *HYBRID grids must share the same well direction, and if the well through one *HYBRID grid is to be continued into another *HYBRID grid, both grids must have the same n_{ir} and n_{jr} . For multilevel refinement, only the finest grid may be hybrid.

The *REFINE keyword can be used with *GRID *CART, *GRID *VARI or *GRID *CORNER but not with *GRID *RADIAL.

Use *PERFV with *REFINE *HYBRID.

EXPLANATION:

Regular Refinement

A *REFINE keyword, with accompanying subkeywords, and *RANGE keyword(s), indicate that a local refined grid option is being used. *REFINE and its parameters determine the

amount of refinement, and *RANGE indicates where the refinement is to take place. (The *RANGE keyword is described later.) Refinements are Cartesian (rectangular) unless *HYBRID is specified.

*REFINE $i_1:i_2 j_1:j_2 k_1:k_2$ *INTO ($n_{ir} n_{jr} n_{kr}$) or *REFINE $i_1 j_1 k_1 / i_2 j_2 k_2 \{ \dots \{ / i_n j_n k_n \} \}$ *INTO ($n_{ir} n_{jr} n_{kr}$) specifies the starting and ending indices for a group of fundamental grid blocks or a single multilevel refinement block which are to be refined.

In addition, the *REFINE ... *INTO ... option allows the user to specify the refined grid size on the same keyword the range is defined. *REFINE ... *INTO is therefore a combination of both the standard *REFINE and *RANGE keywords.

Each grid is assigned a number by the simulator. Grids are numbered starting with the fundamental grid which is grid number 1, followed by grids created by *REFINE ... *INTO keywords. Each *REFINE ... *INTO keyword creates grids by running through the parent blocks (I,J,K) with I running fastest from i_1 to i_2 , J next fastest, and K running slowest from k_1 to k_2 . Thus, each *RANGE keyword creates $(i_2-i_1+1) \times (j_2-j_1+1) \times (k_2-k_1+1)$ new grids which are numbered in succession.

Example:

The following describes 4 grids in total:

```
*GRID *VARI 5 5 1
...
...
*REFINE 1:2 2 1 INTO 2 2 3
*REFINE 2 2 1 / 2 2 2 INTO 3 3 1
...
...
```

Grid 1 is the fundamental grid, another grid has parent (1,2,1), another has parent (2,2,1) and the last grid has parent refined grid (2,2,2) in parent (2,2,1).

Example using BLOCKGROUP:

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
```

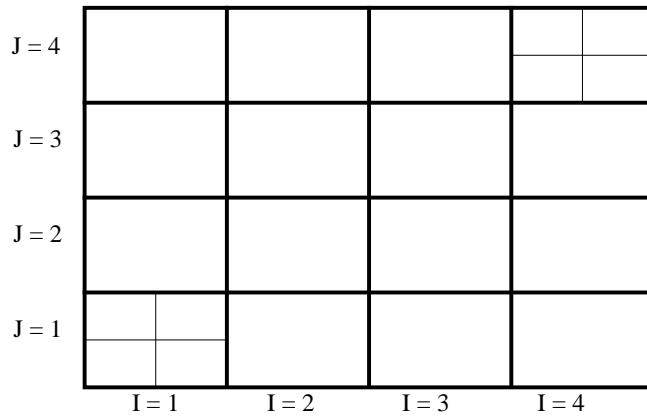
Example:

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, and each of these blocks is to contain a $2 \times 2 \times 2$ grid. The refined grid blocks (1,1,1) and (1,1,2) in the fundamental grid blocks (1,1,1) and (1,1,2) are to be further refined into $2 \times 2 \times 1$ grids.

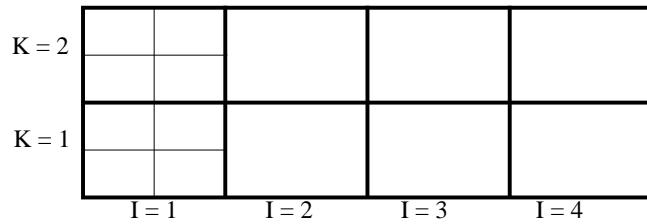
The data may appear as follows:

```
*GRID *CART 4 4 2
:
*REFINE 2 2 2
*RANGE 1 1 1:2
*RANGE 4 4 1:2
*REFINE 2 2 1
*RANGE 1 1 1 / 1 1 1
*RANGE 1 1 1 / 1 1 2
*RANGE 1 1 2 / 1 1 1
*RANGE 1 1 2 / 1 1 2
```

An I-J view of the grid would be:



A cross-section for J=1 would be:



The acceptable range of values for well radius is:

	SI m	Field ft	Lab cm
min	0.0	0.0	0.0
max	1.0	3.280	3.280

The acceptable range of values for the *ALPHA and *ALPHAI ratios are:

	SI m	Field ft	Lab cm
min	1.0	1.0	1.0
max	1.0E+20	3.28E+20	1.0E+22

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

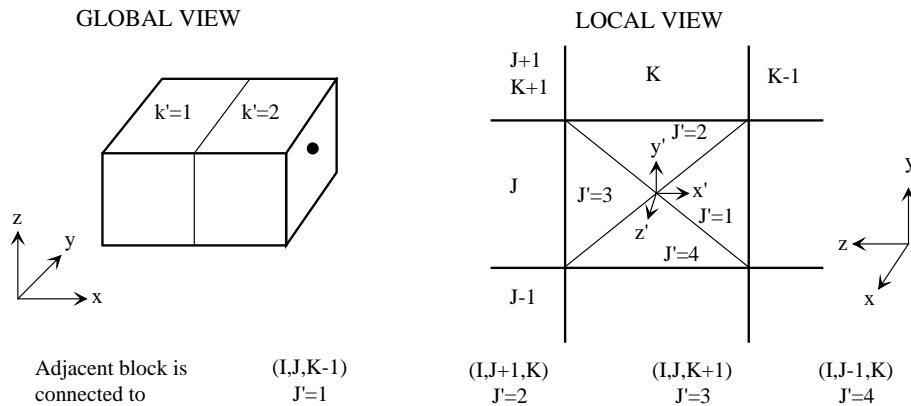
Normally, the user will need to know only which of *IDIR, *JDIR or *KDIR was specified in order to interpret the position of the individual hybrid grid blocks relative to the surrounding fundamental blocks. However, the precise meaning of the hybrid grid's local J and K indices is needed in order to 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.

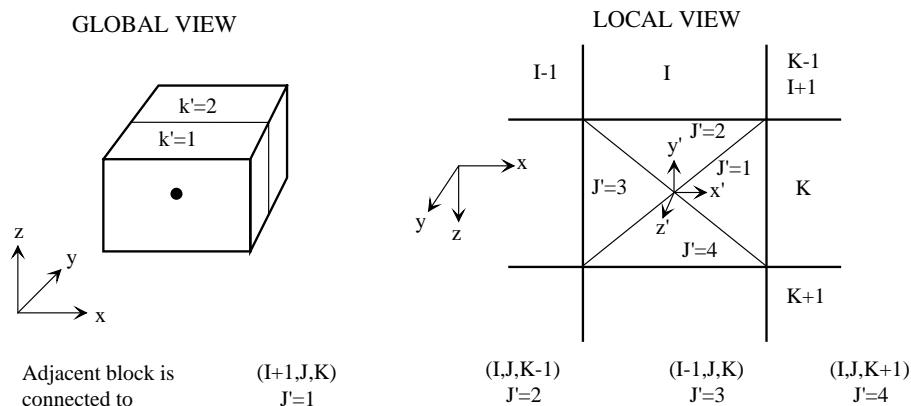
In each of the following orientation cases, both a "GLOBAL VIEW" and a "LOCAL VIEW" are shown. In the "LOCAL VIEW" the point-of-view is on the hybrid z' axis looking in the

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.

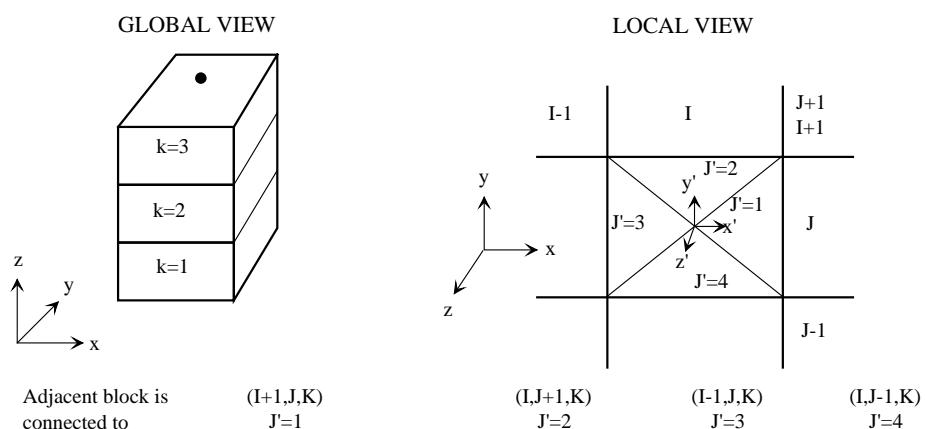
Well in I Direction (*IDIR): $x = xo + z'$, $y = yo + y'$, $z = zo - x'$



Well in J Direction (*JDIR): $x = xo + x'$, $y = yo + z'$, $z = zo - y'$



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

and output

- block sizes
- permeabilities
- transmissibility multipliers
- transmissibilities

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.

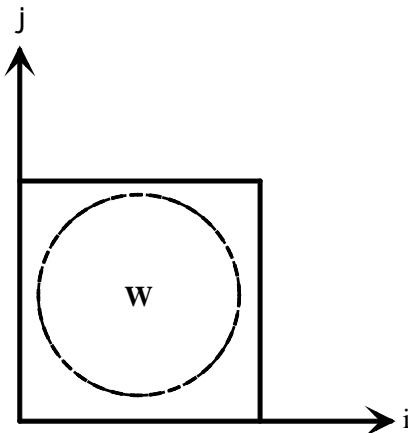
Block Sizes

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

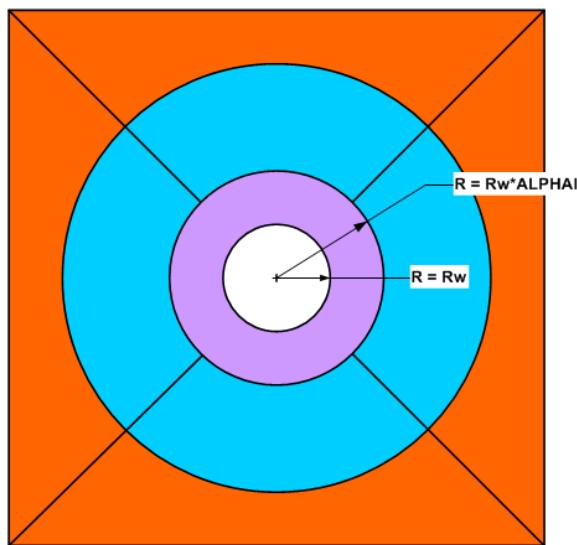
```
*REFINE 1 1 1 *INTO 3 4 1 *HYBRID *IDIR
```

Hybrid Grid Definition

Example shows a $3 \times 4 \times N$ hybrid grid with $Kh/Kv = 1$ (3 solid rings plus well bore, divided into four angular segments).

Note that the inner ring does not have angular divisions, and contains the well bore of radius R_w .

The well bore itself is currently not visualized in Results 3D or Builder unless it is a discretized well bore. In this case the annulus of the well bore is represented by the first, unsegmented, ring, and the tubing (if it exists) by an inner circle.



Example: Hybrid Grid Cell Geometry

Assume a data set has the following data:

```
*GRID *CART 5 5 6
*DI *CON 131.24
*DJK *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 $n_{ir} = 4$ 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 $i=1$ to n_{ir} , 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,

$$R_1 = R_{parent}/(n_{ir}+1) = 14.809.$$

Since *ALPHA is absent,

$$\alpha = [R_{parent} / R_1]^{1/(n_{ir}-1)} = 1.71, \text{ so}$$

$$R_2 = \alpha \cdot R_1 = 25.323,$$

$$R_3 = \alpha \cdot R_2 = 43.301 \text{ and}$$

$$\text{hypothetical radius } \alpha \cdot R_3 = 74.044 = R_{parent}.$$

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

$$V_{b1} = \frac{1}{2} \cdot (2\pi) \cdot (R_1^2 - R_0^2) \cdot L_z = 13555, V_{p1} = 4744,$$

$$V_{b2} = \frac{1}{2} \cdot (\pi/2) \cdot (R_2^2 - R_1^2) \cdot L_z = 6522.8, V_{p2} = 2283,$$

$$V_{b3} = \frac{1}{2} \cdot (\pi/2) \cdot (R_3^2 - R_2^2) \cdot L_z = 19073, V_{p3} = 6676 \text{ and}$$

$$V_{b4} = [\frac{1}{4} \cdot d_i \cdot d_j - \frac{1}{2} \cdot (\pi/2) \cdot R_3^2] \cdot L_z = 55769, V_{p4} = 19519.$$

The radial block sizes L_R are

$$L_{R1} = R_1 - R_0 = 14.509,$$

$$L_{R2} = R_2 - R_1 = 10.514,$$

$$L_{R3} = R_3 - R_2 = 17.979 \text{ and}$$

$$L_{R4} = \frac{1}{2}(131.24) - R_3 = 22.319.$$

Use function $f(\alpha) = \exp[\ln(\alpha) \cdot \alpha^2 / (\alpha^2 - 1) - \frac{1}{2}]$ to obtain angular block lengths L_θ

$$L_{\theta1} = (2\pi) \cdot f(R_1/R_0) \cdot R_0 = 56.526,$$

$$L_{\theta2} = (\pi/2) \cdot f(R_2/R_1) \cdot R_1 = 31.884,$$

$L_{03} = (\pi/2) \cdot f(R_3/R_2) \cdot R_2 = 54.523$ and
 $L_{04} = d_i$ 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
```

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 nir njr nkr  
  *BWHLEN Lhalf  
  ( *IDIR | *JDIR )  
  *INNERWIDTH Winner  
  ( *WF2 Wf2 )  
  ( *K2INT k2int )  
  ( *LAYERSUP nlayup )  
  ( *LAYERSDOWN nlaydown )  
  { grid_array (*MATRIX | *FRACTURE) (*FZ | *NFZ) ... }  
*END_TEMPLATE  
*PLNRFRAC 'Template_Name' fracture_origin *BG_NAME 'bg_name'  
*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'  
Apply template 'Template_Name' to fracture_origin with the blockgroup name 'bg_name'. Blockgroup name 'bg_name' must not have been defined previously.
```

For each invocation of *PLNRFRAC 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 grid-array data to blockgroup '*bg_name*'.

```
*PLNR_REFINE *INTO nir njr nkr
```

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 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 skipped with *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 | cm) 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 propagated from *fracture_origin*. The direction keyword is required.

*INNERWIDTH W_{inner}

Specify width W_{inner} (m | ft | cm) 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 | cm).

*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.

{ *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,	Reservoir Description	Yes
*DJFRAC, *DKFRAC		
*RTYPE, *NDARCYCOR, *SWCON,	Rock Fluid	Yes
*SWCRIT, *SGCRIT, *SORW, *SORMAX,		
*SOIRW, *SORG, *SGCON, *SLCON,		
*SOIRG, *KRWIRO, *KRGCL, *KRGRL,		
*KROCW, *KROCRW, *KROGCG,		
*KROGCRG, *PCWMAX, *JFWMAX,		
*PCGMAX, *JFGMAX		
*ADGMAXV, *ADGCSTV, *ROCKDEN,	Rock Fluid	No
*ADGPCRIT		
*PRES, *SW, *SO, *PB, *PDEW, *API,	Initial Conditions	No
*POLYCONC, *SEAWATFRC, *PBS,		
*SWINIT, *SWNEQ, *SONEQ, *ITYPE		
*PTYPE	Component Properties	No

***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 *MDPLNRBK is absent then *fund_blocks* = 10.

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.

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  
  *BWLLEN 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'  
. . .  
*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 a dual porosity model related to the Warren and Root model in some or all of the simulator's grid blocks.

FORMAT:

*DUALPOR

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.

EXPLANATION:

This keyword indicates that a Warren and Root style dual porosity option be used in the simulator. 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. 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 keyword while fracture properties are described using the *FRACTURE keyword.

In the Warren and Root model, inter-block flows are calculated in much the same manner as they would be in the standard (no *DUALPOR keyword) model. These flows are governed by the fracture properties. However, an additional set of matrix-fracture flows are calculated when *DUALPOR is specified. These flows are governed by the matrix properties. The calculation of this matrix-fracture fluid transfer is described when the *SHAPE keyword is defined.

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 identifier and the other a *FRACTURE identifier. 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 related to the Warren and Root model in some, or all, of the simulator's grid blocks. Moreover, inter-block fracture to fracture flows are augmented by inter-block matrix to matrix flows. The matrix to fracture flows within blocks remain.

FORMAT:

*DUALPERM

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.

EXPLANATION:

The description given above for the *DUALPOR keyword should be studied first, as this option is closely related.

The *DUALPERM option uses the same calculations as the *DUALPOR option, except that inter-block matrix to matrix flows are also calculated in addition to the expected inter-block fracture to fracture flows and the matrix to fracture fluid transfer within blocks.

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 identifier and the other a *FRACTURE identifier. Further details are explained in the descriptions for the individual properties.

Dual permeability is often important 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 this is so, use zero transmissibility modifiers in the I and J directions (see the *TRANSI and *TRANSJ keywords following).

Dual Porosity Subdomain Method (Optional)

*SUBDOMAIN

PURPOSE:

*SUBDOMAIN indicates the use of a dual porosity model using the subdomain method.

FORMAT:

*SUBDOMAIN *idiv* (*3DMF)

DEFINITIONS:

idiv

Number of subdivisions for each matrix blocks, typically chosen from the range of 2 to 5.

*3DMF

If Found, *3DMF enables matrix-fracture transmissibility through the vertical planes, by default areal matrix-fracture flow is not considered when using the SUBDOMAIN option.

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.

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.

Normally when calculating the matrix-fracture-transmissibility term, the vertical flow into the subdomain is ignored. In certain circumstances this may significantly reduce the overall matrix fracture transmissibility, the use of the option *3DMF introduces the M-F transmissibility terms from the vertical direction into the top and bottom subdomain cells. The use of the *3DMF option may increase M-F terms significantly which can increase execution time.

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:

By default the “*SUBDOMAIN *idiv*” option generates subdomains of thickness “Spatial block thickness”/*idiv*. Similarly the “*MINC *idiv*” option generates nested rings of equal volume.

FRACVOL allow the user to alter the size of individual subdomain blocks (*SUBDOMAIN) or nested rings (*MINC) relative to each other. This is used to improve subdomain capillary holdup calculations and is crucial in the proper modeling of reinfiltration.

FORMAT:

*FRACVOL *frac₁* ... *frac_{idiv}*

DEFINITIONS:

frac₁

Fractional volume of first subdomain block / outermost ring block

frac_{idiv}

Fractional volume of last subdomain block / last ring block

DEFAULTS:

Optional keyword. No default. If omitted subdomain volumes / 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 volume/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.

In addition *TRANSD when used with the *MINC option enables vertical matrix to matrix flow in *MINC models (i.e. MINC-DK)

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 and *MINC dual porosity option is in effect.

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 FRACTVOL and SD_REINF.

In the standard *MINC dual continua model, the matrix is modeled as a number of interacting blocks within each dual continua grid block, matrix blocks can flow to matrix blocks only within a dual porosity grid block. Inter dual continua gridblock flow only occurs within the fracture.

The use of non zero *TRANSD with the *MINC model (when DKFRAC = 0.0) allows the MINC model to set up vertical matrix inter dual continua grid block flow, similar to the dual permeability model but only in the vertical direction. We call this MINC-DK. It is important to set *DKFRAC= 0.0 so that matrix-matrix and matrix-fracture transmissibility will not be overestimated when using *MINC-DK. The *SD_REINF keyword should not be used when employing the MINC-DK option as it is meant solely for SDDK gridding.

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	Lab
min	0.0	0.0	0.0
max	1000.0	1000.0	1000.0

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.

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 which type of shape factor is to be used in calculating matrix to fracture flows for various 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.

Only relevant when one of *DUALPOR, *DUALPERM, *SUBDOMAIN, or *MINC 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 (assuming the regions are of similar size). 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 portions 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:

$$\text{W \& R Transmissibility} = 20/3 \cdot k \cdot (1/L_x + 1/L_y + 1/L_z)^2 \cdot M\text{-Vol}$$

whereas the Gilman and Kazemi formulation has:

$$\text{G \& K Transmissibility} = 4 \cdot k \cdot (L_x^{-2} + L_y^{-2} + L_z^{-2}) \cdot M\text{-Vol}$$

where the above assumes fracture planes exist in all three directions with fracture spacings L_x , L_y , L_z (which are set using the *DIFRAC, *DJFRAC and *DKFRAC keywords following), the permeability is k , and the total matrix volume is $M\text{-Vol}$. 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 are modelled with the usual matrix to fracture flows in the individual blocks.

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

DEFAULTS:

Conditional keyword. If the keyword *TRANSFER is not found the default is *TRANSFER 0. If the keyword *TRANSFER is found without an explicitly defined NTRAN, the default is *TRANSFER 1.

CONDITIONS:

This keyword must be located in the Reservoir Description keyword group. This keyword is only required for naturally fractured reservoirs as specified by either the *DUALPOR or *DUALPERM keyword.

EXPLANATION:

Normally, when NTRAN=0, the matrix fracture transfer is calculated following either Warren and Roots method or Gilman and Kazemi's approach (see the *SHAPE keyword for more details).

For example, using the *SHAPE *GK option, the matrix-fracture transfer term τ_{jmf} for phase j would be calculated as

$$\tau_{jmf} = \sigma V \frac{K_{rj}\rho_j}{\mu_j} (P_{jm} - P_{jf}) \quad j = o, w, g$$

where

ρ_j = reservoir condition density of phase j

$P_g = P_o + P_{cog}$

$P_w = P_o - P_{cow}$

$$\sigma = 4 \left(\frac{K_x}{L_x^2} + \frac{K_y}{L_y^2} + \frac{K_z}{L_z^2} \right)$$

with V being the grid block volume and L_x , L_y and L_z the fracture spacing (*DXFRAC, *DYFRAC, *DZFRAC) in the x , y and z directions respectively.

The matrix and fracture blocks are assumed to be at the same depth, thus no gravity term is included above. This equation is used when NTRAN=0. Generally, the matrix capillary pressure is much larger than the fracture capillary pressure and the above expression for the transfer term τ_{jmf} cannot correctly model the gravity drainage process.

The gravity effects needed to model the gravity drainage process can be approximately included by using the NTRAN=1 option. This assumes complete gravity segregation of the oil, water and gas phases as follows:

$$\begin{aligned}\tau_{omf} &= \sigma V \frac{K_{ro} \rho_o}{\mu_o} (P_{om} - P_{of}) \\ \tau_{gmf} &= \sigma V \frac{K_{rg} \rho_g}{\mu_g} \left[P_{gm} - P_{gf} + \left(\frac{S_g}{1 - S_{org} - S_{wc}} - \frac{1}{2} \right)_m \right. \\ &\quad \left. \Delta \gamma_{ogm} h - \left(\frac{S_g}{1 - S_{org} - S_{wc}} - \frac{1}{2} \right)_f \Delta \gamma_{ogf} h \right] \\ \tau_{wmf} &= \sigma V \frac{K_{rw} \rho_w}{\mu_w} \left[P_{wm} - P_{wf} - \left(\frac{1}{2} - \frac{S_w - S_{wc}}{1 - S_{orw} - S_{wc}} \right)_m \Delta \gamma_{wom} h \right. \\ &\quad \left. + \left(\frac{1}{2} - \frac{S_w - S_{wc}}{1 - S_{orw} - S_{wc}} \right)_f \Delta \gamma_{wof} h \right]\end{aligned}$$

where

- $\Delta \gamma_{og}$ = (reservoir condition oil density – reservoir condition gas density) *g
- $\Delta \gamma_{wo}$ = (reservoir condition water density – reservoir condition oil density) *g
- h = height of matrix element in the direction of gravity (h is normally *DZFRAC unless the block thickness is less than *DZFRAC in which case h – block thickness)

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 difference in the height of the fluid columns.

Allowing Addition Matrix-Fracture Connections (Optional)

***DPCONNECT**

PURPOSE:

The DPCONNECT keyword allows the grid module to produce extra connection between blocks in a dual porosity/permeability system when either the matrix or the fracture is missing from one of the two blocks being connected.

FORMAT:

***DPCONNECT 1**

DEFAULTS:

Optional keyword. No default. If omitted normal matrix-fracture connects are made.

CONDITIONS:

This keyword must be located in the Reservoir Description keyword group. *DUALPOR or *DUAL PERM must be specified.

EXPLANATION:

When enabled, a dual porosity or dual permeability block with a missing fracture will connect the matrix to both the matrix and fracture of surrounding blocks. In addition a dual porosity or dual permeability block with a missing matrix will connect the fracture to both the matrix and fracture of surrounding blocks.

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, *DUALPERM, *SUBDOMAIN, and *MINC.

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 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 spacings in the "y"-direction. Spacings are generally inherited from parent blocks for *HYBRID grids and hence, are automatically available.

The acceptable range of values for fracture spacing is:

	SI m	Field ft	Lab cm
min	0.0	0.0	0.0
max	1.0E+4	32,808.0	1.0E+6

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); no blocks are null.

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 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. These properties will just be ignored. 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/*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 $10 \times 10 \times 3$ 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  
1:3 8:8 1:3 0  
1:2 7:7 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.

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 keyword (see following).

Dimensionless fractions are expected (values between 0.0 and 1.0), NOT percentages. Zero values have special significance in setting blocks null.

When a dual porosity model is not being used, a block becomes null (non-participating) when it is assigned a zero porosity.

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 porosities is:

	SI dimensionless	Field dimensionless	Lab dimensionless
min	0.0	0.0	0.0
max	1.0	1.0	1.0

Rock Compressibility (Required)

*PRPOR, *CPOR

PURPOSE:

*CPOR signals the input of a rock compressibility value that will be used throughout the entire model.

*PRPOR signals the input of a reference pressure for the rock compressibility. This pressure is the fluid (pore) pressure at which the values input using *POR are to hold.

*CPOR and *PRPOR cause modifications to continuously occur in the input *POR porosity values as fluid pressure in the reservoir rises and falls.

FORMAT:

```
*CPOR (*MATRIX)      cpor
          (*FRACTURE)
*PRPOR (*MATRIX)      ref_pressure
          (*FRACTURE)
```

DEFINITIONS:

cpor

Pressure dependence of formation porosity; that is, rock compressibility
(1/kPa | 1/psi | 1/kPa | 1/(kg/cm²)).

ref_pressure

Reference pressure (kPa | psi | kPa | kg/cm²) for calculating the effect of rock compressibility.

DEFAULTS:

*CPOR will default to 0.0 (all units).

*PRPOR will default to 1 atmosphere (101.3 kPa | 14.7 psi | 101.3 kPa | 1 kg/cm²).

CONDITIONS:

These keywords must be in the Reservoir Description keyword group.

EXPLANATION:

The functional form used in the simulator for the calculation of a block's porosity is:

$$por(p) = por_input * [1 + cpot * (p - prpot)]$$

where *por_input* is the porosity input for the block using the *POR keyword, *cpot* and *prpot* are as input using the *CPOR and *PRPOR keywords, and *p* is the pore pressure (in the oil phase) for the block.

Note that the porosities entered using the *POR keyword should be associated with, or measured at, the reference pressure entered using *PRPOR.

When using dual porosity models, both matrix and fracture rock compressibilities and reference pressures should be input. The *MATRIX and *FRACTURE subkeywords are used to indicate the matrix and fracture dependence.

Example:

Porosity values for a dual porosity model were estimated to be near the values .11 and .02, for matrix and fracture respectively, at atmospheric conditions. The rock compressibilities were estimated to be 3.0E-6 for both porosities. The following data should be used:

```
*CPOR *MATRIX 3.0E-6
*PRPOR *MATRIX 14.7
*CPOR *FRACTURE 3.0E-6
*PRPOR *FRACTURE 14.7
*POR *MATRIX *CON .11
*POR *FRACTURE *CON .04
```

The suggested range of values for rock compressibility is:

	SI 1/kPa	Field 1/psi	Lab 1/kPa	Mod. SI 1/kg/cm**2
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89x10-3	1.0E-3	1.0E-5

and for reference pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm**2
min	0.0	0.0	0.0	0.0
max	1.0E+5	14505.0	1.0E+5	1.0E+2

Values outside of these ranges are accepted but a warning will be issued by the simulator.

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:

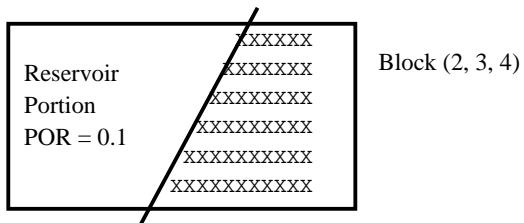
Pore volumes are calculated in the simulator based on the reservoir rock bulk volume, which is calculated from the external grid block dimensions, and the porosity, which depends on pore-pressure and compressibility data. The pore volumes accumulate fluids in the simulator and are basic to all its calculations.

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 gas 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.



```
*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	Lab cm
min	0.0	0.0	0.0
max	1.0E+4	32,808.0	1.0E+6

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-)*

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.

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 V_1, A_1, V_2, A_2 , 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.
```

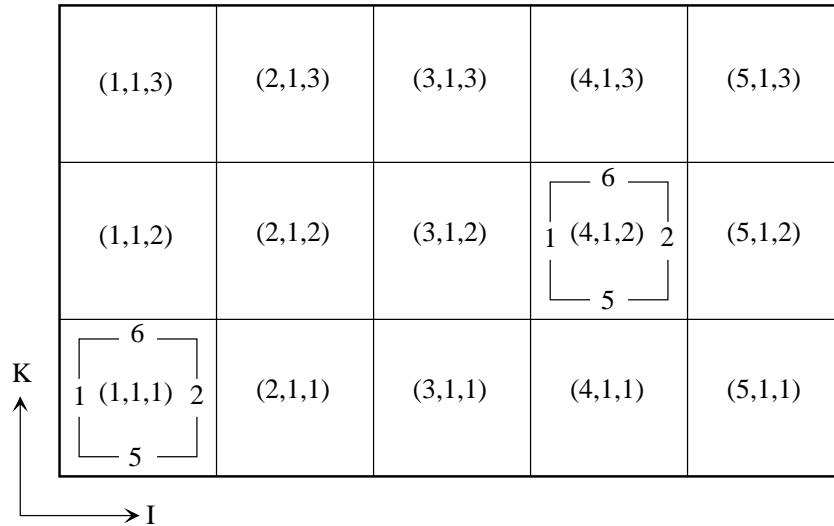


Figure 1: Referencing Grid Block Faces

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

$$\begin{aligned}
 v &= [\text{desired volume}] / [\text{volume from block sizes}] \\
 &= [DX * DY * DZ] / [DX' * DY * DZ] \\
 &= 0.5 \\
 ai &= [\text{desired area}] / [\text{area from block sizes}] \\
 &= [DY * DZ] / [DY * DZ] \\
 &= 1
 \end{aligned}$$

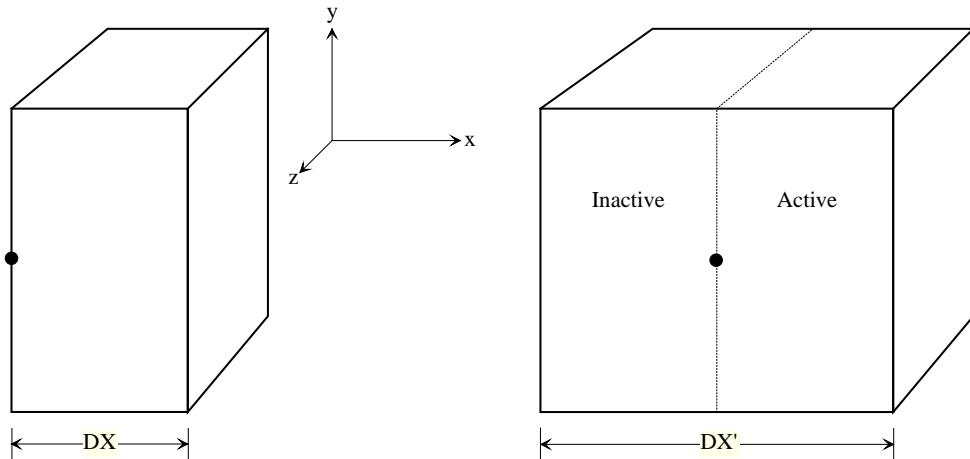
$$\begin{aligned}
 aj &= [\text{desired area}] / [\text{area from block sizes}] \\
 &= [DX * DZ] / [DX' * DZ] \\
 &= 0.5 \\
 ak &= [\text{desired area}] / [\text{area from block sizes}] \\
 &= [DX * DY] / [DX' * DY] \\
 &= 0.5
 \end{aligned}$$

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

```



```
*VAMOD key 0.5 1 0.5 0.5
```

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

$$\begin{aligned}
 v &= [\text{desired volume}] / [\text{volume from block sizes}] \\
 &= [DX * DY * DZ] / [DX' * DY' * DZ] \\
 &= 0.25
 \end{aligned}$$

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

```

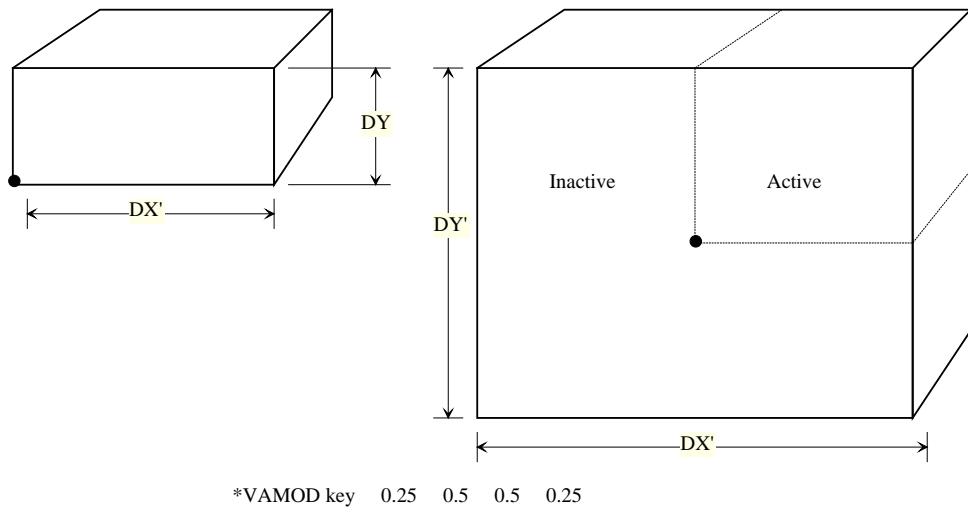


Figure 3: Grid Node in a Block Corner

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.

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 an array of J direction permeabilities.
- *PERMK indicates input of an array of K direction permeabilities.

ARRAY:

*PERMI
*PERMJ
*PERMK

DEFAULTS:

Required keywords. No defaults.

CONDITIONS:

These keywords must be in the Reservoir Description keyword group. They may also appear in recurrent (well) data.

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, provided that a *PERMI array is entered first. *EQUALSI may not be used if a Dual Continua model is used with grid refinement.

Example:

Permeability input for a *DUALPOR model may appear as follows:

```
** 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 acceptable range of values for permeability is:

	SI mD	Field mD	Lab mD
min	0.0	0.0	0.0
max	1.0E+9	1.0E+9	1.0E+9

Netpay (Optional)

*NETPAY

PURPOSE:

*NETPAY indicates input of an array of net pays which are to be 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 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 | cm) 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 used to modify the porosity array and permeability arrays 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) $perm_i$ is replaced by $perm_i * ntg$
- c) $perm_j$ is replaced by $perm_j * ntg$

where por denotes the grid block's porosity as set using the *POR keyword, $perm_i$ and $perm_j$ are the block's permeabilities as set using the *PERMI and *PERMJ keywords. ntg is the net-to-gross ratio as derived from the net pay value assigned by the *NETPAY keyword divided by the appropriate thickness obtained from the *DK array or corner point input.

Note that the permeability in the K direction is not altered. Transmissibility multipliers (see the *TRANSK keyword following) are available for vertical flow adjustments due to full or partial barriers caused by shales or other geological features.

The acceptable range of values for any derived net-to-gross ratio is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0E+4	1.0E+4	1.0E+4

Netgross (Optional)

***NETGROSS**

PURPOSE:

*NETGROSS indicates input of an array of net-to-gross multipliers.

FORMAT:

*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) $perm_i$ is replaced by $perm_i$ $*ntg$
- c) $perm_j$ is replaced by $perm_j$ $*ntg$

where por denotes the grid block's porosity as set using the *POR keyword, $perm_i$ and $perm_j$ 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. 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	Lab
min	0.0	0.0	0.0
max	1.0E+4	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.

Please note that the *TRANSI/*TRANSJ/*TRANSK keywords are not cumulative. A transmissibility multiplier applied to a block in recurrent data will overwrite earlier multipliers on that same block.

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 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 n_j to block 1 in direction J when n_j exceeds 1; that is, when a subdivided ring is being closed. In this case, the multiplier from block n_j 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 of 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). *MATRIX multipliers are applied to matrix to matrix flows within a block for *SUBDOMAIN and *MINC.

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 $3 \times 2 \times 1$ 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 suggested range of values for transmissibility multipliers is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1000.0	1000.0	1000.0

Transmissibility Multipliers for Lower Indexed Block Faces (Optional)

***TRANLI, *TRANLJ, *TRANLK**

PURPOSE:

*TRANLI indicates input of I direction transmissibility multipliers for faces contacting lower indexed blocks.

*TRANLJ indicates input of J direction transmissibility multipliers for faces contacting lower indexed blocks.

*TRANLK indicates input of K direction transmissibility multipliers for faces contacting lower indexed blocks.

FORMAT:

*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.

Please note that the *TRANLI/*TRANLJ*/*TRANLK keywords are not cumulative. A transmissibility multiplier applied to a block in recurrent data will overwrite earlier multipliers on that same block.

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.

The suggested range of values for transmissibility multipliers is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1000.0	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.

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} .

<u>Subkeyword</u>	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.

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.
```

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	<i>iregion₁</i>	<i>iregion₂</i>	<i>tmult</i>
	<i>iregion₁</i>	*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.

When defining *INTER_REGION_TM multipliers, multiple definition of regions are allowed, the last definition always overrides/replaces any previous inter-region multiplier definitions. For example:

Examples:

```
**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 all regions is
** 0.25, the inter region multiplier for region 1 and all
** regions except region 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 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.

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.

*PINCHOUTARRAY cannot appear in the same data set as the *PINCHOUT-TOL keyword. Pinched out cells may be set using *PINCHOUTARRAY, or they can be detected by a thickness tolerance set by *PINCHOUT-TOL, but only one technique is allowed per data set. 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 $n_i \times n_j \times n_k$ 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 be connected to each other UNLESS the pinched out cells form an uninterrupted 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 nearly 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 “pinched out” 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 $100 \times 100 \times 9$ 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.

Pore Volume Cut-Off Threshold (Optional)

*PVCUTOFF

PURPOSE:

*PVCUTOFF controls the level at which a block will be set null due to a small pore volume.

FORMAT:

*PVCUTOFF *pvcut*

DEFINITIONS:

pvcut

Pore volume (block volume multiplied by porosity) below which a block will be considered to be null. Dimensions are (m³ | ft³ | cm³).

DEFAULTS:

Optional keyword. Default is to examine the *POR values and any *NULL keyword input for null block determination.

CONDITIONS:

This keyword, if present, must be in the Reservoir Description keyword group.

EXPLANATION:

This option ensures that blocks with small pore volumes can be systematically removed from the simulation. Such small pore volume blocks can hinder convergence and should not remain in a simulation.

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 *pntol*

DEFINITIONS:

pntol

Minimal thickness required under which a block is removed from the simulation and the block above is connected directly to the block below. Dimensions are (m | ft | cm).

DEFAULTS:

Optional keyword. The defaults are:

- 0.0010 (m | ft | cm) for Corner Point grids on the fundamental grid only, if *PINCHOUTARRAY does not appear;
- 0.0002 (m | ft | cm) for non-Corner Point Grids on the fundamental grid only, if *PINCHOUTARRAY does not appear;
- 0.0 (m | ft | cm) 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.

*PINCHOUT-TOL cannot appear in the same data set as the *PINCHOUTARRAY keyword. Pinched out cells may be set using *PINCHOUTARRAY, or they can be detected by a thickness tolerance set by *PINCHOUT-TOL, but only one technique is allowed per data set. 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 uninterrupted 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:

Block whose thickness are less than “pnctol” are considered to be pinched out. When this occurs, blocks above and blocks 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

PURPOSE:

*CORNER-TOL controls the tolerance used to decide if two corner points are the same. It is also used for miscellaneous tolerance checking for corner point applications.

FORMAT:

*CORNER-TOL *cptol*

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 | cm).

DEFAULTS:

Optional keyword. Default is 0.050 (m | ft | cm).

CONDITIONS:

This keyword, if present, must be in the Reservoir Description keyword group.

EXPLANATION:

Corner points that lie within an (Euclidean) distance of *cptol* are considered to be the same for the purpose of deciding whether the cells make contact. 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). Extensions of this notion are used to determine connectivity around faults and when refined grids are present, and *cptol* plays a similar role in these situations.

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 uninterrupted stack of contacting blocks. The presence of a gap will break the connection, where *cptol* gives the tolerance describing how close cells need to be before they are deemed to make contact.

cptol is also used when determining corner orderings and making some miscellaneous checks of cell quality.

Faults (Optional)

*FAULT

PURPOSE:

*FAULT indicates the input of designations for grid blocks whose flow connections are to take into account their exact position in the reservoir with respect to their lateral neighbours. Each *FAULT keyword is expected to describe a group of grid blocks that together form a geological fault block.

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 | cm). (See FIGURE 6 in Appendix D.)

In the simulator, throws provide modifications to depth data given earlier through use of the *DEPTH, *DTOP, *PAYDEPTH or *DEPTH-TOP keywords. A zero throw is valid if the depth information is already complete and only the identification of the grid blocks involved in the geological fault block is required.

i₁:i₂ j₁:j₂

The indices, i_1 , i_2 , j_1 , and j_2 locate grid block columns whose first index (I index) lies between i_1 and i_2 inclusive, whose second index (J index) lies between j_1 and j_2 inclusive, and whose third index (K index) lies between 1 and n_k inclusive.

The grid block columns identified by successive lines of these indices will make up a geological fault block.

DEFAULTS:

Optional keyword. Default: no faults.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group.

*FAULT should not be used with *GRID *CORNER. (Fault data can be entered directly for corner point 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 cannot follow the usual geological strata in these cases. 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 range descriptions which operate as noted above. Note that fault blocks must extend through the entire reservoir. For convenience, a "throw" value can be applied to the depths of all grid blocks in a fault block.

Note that throws can be positive, zero, or negative, and that they will be added directly to the already existing depth values. Thus, comments given earlier regarding depth measurements (see *DEPTH, *DTOP, *PAYDEPTH and *DEPTH-TOP 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 accumulate in the order they are input.)

When it comes time to compute transmissibilities for lateral inter-block flow, and fault blocks are present, special checking is carried out. For instance, if lateral flow into grid block (I,J,K) from its positive I-direction is being considered, which is normally flow from block (I+1,J,K), and *FAULT keywords appeared, the following is done.

If the high side of block (I,J,K) lies on the edge of a fault block (that is, (I,J,K) was identified in a *FAULT list that did not refer to (I+1,J,K)), or any block in the adjacent column has its low side on the edge of a (necessarily) different fault block (that is, (I+1,J,K) was identified in a *FAULT list that did not refer to (I,J,K)), then block (I,J,K) will be connected to ANY block of the form (I+1,J,KK) that has a positive vertical overlap with block (I,J,K).

Moreover, the transmissibility calculation will take into account the amount of actual overlap. A similar calculation will be done for the low side of block (I,J,K), and for the J direction cases.

Vertical transmissibility calculations are not affected by such fault considerations, as are flows internal to fault blocks.

Thus, exact positioning at fault block boundaries governs inter-block flows, as grid blocks in one fault block will no longer align with their usual lateral neighbours.

The acceptable range of values for throws area:

	SI m	Field ft	Lab cm
min	1.0E-3	.00328	0.1
max	1.0E+3	3,280.0	1.0E+5

Aquifers (Optional) see Appendix G *AQUIFER, *AQPROP, *AQMETHOD, *AQLEAK

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  $i_1(:i_2)$   $j_1(:j_2)$   $k_1(:k_2)$  (*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 n_i , or J equals 1 or n_j , or both) in the appropriate I or J direction (n_i = last block in x direction, n_j = last block in y direction).

*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.

*REGION

The aquifer is to be connected to the region indicated by the following range ($i_1:i_2$ $j_1:j_2$ $k_1:k_2$) and direction keywords (*IDIR, *JDIR or *KDIR) keywords.

i₁(:i₂) j₁(:j₂) k₁(:k₂)

Assign cells to aquifer region. This is a table of I-J-K index range triplets, one triplet per line. Each triplet is a mandatory integer, optionally followed by a second integer (not less than the first integer) to specify a range.

***IDIR, *JDIR, *KDIR**

The direction of inflow between the aquifer and the reservoir boundary is to be in the I, J or K direction.

thickness

Aquifer thickness (m | ft | cm).

porosity

Aquifer porosity (fraction).

permeability

Aquifer permeability (md | md | md).

radius

Effective reservoir radius (m | ft | cm).

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 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- 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 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.

Hydrocarbon to aquifer volume ratio Printout

The output is in the block “Hydrocarbon Reservoir Information (including subgrids)” and is called “Hydrocarbon to aquifer volume ratio”.

An example is presented below:

Hydrocarbon to aquifer volume ratio: 1.84110E-02 rm³/rm³

The Aquifer volume includes both gridded and analytical aquifer volumes.

In order to give the correct analytical aquifer volume, the aquifer's R-Ratio must be entered for each aquifer (see keyword *AQPROP). This is true even if the aquifer is a Carter Tracy Aquifer. The Carter Tracy aquifer model does not explicitly use the input R-Ratio in its calculation as the R-Ratio is implicitly entered using the Aquifer pressure influence function tables. However, without the explicit input of the R-Ratio, there is no way for the Simulator to determine the R-Ratio to be used.

For Carter-Tracy aquifers, the R-Ratio is only used for printout purposes, but in order to be meaningful, the R-Ratio input should be identical to the R-Ratio used to derive the pressure influence function table in use. If R-Ratio is allowed to default, it will assume a value of 10.00.

The gridded aquifer volume used in this calculation is estimated by accumulating the volume of grid blocks whose water saturation is at least 99.9% of *WOC_SW. *WOC_SW defaults to 1.00 in IMEX.

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.

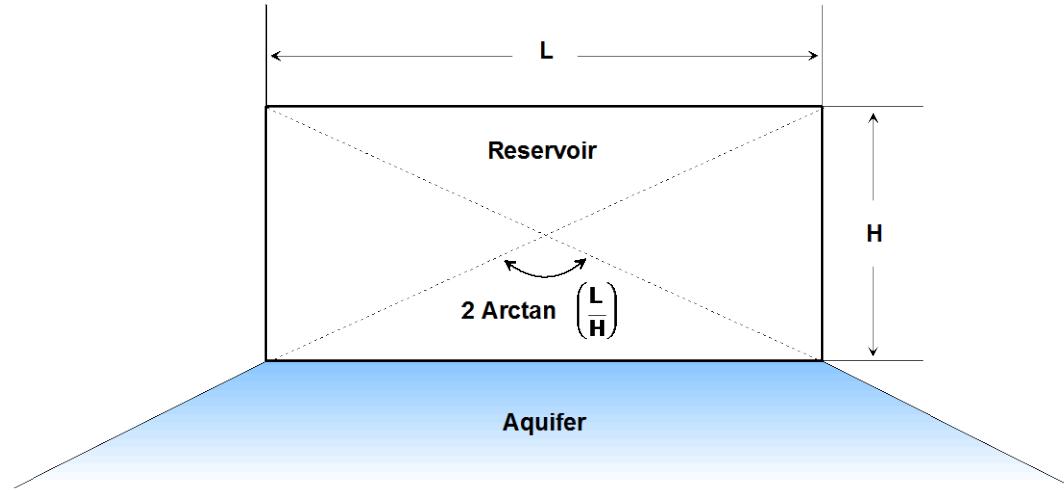
Since outflow from the reservoir into the aquifer is modelled analytically when *AQLEAK *ON is used, it is possible to have water outflow from a grid block attached to the aquifer that has no mobile water. Thus, care must be exercised when using analytical aquifer models. It is recommended that aquifers 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 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 $2*\text{atan}(L/H)$. Then, the theory for radial aquifers is applied in this side view, with the value for Angle taken as $2*\text{atan}(L/H)/360$, Thickness taken as L (viewed as a dimension into the page), and Radius taken as the square root of (L^2H/π) . 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), pp. 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 IMEX 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, pp. 814-828). This approach is able to model finite aquifers without having to generate dimensionless pressure functions. Only a "R-Ratio" parameter is required.

Please be aware that for large aquifers, “R-Ratio” > 5, the Fetkovitch Method which ignores transient effects, may not properly model aquifer influx. For larger aquifers use the Carter-Tracy Method.

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 log(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. If 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 9 *JDIR  
*REGION 4:7 5:7 9 *KDIR
```

A block may have multiple aquifer connections (directions), such as at the corner of a 10×10×5 reservoir. For example, the following specifies aquifer on the bottom plus all sides of a reservoir. Each block on the side edges (e.g., i=1,j=1,k=2:5) has 2 aquifer connections, and each block at the bottom corners (e.g., i=1,j=1,k=1) has three aquifer connections.

```
*AQUIFER *REGION 1:10 1:10 1 *KDIR ** Bottom face  
*REGION 1 1:10 1:5 *IDIR ** I- face  
*REGION 10 1:10 1:5 *IDIR ** I+ face  
*REGION 1:10 1 1:5 *JDIR ** J- face  
*REGION 1:10 10 1:5 *JDIR ** J+ face
```

The acceptable range of values for the aquifer thickness is:

	SI m	Field ft	Lab cm
min	0.0	0.0	0.0
max	1.0E+4	3.28E+4	1.0E+6

Pressure Influence Function (Conditional) see Appendix G

***AQFUNC**

PURPOSE:

*AQFUNC indicates input of a dimensionless pressure influence function as a table.

TABLE:

*AQFUNC
dime time function
: :
:

DEFINITIONS:

dime time
Dimensionless time

function
Dimensionless pressure influence function

DEFAULTS:

Conditional keyword. See EXPLANATION for a discussion of the default table.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group and must be used with *AQUIFER.

EXPLANATION:

The simulator calculates water influx from (to) 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 requires the use of a dimensionless pressure influence function $P(td)$ as a function of dimensionless time td (see Appendix G).

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 and an infinite radial aquifer. Influence functions for limited extent aquifers can also be found in this reference.

The default pressure influence functions and the corresponding dimensionless times for the default case are tabulated below:

Dimensionless Time, td	Influence Function, P(td)
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
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

Additional values may be found in the above reference.

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 IMEX are given above under the *AQUIFER section.

Sectors (Optional)

*SECTOR

PURPOSE:

*SECTOR controls the definitions of sectors, which are used to summarize regional reservoir activity.

FORMAT:

```
*SECTOR      'sector'  
           { i1(:i2) j1(:j2) k1(:k2) }
```

DEFINITIONS:

'sector'

Sector identification name, enclosed in single quotes (16 character maximum).

{ i₁(:i₂) j₁(:j₂) k₁(:k₂) }

Assign cells to sector 'sector'. This is a table of I-J-K index range triplets, one triplet per line. Each triplet is a mandatory integer, optionally followed by a second integer (not less than the first integer) to specify a range.

DEFAULTS:

Optional keyword. The first sector is internally defined to include all active grid blocks and is named: 'Entire Field'. Other sectors defined as above are assigned higher sector numbers.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group. Keywords *SECTOR, *SECTORARRAY and *SECTORNAMES may not be used together.

EXPLANATION:

Sectors can be used for printing the values of several quantities relating to the grid blocks that were assigned to the sector. These quantities include:

- Total and hydrocarbon pore volume
- Fluids in place at surface and reservoir conditions
- Average Pressure for the total pore volume
- Average pressure with datum correction for total pore volume
- Average pressure for the hydrocarbon pore volume
- Average saturation for oil, gas, water or solvent
- Average concentration for polymers
- Percentage recovery, defined as:

$$\% \text{RC} = \frac{(\text{cumulative production} - \text{cumulative injection})}{(\text{initial material in place})}$$

- Cumulative injection and production values
- Production and injection rates

The frequency of printing the table containing these quantities to the output-file is controlled by the keyword *WPRN *SECTOR. The same comments apply to *WSRF *SECTOR and the writing of sector statistics to the SR2.

Grid blocks can belong to several sectors. Of course, all grid blocks are in the first sector by default.

Example:

A grid system with some desired sector allocations is given below:

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 will define those sectors:

```
*SECTOR    'areal'    1:3    4:6    1
           4       6     1
           'area2'   3:6    1     1
*SECTOR    'area2'   4:7    2     1
           5:7    3     1
*SECTOR    'area2'   6:7    4     1
           7       5     1
```

Note that there is considerable flexibility in the way that the *SECTOR keyword can be used. See the following *SECTORARRAY and *SECTORNAMES keywords for alternative methods to specify sector data.

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 $n_i \times n_j \times n_k$ 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). Keywords *SECTOR, *SECTORARRAY and *SECTORNAMES may not be used together.

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. Specifically, *SECTORARRAY allows you to refer to particular grids created by *REFINE as well as matrix versus fracture cells. The array qualifiers and values should come after 'Sector_Name'. See **Input of Grid Property Arrays** in chapter "Keyword System".

Example:

To set sector membership in a sector named 'Sector-1' for a few cells in the reservoir and a locally refined 3×3×2 grid in (3,3,1), use the following:

```
*SECTORARRAY 'Sector-1' *IJK 1:5 1:5 1:1 0
              4   4   1   1
              2   3   1   1
*SECTORARRAY 'Sector-1' *RG 3 3 1 *IJK 1 1 1:2 1
              2 2 1:2 1
              3 3 1:3 1
```

where it is assumed the grid is dimensioned 5×5×1. 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* ...

*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 must appear once and must appear before *ISECTOR. All array qualifiers and array reading options are allowed for specifying the required $n_i \times n_j \times n_k$ values for *ISECTOR. The values assigned by *ISECTOR should consist of the values that appear in the *SECTORNAMES list. Keywords *SECTOR, *SECTORARRAY and *SECTORNAMES may not be used together.

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 $5 \times 5 \times 2$. Provided that this is 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 Sector (Optional)

***FLUX_SECTOR NAMES, *FLUX_ISECTOR**

PURPOSE:

The keyword group flux sector introduces special sector definitions which are used to calculate reservoir flow between these “flux sectors”. Flow into these sectors and between these sectors can be reported to the output print file and to the simulation results file as time series data. Flow can be further broken down into I-direction, J-direction and K-direction flow components.

*FLUX_SECTOR NAMES defines a list of flux sector names and their corresponding flux sector numbers. *FLUX_ISECTOR assigns these flux sector numbers to cells using standard array input options. Both keywords can only appear once in the data set.

FORMAT:

```
*FLUX_SECTOR NAMES 'Flux_Sector_Name_1' i1 'Flux_Sector_Name_2' i2 ...  
*FLUX_ISECTOR
```

DEFINITIONS:

'Flux_Sector_Name_1' i1

'Flux_Sector_Name_2' i2

...

Flux sector names (16 characters maximum) and their associated numbers which must be greater than one.

DEFAULTS:

Optional keyword. If a grid block is left undefined via *FLUX_ISECTOR it will be given a zero value as the sector number.

CONDITIONS:

These keywords must be in the RESERVOIR DESCRIPTION keyword group.

The keyword *FLUX_SECTOR NAMES should appear before keyword *FLUX_ISECTOR. All array qualifiers and array reading options are allowed for specifying the required values for *FLUX_ISECTOR. The values assigned by *FLUX_ISECTOR should be one of the values that appear in the *FLUX_SECTOR NAMES list.

EXPLANATION:

Flux sectors are collections of grid blocks like normal sectors. Unlike normal sectors, flux sectors are used to determine flow between these 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 also be calculated.

The *FLUX_ISECTOR keyword provides an array-based assignment for flux sector numbers that have been defined using *FLUX_SECTORNAMES. As the result, each defined grid block holds a non-zero flux sector number. Undefined grid blocks will hold a zero flux sector number which will not be included in the flux sector reports. The map of flux sectors thus comes out without overlaps. This is different from normal sectors which are oriented to inplace calculations and therefore allow overlaps.

The flux sector does not conduct inplace calculations as the normal SECTOR option does. However, if necessary, maps of flux sectors can simply be duplicated in SECTOR option via similar pair of keywords *SECTORNAMES and *ISECTOR.

Example:

To set flux sector membership in four sectors named 'Area-I', 'Area-II', 'Area-III' and 'Area-IV' use the following:

```
*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
```

where it is assumed that the grid is dimensioned 13x14x6.

The output frequency for flux sectors is controlled by *WPRN *SECTOR and *WSRF *SECTOR. Please also see *OUTPRN *FLUX-SECTOR and *OUTSRF *FLUX-SECTOR for output options.

Lease Planes (Optional)

***LEASE**

PURPOSE:

*LEASE defines a lease plane which is used to summarize various regional flows.

FORMAT:

*LEASE 'lease' $i_1(:i_2)$ $j_1(:j_2)$ $k_1(:k_2)$ (*IDIR | *JDIR | *KDIR)

DEFINITIONS:

'lease'

Lease plane identification name, enclosed in single quotes (16 character maximum).

$i_1(:i_2)$ $j_1(:j_2)$ $k_1(:k_2)$

Assign cells to lease plane 'lease'. This is a table of I-J-K index range triplets, one triplet per line. Each triplet is a mandatory integer, optionally followed by a second integer (not less than the first integer) to specify a range.

*IDIR, *JDIR, *KDIR

Lease plane is perpendicular to the I, J or K direction.

DEFAULTS:

Optional keyword. Default: no lease data.

CONDITIONS:

This keyword must be in the Reservoir Description keyword group.

EXPLANATION:

Lease planes can be used to summarize and report the flow of fluids across planes or regions in the reservoir.

Lease planes are composed of collections of subregions of the form " $i_1:i_2 j_1:j_2 k_1:k_2$ " that are each assigned a flow direction, indicated by one of the *IDIR, *JDIR, *KDIR keywords.

Usually if *IDIR is specified, i_1 will be the same i_2 . The same will hold for the other directions. If this is not the case, the flow calculation will accumulate the net flow across the region.

Flows are determined for each subregion by first calculating the flow into each block from its lower indexed neighbour, as determined from the direction specified for that subregion. These contributions are then accumulated over the subregion and reported. (Note that flow from a lower indexed block to its higher indexed neighbour is accumulated with a positive sign.)

Thus, if grid block (I,J,K) lies in a subregion carrying the *IDIR direction indicator, the flow from block (I-1,J,K) to (I,J,K) is added into the subregion's total. (This calculation is only done if (I-1,J,K) is a valid block in communication with block (I,J,K).) Similar comments hold for the other directions.

Example:

```
*LEASE  'LEASE1'  5:5  1:10  1:3  *IDIR
```

can be used to calculate the flow across a plane perpendicular to the I direction and separating all the grid points (5,J,K) from the grid points (4,J,K) where J=1,..,10 and K=1,..,3.

Irregular Connection (Optional)

*IRCONNECT

PURPOSE:

Specify irregular interblock connection data.

FORMAT:

*IRCONNECT { *uba₁* *uba₂* *L₁* *L₂* *A₁₂* (*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.

uba₁ *uba₂*

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.

L₁ *L₂*

Distance (m | ft | cm) between each cell's center (pressure reference location) and their common face. *L₁* is the distance in cell *uba₁*, and *L₂* is the distance in cell *uba₂*.

A₁₂

Area (m^2 | ft^2 | cm^2) of the face common to cells *uba₁* and *uba₂*. *A₁₂* 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.

<u>Physical Process</u>	<u>Analogous Property</u>
Fluid convection	Permeability
Molecular diffusion	Diffusivity
Thermal conduction	Thermal conductivity
Electrical conduction	Electrical 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.

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_desc1 2nd_cell_desc1 trms1
              1st_cell_desc2 2nd_cell_desc2 trms2
              1st_cell_desc3 2nd_cell_desc3 trms3
              ...
              ...
```

DEFINITIONS:

1st_cell_desc_n *2nd_cell_desc_n*

Descriptions of active cells. If the cell is on the 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_n

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 | md-ft).

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

This keyword, if present, must be in the Reservoir Description keyword group.

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 *SCONNECT 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.
...
```

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, GridBuilder 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*, *njhight* 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 (or *j*+1).

The *FAULTARRAY binary integer flag uses the following convention:

nilow, *nihigh*, *njlow*, *njhight* = 0 if the connection is a standard connection

nilow, *nihigh*, *njlow*, *njhight* = 1 if the connection is a fault connection

The value of *FAULTARRAY for a block is:

$$\text{IVAL} = \text{nilow} + 2\text{nihigh} + 4\text{njlown} + 8\text{njhight}$$

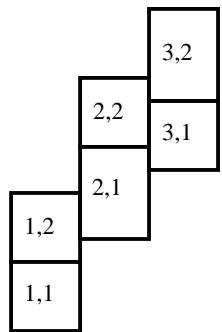
Thus if all connections are standard $\text{IVAL} = 0$, and if all connections take into account block depths (are fault connections), $\text{IVAL} = 15$.

Use:

```
**all connections are really 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



i →

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.

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 $\text{por}(p)$ as a function of a constant rock compressibility, a reference pressure and reference porosity (below),

$$\text{por}(p) = \text{por_input} * [1 + \text{ccpor} * (\text{p} - \text{cprpor})]$$

where:

<i>por_input</i>	:	initial porosity
<i>ccpor</i>	:	constant rock compressibility
<i>cprpor</i>	:	constant reference pressure
<i>p</i>	:	pressure
<i>por(p)</i>	:	porosity at pressure <i>p</i>

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.

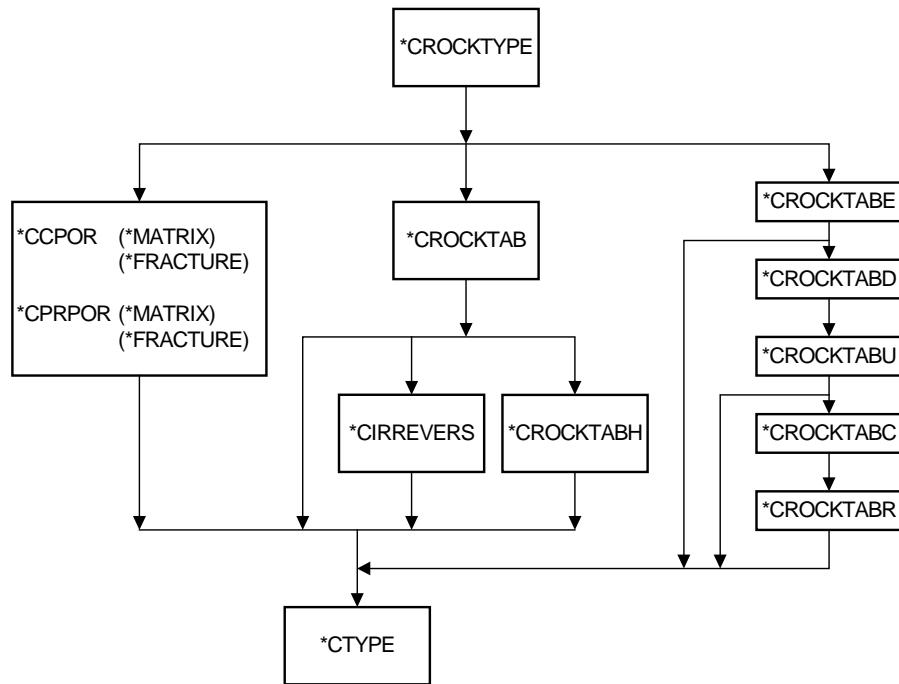
If the constant rock compressibility and reference pressure option is used, the equation above is used to calculate the porosity at pressure *p*. If the table of multipliers vs. pressure option is used, the relation below is applied to calculate porosity at pressure *p*:

$$por(p) = por_input * mul_por(p)$$

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.*



Keyword Relation in Rock Compaction/Dilation Model

The figure above 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, other keywords under it are invalid. Descriptions and usage of all the compaction/dilation keywords can be seen on consecutive pages. It is also noted that keywords used for the rock compaction model can not 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 your reservoir model. However, the two regions must not be in communication with each other and must be separated by null blocks. Examples of using those keywords follow the keyword descriptions.

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 *FRACTURE)	<i>ccpor</i>
*CPRPOR	(*MATRIX *FRACTURE)	<i>cprpor</i>

DEFINITIONS:

ccpor

Pressure dependence of formation porosity; that is, rock compressibility.
(1/kPa | 1/psi | 1/kPa | 1/(kg/cm²))

cprpor

Reference pressure for calculating the effect of rock compressibility *ccpor*.
(kPa | psi | kPa | kg/cm²)

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 | kPa | kg/cm²).

por_mult

porosity multiplier (dimensionless) = V/V_i

where:

V is the pore volume measured at the pressure *press*.

V_i 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) = $\text{perm}_h/\text{perm}_{hi}$

where:

perm_h is the horizontal permeability measured at the pressure *press*.

perm_{hi} is the input (initial) horizontal permeability. *perm_{hi}* 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) = $\text{perm}_v/\text{perm}_{vi}$

where:

perm_v is the vertical permeability measured at the pressure *press*.

perm_{vi} is the input (initial) vertical permeability. *perm_{vi}* 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. If values of the porosity multipliers in the table are all same, zero rock compressibility will be assumed. If *DEPLETION is on, this may result in invalid block reference pressure.
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. 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 bottom hole 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:

<i>por_input</i>	:	initial input porosity (*PORI)
<i>mul_por(p)</i>	:	porosity multiplier at a pressure <i>p</i>
<i>por(p)</i>	:	porosity at pressure <i>p</i>

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:

press

Minimum pressure is 101 kPa (14.7 psia) (kPa | psi | kPa | kg/cm²).

por_mult

porosity multiplier (dimensionless) = V/V_i

where:

V is the pore volume measured at the pressure *press*.

V_i 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) = $perm_h/perm_{hi}$

where:

perm_h is the horizontal permeability measured at the pressure *press*.

perm_{hi} is the input initial horizontal permeability. *perm_{hi}* 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) = $perm_v/perm_{vi}$

where:

perm_v is the vertical permeability measured at the pressure *press*.

perm_{vi} is the input initial vertical permeability. *perm_{vi}* 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. First pressures in subtables must increase monotonically from the first subtable to the last subtable.
9. 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.
10. The *INT table input option does not apply to this table.

EXPLANATION:

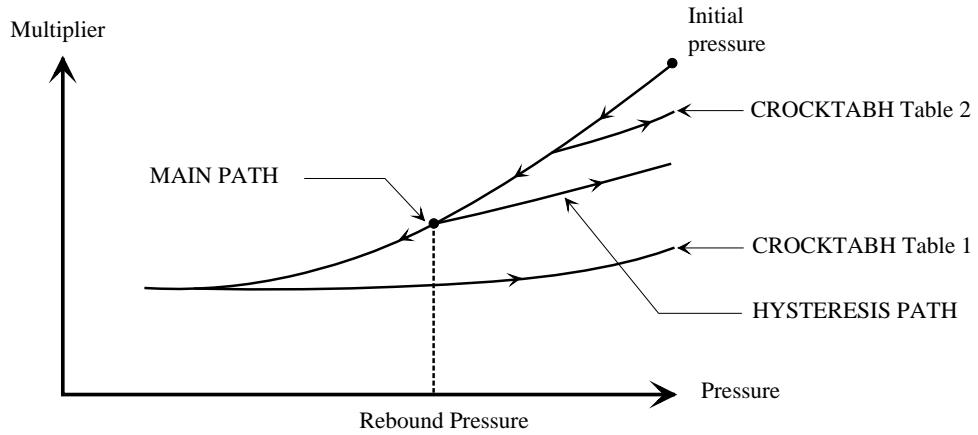
This keyword is designed to model porosity and permeability multipliers versus pressure behavior that exhibits hysteresis effects. In the explanation below, porosity behavior will be described but the behavior also applies to permeability multipliers.

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.

When the *CTROCKTAB table is quite nonlinear it is important to define a sufficient number of hysteresis branches using *CROCKTABH. If the *CROCKTABH data is sparse it is possible that there will be a porosity discontinuity at the rebound pressure. This can be minimized by increasing the number of *CROCKTABH tables and can be eliminated entirely by defining a *CROCKTABH table for every pressure point in the main table.



Hysteresis Process for a Rock Type with Two *CROCKTABH Tables

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 options.

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 ($Prs_0(iblk)$). 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 $Prs_0(iblk)$).

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

$$P^* = P_{shift}(itype) + Prs(iblk) - Prs_0(iblk).$$

Where $Prs(iblk)$ is the current grid block pressure, $Prs_0(iblk)$ is the initial grid block pressure, $itype$ is the *CROCKTYPE of grid block $iblk$ and $P_{shift}(itype)$ is the pressure in the *CROCKTAB table of *CROCKTYPE $itype$ where the Porosity Multiplier (por_mult) is equal to or closest to 1.00.

IMEX will interpolate within the *CROCKTAB table to find the best value of P_{shift} .

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 $P_{shift}(itype)$. 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 $P_{shift}(itype)$ by $Prs(iblk) - Prs_0(iblk)$.

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, IMEX will printout the value of $P_{shift}(itype)$ 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 P_{shift} 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 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:

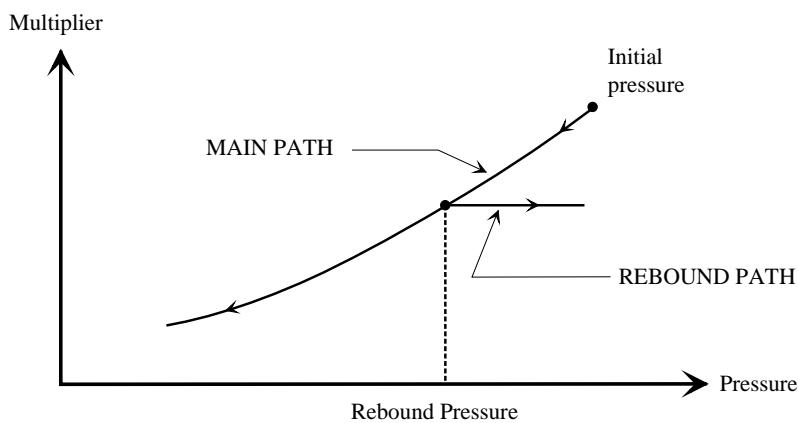
This keyword must be located in Reservoir Description section.

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 | kPa | kg/cm²).

Maximum pressure is a yield point pressure of a rock type.

por_mult

Porosity multiplier (dimensionless) = V/V_i

where:

V is the pore volume measured at the pressure *press*

V_i 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) = $\text{permh}/\text{permhi}$

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) = $\text{permv}/\text{permvi}$

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 *CROCKTABLE 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 *CROCKTABLE 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. Minimum pressure in the table must be greater than zero.
8. There is only one keyword *CROCKTABLE for one rock type.
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 *CROCKTABLE, 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 | kPa | kg/cm²) which is equal to maximum pressure in the *CROCKTABE table.

por_mult

Porosity multiplier (dimensionless) = V/V_i

where:

V is the pore volume measured at the pressure *press*

V_i 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. First pressure in the table must be the same as that of the last pressure in the *CROCKTABE table for the same rock type.
6. There is only one keyword *CROCKTABD for one rock type.
7. This keyword can not be combined with any other keywords related to the rock compaction model.
8. 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 | kPa | kg/cm²). Minimum pressure must be greater than zero.

por_mult

Porosity multiplier (dimensionless) = V/V_i

where:

V is the pore volume measured at the pressure *press*

V_i 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. First pressure in each table must be the same as one of pressures in the *CROCKTABD table for the same rock type.
6. There must be at least two *CROCKTABU tables for one rock type.
7. Each subsequent *CROCKTABU table must start at a pressure higher than the last.
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 *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 | kPa | kg/cm²).

por_mult

Porosity multiplier (dimensionless) = V/V_i

where:

V is the pore volume measured at the pressure *press*

V_i 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. First pressure in the table must be the same as that of the last pressure in the *CROCKTABU table for the same rock type.
6. There must be at least two *CROCKTABC tables for one rock type corresponding to at least two *CROCKTABU tables.
7. The minimum pressure in the table must be greater than zero.
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 *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 | kPa | kg/cm²). 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/V_i

where:

V is the pore volume measured at the pressure *press*

V_i 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:

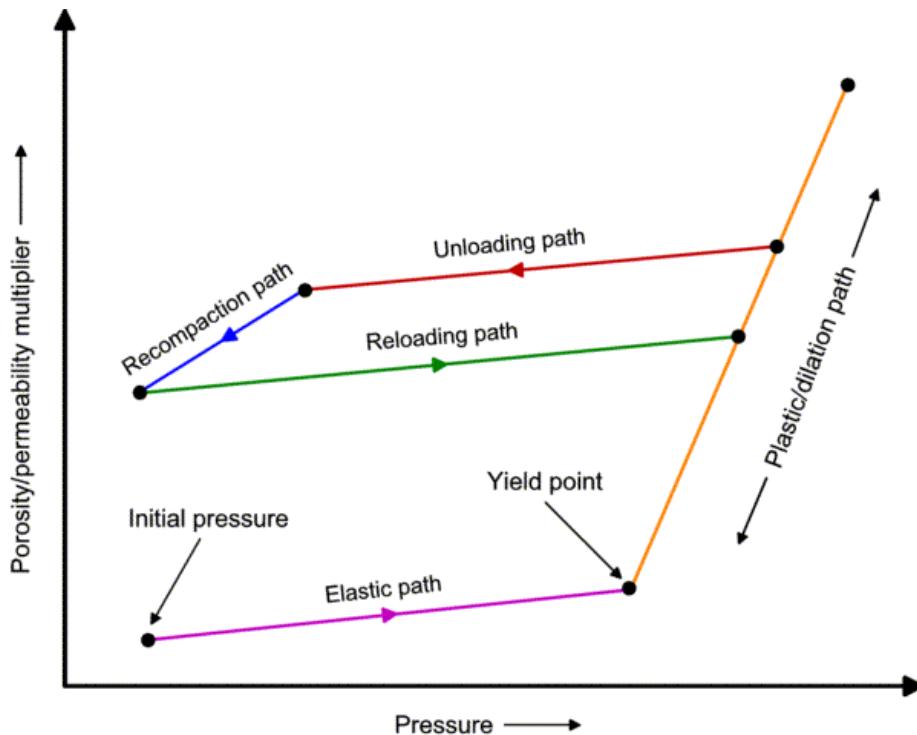
1. This keyword must be located in Reservoir Description section.
2. The first two columns in the table such as pressure and porosity multiplier always exist. The permeability multipliers (third and fourth columns) are optional.
3. Pressure in the table should increase monotonically down the column.
4. First pressure in the table must be the same as that of the last pressure in the *CROCKTABC table for the same rock type.
5. 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.
6. There must be at least two *CROCKTABR tables for one rock type.
7. The minimum pressure in the table must be greater than zero.
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:

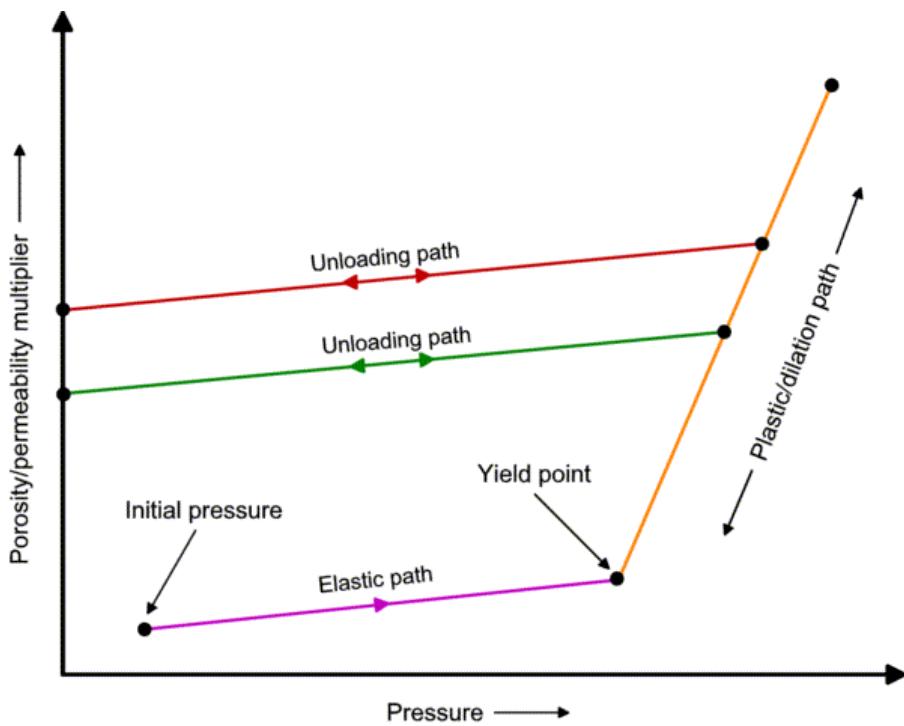
Under the keyword *CROCKTABR, multipliers (porosity and permeability) at one pressure are computed on the basis of table lookup and linear interpolation methods applied to two unloading paths given by two *CROCKTABR tables for one rock type. On the recompaction path, when pressure in the rock increases, 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:

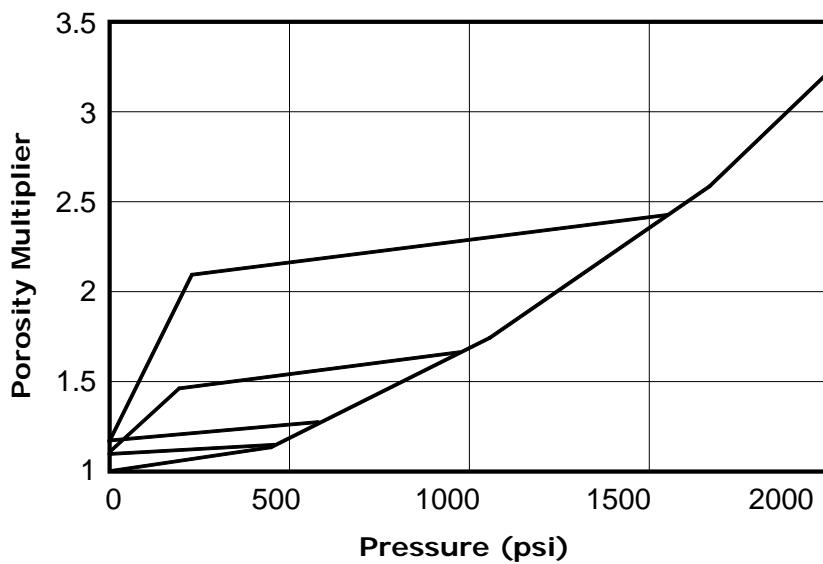
```

*CROCKTYPE 1
*CROCKTABLE
** 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

```



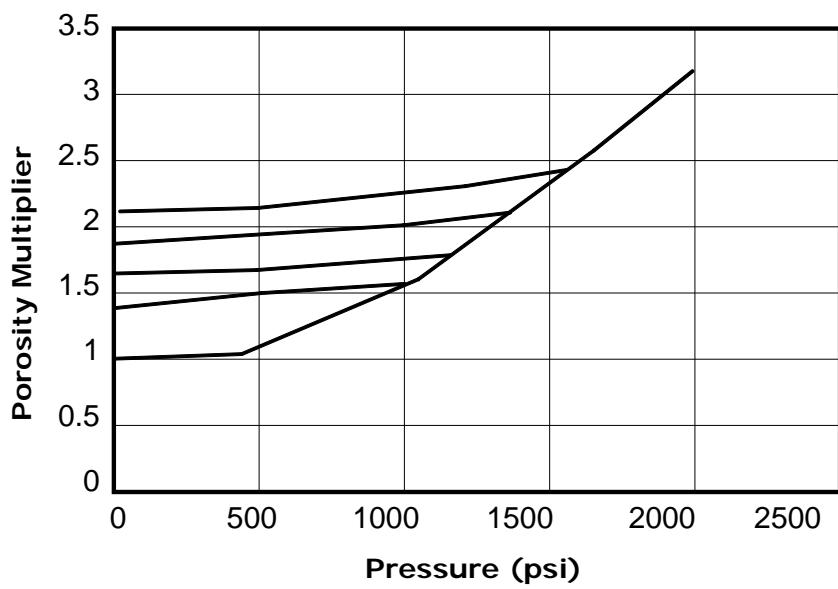
Example2: A sample data of open loop dilation model is given as below:

```

*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		por.mult	hor.trans.mult	ver.trans.mult
** pressure				
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		



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).

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 switched tables MUST contain identical porosity multipliers.

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 50×50×4 grid blocks.

Example 1:

```
*CROCKTYPE      1
*CROCKTAB: table of rock compaction data for rock type 1
**pressure      porosity          perm.          perm.
                multiplier      multiplier (horiz.)    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     1.0765494    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

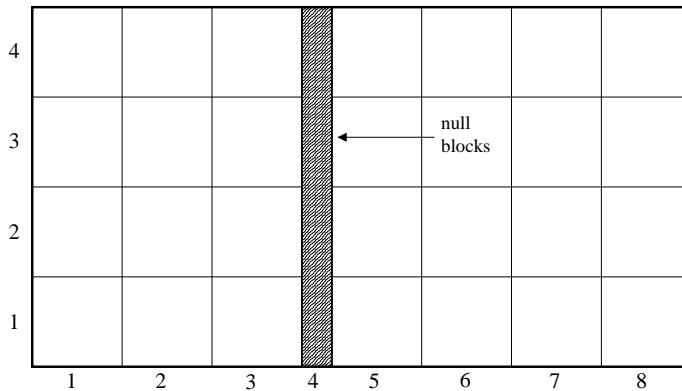
```

```

*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.

Dispersion Coefficients (Conditional)

*DISPI, *DISPJ, *DISPK

PURPOSE:

Input of polymer dispersion coefficients for the I, J and K directions.

ARRAY:

*DISPI
*DISPJ
*DISPK

Please note in the 2002.10 release the time unit of the dispersion coefficient changed from seconds to the time unit which is consistent with the value of *INUNIT. The user must alter data accordingly.

DEFINITIONS:

*DISPI

Indicates these are dispersion coefficients for the I direction (m^2/day | ft^2/day | cm^2/min).

*DISPJ

Indicates these are dispersion coefficients for the J direction (m^2/day | ft^2/day | cm^2/min).

*DISPK

Indicates these are dispersion coefficients for the K direction (m^2/day | ft^2/day | cm^2/min).

DEFAULTS:

Conditional keyword. By default, all dispersion coefficients are 0.0

CONDITIONS:

This keyword must be in the Reservoir Description keyword group. This keyword is only used with *MODEL *POLY or *MODEL *POLYOW.

EXPLANATION:

The dispersion coefficient keywords are arrays. All array reading option subkeywords are valid. *EQUALSI can be used to set *DISPJ and *DISPK after *DISPI has been input. See the section on Array Reading Options for more details.

Example:

Enter the dispersion coefficients for the polymer, when it is a constant value for all directions and all fracture blocks.

```
*DISPI *FRACTURE *CON 0.25
*DISPJ *EQUALSI
*DISPK *EQUALSI
```

Physical dispersion may be calculated using the equation:

$$\text{disp} = \frac{\text{molard}}{L} + \alpha * \frac{|v|}{\theta}$$

where disp is the dispersion coefficient in the i, j, or k directions, molard is the molar diffusion coefficient, L is the tortuosity, alpha is the dispersion parameter in the i, j, and k direction, and v is the velocity of the water phase.

Please refer to Appendix E for more information about dispersion coefficients.

The acceptable range of values for the dispersion coefficients is:

	SI m²/day	Field ft²/day	Lab cm²/min
min	0.0 1.0E+20	0.0 1.076E+21	0.0 1.0E+24

Fault Transmissibilities (Optional)

*TRANSF

PURPOSE:

Adjusts transmissibilities on a fault basis.

FORMAT:

*TRANSF	'Fault_Name'	<i>fault_trans_mult</i>
<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:

$i_1 j_1 k_1$ [*IDIR or *JDIR or *KDIR] $i_2 j_2 k_2$

which refers to an existing connection between cells, while a “single” identifier looks like:

$i_1 j_1 k_1$ [*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-/+ 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.

Component Properties

Fluid Model (Required)

*MODEL

PURPOSE:

*MODEL signals the input of the fluid component model to use for the simulation.

FORMAT:

```
*MODEL      (*BLACKOIL)
            (*OILWATER)
            (*MISCG)
            (*MISNCG)
            (*POLY)
            (*POLYOW)
            (*API-INT)
            (*API-INTOW)
            (*GASWATER)
            (*GASWATER_WITH_CONDENSATE)
            (*BLACKOIL_SEAWATER)
            (*OILWATER_SEAWATER)
            (*VOLATILE_OIL)
```

DEFINITIONS:

*BLACKOIL

Use a black-oil model, modelling flow of oil, water, and gas.

*OILWATER

Use a two phase, oil and water model, with no modelling of free gas, or variation in solution gas.

*MISCG

Use a pseudo-miscible model, with chase gas. Assumes solution gas always remains in solution.

*MISNCG

Use a pseudo-miscible model with no chase gas. Assumes gas injected (if any) has the same composition as solution gas.

***POLY**

Use a polymer model, modelling the flow of oil, water, gas, and polymer.

***POLYOW**

Use a polymer model, with no gas flow or variation in solution gas.

***API-INT**

Use an API tracking model, handling three phases (water, oil and gas) and four components (water, gas and two oil components). The two oil components flow only in the oil phase while the gas component can be present both in the oil and gas phases. See: *PVTAPI, *APIGRAD, *API, *APIT, *APIGRAV, *BOTAPI, *VOTAPI.

***API-INTOW**

Use an API tracking model, with no gas flow or variation in solution gas. See: *PVTAPI, *APIGRAD, *API, *APIT, *APIGRAV, *BOTAPI, *VOTAPI.

***GASWATER**

Use a two phase gas and water model, with no oil phase modelling. Only two equations are solved for simultaneously. Gas PVT properties are entered using the *PVTG option. Krow, Krog, So, Pb, Co and Cvo are not input. PCGW, if required, is input on the *SWT table and is properly handled in both *BLOCK_CENTER and *DEPTH_AVE *TRANZONE initialization methods.

***GASWATER_WITH_CONDENSATE**

Uses an extension of the *GASWATER option. Oil (as condensate) can initially exist in the gas phase either in a saturated or undersaturated state. Condensate can be produced at surface or can drop out in the reservoir (and possibly be produced as a liquid). Condensate, Gas, and Water equations are solved for simultaneously.

Saturated gas PVT properties are entered using the PVTCOND table.

Undersaturated GAS PVT properties are entered using the

*EGUST/*BGUST/*ZGUST and *VGUST keywords. Condensate properties are entered using oil phase keywords (e.g. *DENSITY *OIL).

The keywords associated with the use of this option are: *PVTCOND, *EGUST, *VGUST, *PDEW, *PDEWT, *PSPLIT, *OUTPRN/*OUTSRF GRID (*DPP | *RV), *NORM *PDW and *MAXCHANGE *PDW

***BLACKOIL_SEAWATER**

Use a seawater injection model, modelling the flow of oil, water, gas, and seawater. This model tracks seawater production and reduces well productivity accordingly if used with the Scale Buildup/Well Damage model.

***OILWATER_SEAWATER**

Use a seawater injection model, with no gas flow or variation in solution gas.

This model tracks seawater production and reduces well productivity accordingly if used with the Scale Buildup/Well Damage model.

The *BLACKOIL_SEAWATER and OILWATER_SEAWATER models simply track seawater flow in the reservoir. Seawater viscosity is modeled separately from formation water viscosity (See *SVISC). The “Seawater” models are designed to be coupled to the Scale Deposition/Damage model defined using the *SCLDPS, *SCLDMG, *SCLTBL-WELL, *SCLTBL-LAYER, *SCLRMV-WELL and *SCLRMV-LAYER keywords. This allows scale buildup and well damage to be both a function of water produced and seawater volume fraction in the produced water.

The Scale Deposition/Damage model can be used without the Seawater model, however; when this is done well damage is only a function of formation water produced.

***VOLATILE_OIL**

Uses an extension of the *BLACKOIL option. Gas can exist in solution in the oil phase and oil can exist in the gas phase. This option is similar to the *GASWATER_WITH_CONDENSATE. It is more flexible in that the gas phase need not always exist. This allows the model to be used in situations where oil can be above its bubble point.

The model initializes like a *BLACKOIL model and uses either PVTCOND or PVTVO in place of the *BLACKOIL model’s PVT table. Both *PB and *PDEW need to be defined with this model. In addition *DWOC and *DGOC need to be defined. It is possible to use *BGUST and *VGUST to describe undersaturated gas conditions and *BOT and *VOT tables (or *CO, *COT and *CVO) to describe undersaturated oil conditions when using the *PVTCOND saturated table. When using the *PVTVO table *BOT and *VOT tables can be used to define undersaturated oil conditions.

DEFUALTS:

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.

EXPLANATION:

Thirteen different fluid models are available for use in IMEX. Only one option may be entered. When the *OILWATER keyword is used, the model solves only two equations per grid block. The gas phase is absent and consequently is not modelled. When the *GASWATER keyword is used, the model solves only two equations per grid block. The oil phase is absent and consequently is not modelled.

When the chase gas option, *MISCG, is used the solution gas always remains in solution. The solution GOR is fixed and is input by the user using the *GORINT keyword. The bubble point pressure vs. Rs curve in the PVT table belongs to the chase gas. So the total amount of dissolved gas in oil consists of two parts. The solution gas given by *GORINT and the dissolved chase gas given by the Rs vs. bubble point pressure curve, or

$$\text{Total_Rs} = \text{GORINT} + \text{Chase_gas_Rs}$$

PVT DATA:

PVT tables may be entered directly using the *PVT keyword. Alternatively, a differential liberation table may be entered using the *DIFLIB keyword. If the *DIFLIB option is used then the *BUBBLE, *BOBF and *RSIF keywords must also be present in this data section. One of *PVT or *DIFLIB must be present in the data set.

If the *API-INT or *API-INTOW options are used then the PVT properties MUST be specified using the keywords *PVTAPI and *APIGRAD.

If the *GASWATER option is used then the PVT properties MUST be specified using the *PVTG keyword.

If the *GASWATER_WITH_CONDENSATE option is used then the PVT properties MUST be specified using the *PVTCOND keyword.

If the *VOLATILE_OIL option is used then the PVT properties MUST be specified using either the *PVTCOND or *PVTVO keyword. For multiple PVT region models, all regions must use one type of table (*PVTCOND or *PVTVO).

Reservoir Temperature (Optional)

*TRES

PURPOSE:

*TRES indicates the input of reservoir temperature.

FORMAT:

*TRES *value*

DEFINITIONS:

value

Value of reservoir temperature. (F | C | C)

DEFAULTS:

Required keyword. No default values.

CONDITIONS:

These keywords must be in the Component Property keyword group. This keyword is required if the *ZG option will be used to input the gas compressibility in the PVT table.

The value of *TRES entered acts on PVT tables which follow its occurrence. Multiple *TRES keywords may be entered. PVT table temperature is determined from the last defined value of *TRES.

EXPLANATION:

The reservoir temperature specified here is used to convert the Z-factor of gas Zg to the gas expansion factor Eg as follows:

It is assumed that the stock tank conditions are 14.7 psia, (101.325 kPa | 1.03 kg/cm²) and 60 F, (15.56C).

Then in field units:

$$Eg = 198.588 * P/ZT \text{ (SCF/Res.BBL)}$$

with P in psia
and T in R

and in SI and Lab units:

$$Eg = 2.84935 * P/ZT \text{ (Std.m}^3/\text{Res.m}^3\text{)}$$

with P in kPa
and T in K

Example:

*TRES 180.0

The acceptable range of values for reservoir temperature is:

	SI C	Field F	Lab C
min	-100	-100	-100
max	1000	1000	1000

Oil and Gas PVT Table (Optional)

*PVT

PURPOSE:

*PVT indicates start of the oil and gas PVT table.

TABLE:

*PVT	(*EG *BG *ZG)	set_number						
p	rs	bo	eg bg zg	viso	visg	(co)	(srftm)	
:	:	:	:	:	:	:	:	:

DEFINITIONS:

set_number

Set number for this particular PVT region. This is the number used with *PTYPE to assign PVT tables to grid blocks.

*EG

Keyword indicating that the gas expansion factor will be used. If all *EG, *BG and *ZG are not present in the data set then this is the default.

*BG

Keyword indicating that the gas formation volume factor will be used instead of the gas expansion factor.

*ZG

Keyword indicating that the gas compressibility factor will be used instead of the gas expansion factor. The reservoir temperature TRES must be specified before the *PVT keyword in order to use this option.

p

Pressure (kPa | psi | kPa | kg/cm²).

rs

Solution gas-oil ratio (m³/m³ | scf/STB | cm³/cm³) for saturated oil at pressure p. When the chase gas option is used this is the Rs of the dissolved chase gas.

bo

Formation volume factor (m³/m³ | RB/STB | cm³/cm³) for saturated oil at pressure p.

eg

Gas expansion factor (m³/m³ | scf/RB | cm³/cm³) at pressure p. When the chase gas option is used this is the chase gas eg.

bg

Gas formation volume factor (m³/m³ | RB/scf | cm³/cm³) at pressure p. When the chase gas option is used this is the chase gas bg.

zg

Gas compressibility factor (dimensionless) at pressure p . When the chase gas option is used this is the chase gas *zg*.

viso

Viscosity (mPa-s | cp | cp) of saturated oil at pressure p .

visg

Gas viscosity (mPa-s | cp | cp) at pressure p . When the chase gas option is used this is the chase gas viscosity.

srftn

Gas-oil interfacial tension. It must be monotonically decreasing and greater than zero (dyne/cm).

co

Oil compressibility. It must remain constant or be monotonically increasing. Oil compressibility may be entered in the *APIGRAD table when using API tracking option or by using the *BOTAPI table. Other entry methods (*CO *COT) are disabled.

DEFUALTS:

Optional keyword. No default values. One of *PVT, *DIFLIB *PVTAPI *PVTG, *PVTCOND or *PVTVO must be present in the data set.

CONDITIONS:

This keyword must be in the Component Property keyword group.

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure. Surface tension is optional.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired.

When the two phase (oil and water) option is used, there are two ways of specifying the initial solution gas-oil ratio. In the first method the first entry of the table must be at the saturation pressure and one entry will suffice. The R_s value from the first entry will be set as the initial solution GOR. In the second method the solution gas-oil ratio may be specified using the *PB or *PBT keywords. The producing GOR from each well layer is obtained from the input bubble point pressure (from *PB or *PBT input for each block) If the *PB or *PBT keywords are not found then the first method is used.

Typical PVT data curves used by this simulator are given by the solid lines in Figure 7(a), (b), (c), (d), and (e) in Appendix D. Note that they are the curves that would be obtained if there were unlimited gas present (saturated conditions).

The dotted lines in Figure 7(a), (b), and (d) indicate the usual curves obtained when the pressure exceeds the bubble point pressure (PB). The latter curves are assumed to be linear.

The gradient of the Rs curve above the bubble point is equal to zero. IMEX automatically uses a variable substitution technique to model undersaturated conditions (see theoretical outline in Appendix A).

The minimum pressure in the PVT table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of zero would be chosen. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships.

STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$rs = \frac{(\text{volume of dissolved gas at STC})}{(\text{volume of oil at STC})}$$

$$bo = \frac{(\text{volume of oil at reservoir conditions})}{(\text{volume of oil at STC})}$$

$$eg = \frac{(\text{volume of gas at STC})}{(\text{volume of gas at reservoir conditions})}$$

$$bg = \frac{(\text{volume of gas at reservoir conditions})}{(\text{volume of gas at STC})}$$

The default is *EG. Use *BG to enter values for bg instead of eg.

Set numbers are used to assign different PVT tables to grid blocks in the reservoir. If only one PVT region is used, *set_number* may be omitted since the default is 1.

If there is more than one PVT region, the first PVT table 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 *PTYPE keyword.

Example:

```
*PVT 1    ** field units
** p      rs      bo      eg      viso     visg
 14.7    1.0    1.062    6.0    1.04    0.0080
 264.7   90.5   1.15    82.692   0.975   0.0096
 514.7   180.0   1.207   159.388   0.91    0.0112
1014.7   371.0   1.295   312.793   0.83    0.0140
2014.7   636.0   1.435   619.579   0.69    0.0189
2514.7   775.0   1.5      772.798   0.64    0.0208
3014.7   930.0   1.565   925.926   0.594   0.0228
4014.7  1270.0   1.695  1233.046   0.51    0.0268
5014.7  1618.0   1.827  1540.832   0.449   0.0309
9014.7  2984.0   2.36   2590.674   0.203   0.0470
```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the solution gas-oil ratios is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

The acceptable range of values for the formation volume factors is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	0.5	0.5	0.5
max	5.0	5.0	5.0

The acceptable range of values for the gas expansion factors is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	1.0E-20	5.617E-20	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the oil viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	.0095	.0095	.0095
max	86,400.	86,400.	86,400.

The acceptable range of values for the gas viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	9.5E-5	9.5E-5	9.5E-5
max	86,400.	86,400.	86,400.

The acceptable range of values for the surface tension is:

	SI dyne/cm	Field lbf/ft	Lab dyne/cm
min	-1.1E-20	-6.852E-25	-1.1E-20

The acceptable range of values for the oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

A positive total hydrocarbon compressibility check is incorporated in IMEX. This check ensures physically meaningful results when gas comes out of solution. Fundamentally the check ensures that when gas comes out of solution, the resulting oil-gas mixture takes up less volume, or:

$$bg \cdot drs/dp - dbo/dp > 0.0 \text{ must be true,}$$

or in terms of table entries i and $i+1$ we check that both:

$$bg(i) \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)]$$

and

$$bg(i+1) \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)]$$

are greater than or equal to

$$[bo(i+1) - bo(i)] / [p(i+1) - p(i)]$$

A warning is printed if this check is violated. However the run will continue.

Oil and Gas Differential Liberation Table (Optional) *DIFLIB

PURPOSE:

*DIFLIB indicates start of the oil and gas differential liberation table.

TABLE:

*DIFLIB	(*EG *BG *ZG)	<i>set_number</i>					
	*BUBBLE	<i>pressure</i>					
	*BOBF	<i>bobf</i>					
	*RSIF	<i>rsif</i>					
<i>p</i>	<i>rsd</i>	<i>bod</i>	<i>eg</i> <i>bg</i> <i>zg</i>	<i>viso</i>	<i>visg</i>	(<i>co</i>)	(<i>srftn</i>)
:	:	:	:	:	:	:	:

DEFINITIONS:

set_number

Set_number for this particular PVT region. This is the number used with *PTYPE to assign PVT tables to grid blocks.

*EG

Keyword indicating that the gas expansion factor will be used. If all *EG, *BG and *ZG are not present in the data set then this is the default.

*BG

Keyword indicating that the gas formation volume factor will be used instead of the gas expansion factor.

*ZG

Keyword indicating that the gas compressibility factor will be used instead of the gas expansion factor. The reservoir temperature TRES must be specified before the *PVT keyword in order to use this option.

*BUBBLE *pressure*

Indicates the bubble point pressure in the *DIFLIB table used for converting *DIFLIB data to *PVT data (kPa | psi | kPa | kg/cm²).

*BOBF *bobf*

Is the oil formation volume factor of the bubble point oil determined by flashing the oil through the appropriate surface separators and is measured relative to the stock tank oil volume (m³/m³ | RB/STB | cm³/cm³).

*RSIF *rsif*

Is the solution gas-oil-ratio of the bubble- point oil, determined by flashing the oil through the appropriate surface separators, and is measured relative to the oil volume at standard conditions (m³/m³ | scf/STB | cm³/cm³).

<i>p</i>	Pressure (kPa psi kPa kg/cm ²).
<i>rsd</i>	Differential solution gas-oil-ratio (m ³ /m ³ scf/STB cm ³ /cm ³) for saturated oil at pressure <i>p</i> . When the chase gas option is used this is <i>rs</i> of the dissolved chase gas.
<i>bod</i>	Differential formation volume factor (m ³ /m ³ RB/STB cm ³ /cm ³) for saturated oil at pressure <i>p</i> .
<i>eg</i>	Gas expansion factor (m ³ /m ³ scf/RB cm ³ /cm ³) at pressure <i>p</i> . When the chase gas option is used this is the chase gas <i>eg</i> .
<i>bg</i>	Gas formation volume factor (m ³ /m ³ RB/scf cm ³ /cm ³) at pressure <i>p</i> . When the chase gas option is used this is the chase gas <i>bg</i> .
<i>zg</i>	Gas compressibility factor (dimensionless) at pressure <i>p</i> . When the chase gas option is used this is the chase gas <i>zg</i> .
<i>viso</i>	Viscosity (mPa-s cp cp) of saturated oil at pressure <i>p</i> .
<i>visg</i>	Gas viscosity (mPa-s cp cp) at pressure <i>p</i> . When the chase gas option is used this is the chase gas viscosity.
<i>srftn</i>	Gas-oil interfacial tension. It must be monotonically decreasing and greater than zero (dyne/cm).
<i>co</i>	Oil compressibility. It must remain constant or be monotonically increasing. Oil compressibility can alternatively be entered using the *CO or *COT keywords (1/kPa 1/PSI 1/kPa 1/kg/cm ²).

DEFUALTS:

Optional keyword. No default values. One of *PVT, *DIFLIB *PVTAPI *PVTG, *PVTCOND or *PVTVO must be present in the data set.

CONDITIONS:

This keyword must be in the Component Property keyword group.

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure. Surface tension is optional.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired.

When the two phase (oil and water) option is used, there are two ways of specifying the initial solution gas-oil ratio. In the first method the first entry of the table must be at the saturation pressure and one entry will suffice. The Rs value from the first entry will be set as the initial solution GOR. In the second method the solution gas-oil ratio may be specified using the *PB or *PBT keywords. The producing GOR from each well layer is obtained from the input bubble point pressure (from *PB or *PBT input for each block). If the *PB or *PBT keywords are not found then the first method is used.

Typical PVT data curves used by this simulator are given by the solid lines in Figure 7(a), (b), (c), (d), and (e) in Appendix D. Note that they are the curves that would be obtained if there were unlimited gas present (saturated conditions).

The dotted lines in Figure 7(a), (b), and (d) indicate the usual curves obtained when the pressure exceeds the bubble point pressure (PB). The latter curves are assumed to be linear.

The gradient of the Rs curve above the bubble point is equal to zero. IMEX automatically uses a variable substitution technique to model undersaturated conditions (see theoretical outline in Appendix A).

The minimum pressure in the PVT table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of zero would be chosen. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships.

STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$rsd = \frac{\left(\begin{array}{l} \text{volume of dissolved gas at STC determined} \\ \text{by the differential liberation experiment} \end{array} \right)}{\left(\begin{array}{l} \text{volume of residual oil in the} \\ \text{differential liberation experiment} \end{array} \right)}$$

$$bod = \frac{\left(\begin{array}{l} \text{volume of oil at differential liberation conditions} \end{array} \right)}{\left(\begin{array}{l} \text{volume of residual oil in the} \\ \text{differential liberation experiment} \end{array} \right)}$$

$$eg = \frac{\left(\begin{array}{l} \text{volume of gas at STC} \end{array} \right)}{\left(\begin{array}{l} \text{volume of gas at reservoir conditions} \end{array} \right)}$$

$$bg = \frac{\left(\begin{array}{l} \text{volume of gas at reservoir conditions} \end{array} \right)}{\left(\begin{array}{l} \text{volume of gas at STC} \end{array} \right)}$$

The default is *EG. Use *BG to enter values for bg instead of eg.

CONVERSION OF *DIFLIB DATA TO *PVT DATA

Differential liberation data must be converted to PVT data. This done by following the method of L.P. Dake, "Fundamental of Reservoir Engineering", pp. 66-69, First Edition, Elsevier Scientific Publishing Company.

$$\begin{aligned}bo &= bod \times (bobf / bobd) \\rs &= rsif - (rsid - rsd) \times (bobf / bobd)\end{aligned}$$

where:

- bo* = the corrected PVT Bo;
bod = bod entry in the above *DIFLIB table;
bobf = bobf entry using the *BOBF keyword;
bobd = bod entry in the above *DIFLIB table at the pressure specified by the *BUBBLE keyword;
rs = the corrected PVT Rs;
rsif = rsif entry using the *RSIF keyword;
rsid = rsd entry in the above *DIFLIB table at the pressure specified by the *BUBBLE keyword; and
rsd = rsd entry in the above *DIFLIB table.

Set numbers are used to assign different *DIFLIB tables to grid blocks in the reservoir. If only one PVT region is used, the *set_number* may be omitted since the default is 1.

If there is more than one PVT region, the first DIFLIB table 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 *PTYPE keyword.

Example:

```
*DIFLIB 1 ** field units
      *BUBBLE 2014.7
      *RSIF    600.0
      *BOBF    1.35
** p      rsd      bod      eg       viso      visg
 14.7     1.0     1.062    6.0      1.04     0.0080
 264.7    90.5    1.15     82.692   0.975    0.0096
 514.7   180.0    1.207    159.388  0.91     0.0112
1014.7   371.0    1.295    312.793  0.83     0.0140
2014.7   636.0    1.435    619.579  0.69     0.0189
2514.7   775.0    1.5      772.798  0.64     0.0208
3014.7   930.0    1.565    925.926  0.594    0.0228
4014.7  1270.0    1.695   1233.046  0.51     0.0268
5014.7  1618.0    1.827   1540.832  0.449    0.0309
9014.7  2984.0    2.36    2590.674  0.203    0.0470
```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm ²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the solution gas-oil ratios is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

The acceptable range of values for the formation volume factors is:

	SI m³/m³	Field Rb/STB	Lab cm³/cm³
min	0.5	0.5	0.5
max	5.0	5.0	5.0

The acceptable range of values for the gas expansion factors is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	1.0E-20	5.617E-20	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the oil viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	.0095	.0095	.0095
max	86,400.	86,400.	86,400.

The acceptable range of values for the gas viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	9.5E-5	9.5E-5	9.5E-5
max	86,400.	86,400.	86,400.

The acceptable range of values for the surface tension is:

	SI dyne/cm	Field lbf/ft	Lab dyne/cm
min	-1.1E-20	-6.852E-25	-1.1E-20

The acceptable range of values for the oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

A positive total hydrocarbon compressibility check is incorporated in IMEX. This check ensures physically meaningful results when gas comes out of solution. Fundamentally the check ensures that when gas comes out of solution, the resulting oil-gas mixture takes up less volume, or:

$$bg \cdot drs/dp - dbo/dp > 0.0 \text{ must be true,}$$

or in terms of table entries i and $i+1$ we check that both:

$$bg(i) \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)]$$

and

$$bg(i+1) \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)]$$

are greater than or equal to

$$[bo(i+1) - bo(i)] / [p(i+1) - p(i)]$$

A warning is printed if this check is violated. However the run will continue.

Gas PVT Table for API Model (Optional)

*PVTAPI

PURPOSE:

*PVTAPI indicates start of PVT information for a PVT region and also indicates start of gas PVT table. The *APIGRAD Oil PVT Tables must follow the *PVTAPI gas table for each PVT region.

TABLE:

*PVTAPI	(*EG *BG *ZG)	<i>set_number</i>
<i>p</i>	<i>eg</i> <i>bg</i> <i>zg</i>	<i>visg</i>
:	:	:

DEFINITIONS:

set_number

Set_number for this particular PVT region. This is the number used with *PTYPE to assign PVT tables to grid blocks

*EG

Keyword indicating that the gas expansion factor will be used. If all *EG, *BG and *ZG are not present in the data set then this is the default.

*BG

Keyword indicating that the gas formation volume factor will be used instead of the gas expansion factor.

*ZG

Keyword indicating that the gas compressibility factor will be used instead of the gas expansion factor. The reservoir temperature TRES must be specified before the *PVTAPI keyword in order to use this option.

p

Pressure (kPa | psi | kPa | kg/cm²).

eg

Gas expansion factor (m³/m³ | scf/RB | cm³/cm³) at pressure *p*.

bg

Gas formation volume factor (m³/m³ | RB/scf | cm³/cm³) at pressure *p*.

zg

Gas compressibility factor (dimensionless) at pressure *p*.

visg

Gas viscosity (mPa-s | cp | cp) at pressure *p*. When the chase gas option is used this is the chase gas viscosity.

DEFAULTS:

Optional keyword. No default values. *PVTAPI must be present in the data set when the *API-INT model is used.

CONDITIONS:

This keyword must be in the Component Property keyword group.

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired. Typical gas PVT data curves used by this simulator are given by the Figure 7(c) and (e) in Appendix D.

The minimum pressure in the PVT table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of zero would be chosen. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships. STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$eg = \frac{(\text{volume of gas at STC})}{(\text{volume of gas at reservoir conditions})}$$

$$bg = \frac{(\text{volume of gas at reservoir conditions})}{(\text{volume of gas at STC})}$$

The default is *EG. Use *BG to enter values for bg instead of eg.

Set_numbers are used to assign different PVT tables to grid blocks in the reservoir. If only one PVT region is used, the set_number may be omitted since the default is 1.

If there is more than one PVT region, the first PVT table 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 *PTYPE keyword.

Example: (SI units)

```
*PVTAPI 1
** gas table
** p      eg      visg
 100.    1.000   0.020
 1000.   5.000   0.010
 1500.  300.000  0.015
 2000.  600.000  0.019
 2500.  800.000  0.021
 3000. 1000.000  0.023
 3500. 1100.000  0.026
 4000. 1200.000  0.028
 4500. 1500.000  0.031
```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the gas expansion factors is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	1.0E-20	5.617	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the gas viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	9.5E-5	9.5E-5	9.5E-5
max	86,400.	86,400.	86,400.

Oil PVT Tables for API Model (Optional)

*APIGRAD

PURPOSE:

*APIGRAD indicates the start of an oil PVT table. The first *APIGRAD card for a PVT region must follow immediately after the *PVTAPI table for that region.

The *stc_oil_dens* of the first *APIGRAD table in a PVT region is the density of the heavy oil component. The *stc_oil_dens* of the last *APIGRAD table in a PVT region is the density of the light component.

TABLE:

*APIGRAD	<i>stc_oil_dens</i>				
<i>p</i>	<i>rs</i>	<i>bo</i>	<i>viso</i>	<i>co</i>	(<i>srftn</i>)
:	:	:	:	:	:

DEFINITIONS:

stc_oil_dens

oil phase density at STC (kg/m^3 | lbf/ft^3 | g/cm^3) for which the PVT table is valid.

p

Pressure (kPa | psi | kPa | kg/cm^2).

rs

Solution gas-oil ratio (m^3/m^3 | scf/STB | cm^3/cm^3) for saturated oil at pressure *p*.

bo

Formation volume factor (m^3/m^3 | RB/STB | cm^3/cm^3) for saturated oil at pressure *p*.

viso

Viscosity (mPa-s | cp | cp) of saturated oil at pressure *p*.

srftn

Gas-oil interfacial tension. It must be monotonically decreasing and greater than zero (dyne/cm).

co

Oil compressibility. It must remain constant or be monotonically increasing. Oil compressibility may be entered in the *APIGRAD table when using API tracking option or by using the *BOTAPI table. Other entry methods (*CO *COT) are disabled.

DEFAULTS:

Optional keyword. No default values. *APIGRAD must be present in the data set when the *API-INT model is used.

CONDITIONS:

This keyword must be in the Component Property keyword group, following the *PVTAPI keyword. At least two *APIGRAD must be specified for each *PVTAPI keyword. The *APIGRAD tables must be entered in order of decreasing STC oil density (increasing API gravity).

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure. Surface tension is optional.

Even though the flexibility of multiple PVT regions is allowed in the API tracking option, the user should model density changes across connected portions of the reservoir using the API tracking option explicitly (i.e. all APIGRAD tables the same in each PVT region).

The multiple PVT regions should be used to model multiple equilibrium regions or multiple PVT regions which are not connected and so cannot communicate.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired.

A minimum of two *APIGRAD tables have to be specified for each PVT region, although the user can specify as many as he or she wants to better describe properties dependencies with the oil composition. The *stc_oil_dens* for the first table corresponds to STC oil density for the heavy component, while *stc_oil_dens* for the last table corresponds to STC oil density for the light component. When using the API tracking option, the only way to input the oil compressibility, *co*, is through the *APIGRAD keyword.

Also, the user is not allowed to use the *DENSITY *OIL keyword with the API tracking option, the input of STC oil density is only allowed through *APIGRAD, since this quantity can vary within the reservoir according to oil composition.

When using the API tracking option, the oil is assumed to be comprised of two components, a light component and a heavy component.

The volume fraction of light oil at STC is defined as *Vl*, the volume fraction of heavy oil is *Vh* = (1-*Vl*). The density of pure light oil at STC is *denol(STC)*, and the density of pure heavy oil is *denoh(STC)*. The density *denoh(STC)* is the density of the oil mixture at *Vl*=0.0. This is the density (*stc_oil_dens*) associated with the first *APIGRAD table. The last *APIGRAD table must be at defined for pure light oil (*Vl*=1.0), The value of *stc_oil_dens* for the final *APIGRAD table is therefore *denol(STC)*.

The density of the mixture *deno(STC)* is:

$$\text{deno(STC)} = \text{denoh(STC)} * \text{Vh} + \text{denol(STC)} * \text{Vl}$$

The solution gas ratio of the mixture *Rs* is:

$$Rs = Rs(\text{mix}),$$

where *Rs(mix)* is obtained by interpolation of *APIGRAD tables whose oil mixture densities are just lower and just higher than the density defined by the equation for *deno(STC)* above.

If only two APIGRAD tables are used for a PVT region, this is equivalent to:

$$Rs(\text{mix}) = Rsh * Vh + Rsl * Vl$$

where Rsh is the Rs of the heavy component and Rsl is the Rs of the light component.

Oil mixture formation volume factor oil compressibility (co) and viscosity are derived in the same manner.

Using the $*\text{API}$ keyword or the $*\text{APIT}$ keyword light oil volume fraction (at STC) is initialized. Using the $*\text{PVTAPI}$ keyword a gas PVT property table is input for each PVT region (note: separate gas and liquid tables are used in this option).

The $*\text{APIGRAD}$ tables follow the $*\text{PVTAPI}$ gas table for each PVT region. Each APIGRAD table is associated with a Oil mixture density for a PVT region. Oil mixture densities must be entered from heaviest ($Vl=0$) to lightest ($Vl=1.0$). Multiple PVT Regions may be used with this option.

When the two phase (oil and water) option is used, there are two ways of specifying the initial solution gas-oil ratio. In the first method, $*\text{PB}$ is not specified, then the first entry of the table must be at the saturation pressure and one entry will suffice. The Rs value from the first entry will be set as the solution GOR.

In the second method for specifying initial gas-oil ratio in two phase problems (oil and water), the solution gas-oil ratio is specified using the $*\text{PB}$ keyword. The solution gas ratio for producing well layers will be the Rs value of the block the layer is in. $*\text{PB}$ will not vary as it is not being solved for. In addition, the user must ensure the reservoir pressure never drops below $*\text{PB}$, as gas is assumed never to come out of solution.

Care must be taken that the desired method is employed. A $*\text{PB}$ card in the data will override using the RS from the first PVT table entry.

Typical oil PVT data curves used by this simulator are given by the solid lines in Figure 7(a), (b), and (d) in Appendix D. Note that they are the curves that would be obtained if there were unlimited gas present (saturated conditions).

The dotted lines in Figure 7(a), (b), and (d) indicate the usual curves obtained when the pressure exceeds the bubble point pressure (PB). The latter curves are assumed to be linear.

The gradient of the Rs curve above the bubble point is equal to zero. IMEX automatically uses a variable substitution technique to model undersaturated conditions (see theoretical outline in Appendix A).

The minimum pressure in the PVT table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of zero would be chosen. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships.

STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$rs = \frac{(\text{volume of dissolved gas at STC})}{(\text{volume of oil at STC})}$$

$$bo = \frac{(\text{volume of oil at reservoir conditions})}{(\text{volume of oil at STC})}$$

Example: (Field units)

```

*APIGRAD 60.0
** oil pvt table for the heavy component
** p    rs    bo    viso    co
1000.   70.   1.03   160.0  1.0e-5
1500.   110.  1.04   125.0  1.0e-5
2000.   150.  1.05   110.0  1.0e-5
2500.   190.  1.06   100.0  1.0e-5
3000.   230.  1.07   90.0   1.0e-5
3500.   260.  1.08   70.0   1.0e-5
4000.   310.  1.09   60.0   1.0e-5
4500.   350.  1.10   50.0   1.0e-5
5000.   390.  1.11   40.0   1.0e-5
*APIGRAD 45.0
** oil pvt table for the light component
** p    rs    bo    viso    co
1000.   200.  1.10   3.0    5.0e-6
1500.   400.  1.15   2.0    5.0e-6
2000.   500.  1.20   1.0    5.0e-6
2500.   700.  1.25   .90    5.0e-6
3000.   800.  1.30   .85    5.0e-6
3500.   900.  1.35   .80    5.0e-6
4000.   1000. 1.40   .78    5.0e-6
4500.   1200. 1.45   .75    5.0e-6
5000.   1300. 1.50   .70    5.0e-6

```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the solution gas-oil ratios is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

The acceptable range of values for the formation volume factor is:

	SI m³/m³	Field Rb/STB	Lab cm³/cm³
min	0.5	0.5	0.5
max	5.0	5.0	5.0

The acceptable range of values for the oil viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	.0095	.0095	.0095
max	86,400.	86,400.	86,400.

The acceptable range of values for the surface tension is:

	SI dyne/cm	Field lbf/ft	Lab dyne/cm
min	-1.1E-20	-6.852E-25	-1.1E-20

The acceptable range of values for the oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

A positive total hydrocarbon compressibility check is incorporated in IMEX. This check ensures physically meaningful results when gas comes out of solution. Fundamentally the check ensures that when gas comes out of solution, the resulting oil-gas mixture takes up less volume, or:

$$bg \cdot drs/dp - dbo/dp > 0.0 \text{ must be true,}$$

or in terms of table entries i and $i+1$:

$$bg_{int} \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)],$$

is greater than or equal to

$$[bo(i+1) - bo(i)] / [p(i+1) - p(i)]$$

where bg_{int} is obtained from the interpolation of the gas table entered using *PVTAPI keyword at the pressure $p(i)$.

A warning is printed if this check is violated. However the run will continue.

Gas PVT Table for GASWATER Model (Optional) *PVTG

PURPOSE:

*PVTG indicates start of PVT information for a PVT region and also indicates start of gas PVTG table.

TABLE:

*PVTG	(*EG *BG *ZG)	<i>set_number</i>	
<i>p</i>	<i>eg</i> <i>bg</i> <i>zg</i>	<i>visg</i>	(<i>srfn</i>)
:	:	:	:

DEFINITIONS:

set_number

Set_number for this particular PVT region. This is the number used with *PTYPE to assign PVT tables to grid blocks.

*EG

Keyword indicating that the gas expansion factor will be used. If all *EG, *BG and *ZG are not present in the data set then this is the default.

*BG

Keyword indicating that the gas formation volume factor will be used instead of the gas expansion factor.

*ZG

Keyword indicating that the gas compressibility factor will be used instead of the gas expansion factor. The reservoir temperature TRES must be specified before the *PVTG keyword in order to use this option.

Pressure (kPa | psi | kPa | kg/cm²).

eg

Gas expansion factor (m³/m³ | scf/RB | cm³/cm³) at pressure *p*.

bg

Gas formation volume factor (m³/m³ | RB/scf | cm³/cm³) at pressure *p*.

zg

Gas compressibility factor (dimensionless) at pressure *p*.

visg

Gas viscosity (mPa-s | cp | cp) at pressure *p*.

srfn

Water-gas surface tension. It must be monotonically decreasing and greater than zero (dyne/cm).

DEFAULTS:

Optional keyword. No default values. *PVTG must be present in the data set when the *GASWATER model is used.

CONDITIONS:

This keyword must be in the Component Property keyword group.

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired. Typical gas PVT data curves used by this simulator are given by the Figure 7(c) and (e) in Appendix D.

The minimum pressure in the PVT table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of zero would be chosen. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships. STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$eg = \frac{(\text{volume of gas at STC})}{(\text{volume of gas at reservoir conditions})}$$
$$bg = \frac{(\text{volume of gas at reservoir conditions})}{(\text{volume of gas at STC})}$$

The default is *EG. Use *BG to enter values for bg instead of eg.

Set_numbers are used to assign different PVT tables to grid blocks in the reservoir. If only one PVT region is used, the set_number may be omitted since the default is 1.

If there is more than one PVT region, the first PVT table 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 *PTYPE keyword.

Example: (SI units)

```
*PVTG 1
** gas table
** p      eg      visg
 100.    1.000   0.020
 1000.   5.000   0.010
 1500.  300.000  0.015
 2000.  600.000  0.019
 2500.  800.000  0.021
 3000. 1000.000  0.023
 3500. 1100.000  0.026
 4000. 1200.000  0.028
 4500. 1500.000  0.031
```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the gas expansion factors is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	1.0E-20	5.617	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the gas viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	9.5E-5	9.5E-5	9.5E-5
max	86,400.	86,400.	86,400.

Condensate - Oil and Gas PVT Table (Optional)

*PVTCOND

PURPOSE:

*PVTCOND indicates start of the oil and gas PVTCOND table.

TABLE:

*PVTCOND (*EG *BG *ZG) <i>set_number</i>							
<i>p</i>	<i>rs</i>	<i>rv</i>	<i>bo</i>	<i>eg bg zg</i>	<i>viso</i>	<i>visg</i>	(<i>srfn</i>)
:	:	:	:	:	:	:	:

NOTE:

We will define the following terms:

1. Black Oil - Liquid at surface, Liquid at reservoir conditions
2. Condensate - Liquid at surface, Gas at reservoir conditions
3. Free Gas - Gas at surface, Gas at reservoir conditions
4. Solution Gas - Gas at surface, Liquid at reservoir conditions

The Gas phase in the reservoir is made up of Condensate and Free Gas. The Oil phase in the reservoir is made up of Black Oil and Solution Gas. The Gas phase at the surface is made up of Free Gas and Solution Gas. The Oil phase at the surface is made up of Black Oil and Condensate. Thus Oil Gas ratio would be (Condensate + Black Oil) / (Free gas + Sol'n Gas). By default IMEX will output Oil and Gas based production. Condensate, Black Oil, Sol'n Gas and Free Gas production splits may be obtained by using the *PSPLIT keyword in the IO Control section.

DEFINITIONS:

set_number

Set_number for this particular PVT region. This is the number used with *PTYPE to assign PVT tables to grid blocks.

*EG

Keyword indicating that the gas expansion factor will be used. If *EG, *BG and *ZG are not present in the data set then this is the default.

*BG

Keyword indicating that the gas formation volume factor will be used instead of the gas expansion factor.

*ZG

Keyword indicating that the gas compressibility factor will be used instead of the gas expansion factor. The reservoir temperature TRES must be specified before the *PVTCOND keyword in order to use this option.

<i>p</i>	Pressure (kPa psi kPa kg/cm ²).
<i>rs</i>	Solution gas-oil ratio (m ³ /m ³ scf/STB cm ³ /cm ³) for saturated oil at pressure <i>p</i> .
<i>rv</i>	Oil Content (m ³ /m ³ STB/scf cm ³ /cm ³) for condensate saturated gas at pressure <i>p</i> .
<i>bo</i>	Formation volume factor (m ³ /m ³ RB/STB cm ³ /cm ³) for saturated oil at pressure <i>p</i> .
<i>eg</i>	Gas expansion factor (m ³ /m ³ scf/RB cm ³ /cm ³) for condensate saturated gas at pressure <i>p</i> .
<i>bg</i>	Gas formation volume factor (m ³ /m ³ RB/scf cm ³ /cm ³) for condensate saturated gas at pressure <i>p</i> .
<i>zg</i>	Gas compressibility factor (dimensionless) for condensate saturated gas at pressure <i>p</i> .
<i>viso</i>	Viscosity (mPa-s cp cp) of saturated oil at pressure <i>p</i> .
<i>visg</i>	Gas viscosity (mPa-s cp cp) for condensate saturated gas at pressure <i>p</i> .
<i>srfn</i>	Gas-liquid interfacial tension. It must be monotonically decreasing and greater than zero (dyne/cm).

DEFUALTS:

Optional keyword. No default values. . One of *PVT, *DIFLIB, *PVTAPI, *PVTG, *PVTCOND or *PVTVO must be present in the data set.

CONDITIONS:

This keyword must be in the Component Property keyword group.

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure. Surface tension is optional.

Interpolation of gas and oil properties in PVTCOND tables occurs between the first table value and ten times the last table value (extrapolated). During dry gas injection it is important that the dry gas pressure (where $R_v = 0.0$) is included in the table. In addition the largest table entry should be large enough to avoid a large amount of extrapolation (i.e. ten times the maximum entry).

When a detailed EOS representation is used to generate PVT properties for condensate (such as the table below) it is possible that many properties such as R_s , R_v , B_o or Gas viscosity may no longer be monotonic functions of pressure. In most cases this does not present a problem to IMEX, but under some circumstances the existence of minima or maxima in PVT can cause convergence difficulties.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired.

The minimum pressure in the PVTCOND table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of atmospheric would be chosen as this would represent a dry gas value. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships.

STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$r_s = \frac{(\text{volume of dissolved gas at STC})}{(\text{volume of oil at STC})}$$

$$r_v = \frac{(\text{volume of condensate at STC})}{(\text{volume of gas at STC})}$$

$$b_o = \frac{(\text{volume of oil at reservoir conditions})}{(\text{volume of oil at STC})}$$

$$e_g = \frac{(\text{volume of gas at STC})}{(\text{volume of gas at reservoir conditions})}$$

$$b_g = \frac{(\text{volume of gas at reservoir conditions})}{(\text{volume of gas at STC})}$$

The default is *EG. Use *BG to enter values for bg instead of eg.

Set_numbers are used to assign different PVTCOND tables to grid blocks in the reservoir. If only one PVT region is used, the set_number may be omitted since the default is 1.

If there is more than one PVT region, the first PVTCOND table 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 *PTYPE keyword.

Example:

*MODEL *GASWATER_WITH_CONDENSATE

```

*PVTCOND *BG
** pressure solution solution      oil    gas        oil    gas
**     psia       GOR      CGR      FVF    FVF      vis,cp   vis,cp
** -----
14.70          0.0    0.0           1.10   0.20534    0.24    0.012
515.00         288.7  0.025060e-3  1.2644  5.86124e-3  0.19215  0.01240
1015.00        618.7  0.021148e-3  1.4766  2.72302e-3  0.14588  0.01394
1515.00        980.1  0.026445e-3  1.6987  1.71098e-3  0.11643  0.01652
2015.00        1377.7 0.039146e-3  1.9345  1.23923e-3  0.09735  0.02097
2515.00        1781.6  0.062337e-3  2.1601  1.00106e-3  0.08666  0.02779
3015.00        1996.6  0.100128e-3  2.2398  0.89373e-3  0.08849  0.03686
3265.00        1854.7  0.122098e-3  2.1129  0.87063e-3  0.10269  0.04188
3515.00        1501.4  0.138469e-3  1.8633  0.85536e-3  0.14397  0.04624
3765.00        1155.2  0.146526e-3  1.6384  0.84058e-3  0.22699  0.04948
4006.47        912.6  0.149747e-3  1.4891  0.82634e-3  0.34108  0.05189

```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the solution gas-oil ratios is:

	SI m ³ /m ³	Field scf/STB	Lab cm ³ /cm ³
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

The acceptable range of values for the oil content is:

	SI m ³ /m ³	Field STB/scf	Lab cm ³ /cm ³
min	0.0	0.0	0.0
max	1.0E-1	0.0178	1.0E-1

The acceptable range of values for the formation volume factors is:

	SI m^3/m^3	Field Rb/STB	Lab cm^3/cm^3
min	0.5	0.5	0.5
max	5.0	5.0	5.0

The acceptable range of values for the gas expansion factors is:

	SI m^3/m^3	Field scf/STB	Lab cm^3/cm^3
min	1.0E-20	5.617E-20	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the oil viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	.0095	.0095	.0095
max	86,400.	86,400.	86,400.

The acceptable range of values for the gas viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	9.5E-5	9.5E-5	9.5E-5
max	86,400.	86,400.	86,400.

The acceptable range of values for the surface tension is:

	SI dyne/cm	Field lbf/ft	Lab dyne/cm
min	-1.1E-20	-6.852E-25	-1.1E-20

The acceptable range of values for the oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

A positive total hydrocarbon compressibility check is incorporated in IMEX for the condensate model. Its derivation by K.H. Coats can be found in “A Note on IMPES and some IMPES-based Simulation Models” SPEJ Journal 5 (3), Sept 2000.

We check that:

$$rs \cdot rv < 1$$

$$bg > rv \cdot bo$$

$$bo > rs \cdot bg$$

and

$$(bg - rv \cdot bo) \frac{drs}{dp} > (1 - rs \cdot rv) \cdot \frac{dbo}{dp}$$

or in terms of table entries i and $i+1$ we check that:

$$rs(i) \cdot rv(i) < 1$$

$$bg(i) > rv(i) \cdot bo(i)$$

$$bo(i) > rs(i) \cdot bg(i)$$

$$[bg(i) - rv(i) \cdot bo(i)] \cdot [rs(i+1) - rs(i)] / (p(i+1) - p(i)) >$$

$$[1 - rs(i) \cdot rv(i)] \cdot [bo(i+1) - bo(i)] / (p(i+1) - p(i))$$

and

$$[bg(i+1) - rv(i+1) \cdot bo(i+1)] \cdot [rs(i+1) - rs(i)] / (p(i+1) - p(i)) >$$

$$[1 - rs(i+1) \cdot rv(i+1)] \cdot [bo(i+1) - bo(i)] / (p(i+1) - p(i))$$

A warning is printed if this check is violated. However the run will continue.

Volatile - Oil and Gas PVT Table (Optional)

*PVTVO

PURPOSE:

*PVTVO indicates start of the oil and gas PVTVO table.

TABLE:

*PVTVO (*EG *BG *ZG) set_number									
p	rs	rv	bo	eg bg zg	viso	visg	egd bgd zgd	visgd	(srftn)
:	:	:	:	:	:	:	:	:	:

NOTE:

We will define the following terms:

1. Black Oil - Liquid at surface, Liquid at reservoir conditions
2. Condensate - Liquid at surface, Gas at reservoir conditions
3. Free Gas - Gas at surface, Gas at reservoir conditions
4. Solution Gas - Gas at surface, Liquid at reservoir conditions

The Gas phase in the reservoir is made up of Condensate and Free Gas. The Oil phase in the reservoir is made up of Black Oil and Solution Gas. The Gas phase at the surface is made up of Free Gas and Solution Gas. The Oil phase at the surface is made up of Black Oil and Condensate. Thus Oil Gas ratio would be (Condensate + Black Oil) / (Free gas + Sol'n Gas). By default IMEX will output Oil and Gas based production. Condensate, Black Oil, Sol'n Gas and Free Gas production splits may be obtained by using the *PSPLIT keyword in the IO Control section.

DEFINITIONS:

set_number

Set_number for this particular PVT region. This is the number used with *PTYPE to assign PVT tables to grid blocks.

*EG

Keyword indicating that the gas expansion factor will be used. If *EG, *BG and *ZG are not present in the data set then this is the default.

*BG

Keyword indicating that the gas formation volume factor will be used instead of the gas expansion factor.

*ZG

Keyword indicating that the gas compressibility factor will be used instead of the gas expansion factor. The reservoir temperature TRES must be specified before the *PVTVO keyword in order to use this option.

p

Pressure (kPa | psi | kPa | kg/cm²).

<i>rs</i>	Solution gas-oil ratio (m^3/m^3 scf/STB cm^3/cm^3) for saturated oil at pressure p .
<i>rv</i>	Oil Content (m^3/m^3 STB/scf cm^3/cm^3) for condensate saturated gas at pressure p .
<i>bo</i>	Formation volume factor (m^3/m^3 RB/STB cm^3/cm^3) for saturated oil at pressure p .
<i>eg</i>	Gas expansion factor (m^3/m^3 scf/RB cm^3/cm^3) for condensate saturated gas at pressure p .
<i>bg</i>	Gas formation volume factor (m^3/m^3 RB/scf cm^3/cm^3) for condensate saturated gas at pressure p .
<i>zg</i>	Gas compressibility factor (dimensionless) for condensate saturated gas at pressure p .
<i>viso</i>	Viscosity ($\text{mPa}\cdot\text{s}$ cp cp) of saturated oil at pressure p .
<i>visg</i>	Gas viscosity ($\text{mPa}\cdot\text{s}$ cp cp) for condensate saturated gas at pressure p .
<i>egd</i>	Gas expansion factor (m^3/m^3 scf/RB cm^3/cm^3) for dry gas at pressure p .
<i>bgd</i>	Gas formation volume factor (m^3/m^3 RB/scf cm^3/cm^3) for dry gas at pressure p .
<i>zgd</i>	Gas compressibility factor (dimensionless) for dry gas at pressure p .
<i>visgd</i>	Gas viscosity ($\text{mPa}\cdot\text{s}$ cp cp) for dry gas at pressure p .
<i>srfn</i>	Gas-liquid interfacial tension. It must be monotonically decreasing and greater than zero (dyne/cm).

DEFAULTS:

Optional keyword. No default values. One of *PVT, *DIFLIB, *PVTAPI, *PVTG, *PVTCOND or *PVTVO must be present in the data set.

CONDITIONS:

This keyword must be in the Component Property keyword group.

Table entries must be in order of increasing pressure and should extend to above the initial bubble point pressure. Surface tension is optional.

During dry gas injection it is important that the dry gas pressure (where $Rv = 0.0$) is included in the table. When a detailed EOS representation is used to generate PVT properties for condensate (such as the table below) it is possible that many properties such as Rs , Rv , Bo or Gas viscosity may no longer be monotonic functions of pressure. In most cases this does not present a problem to IMEX, but under some circumstances the existence of minima or maxima in PVT can cause convergence difficulties.

EXPLANATION:

In general, this table must contain at least two entries, and may contain more entries if desired.

The minimum pressure in the PVTVO table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of atmospheric would be chosen as this would represent a dry gas value. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section.

The formation volume factor and viscosity of water are calculated from functional relationships.

STC means stock tank conditions. (14.7 psia and 60 F | 101.3 kPa and 288.15 K | 1.03 kg/cm² and 288.15 K)

$$rs = \frac{(\text{volume of dissolved gas at STC})}{(\text{volume of oil at STC})}$$

$$rv = \frac{(\text{volume of condensate at STC})}{(\text{volume of gas at STC})}$$

$$bo = \frac{(\text{volume of oil at reservoir conditions})}{(\text{volume of oil at STC})}$$

$$eg = \frac{(\text{volume of gas at STC})}{(\text{volume of gas at reservoir conditions})}$$

$$bg = \frac{(\text{volume of gas at reservoir conditions})}{(\text{volume of gas at STC})}$$

The default is *EG. Use *BG to enter values for bg instead of eg .

Set_numbers are used to assign different PVTVO tables to grid blocks in the reservoir. If only one PVT region is used, the *set_number* may be omitted since the default is 1.

If there is more than one PVT region, the first PVTVO table 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 *PTYPE keyword.

The *PVTVO table includes dry gas as well as condensate saturated gas properties. At pressures other than the dew point pressure, gas properties are obtained by interpolating between saturated and dry conditions. Thus *EGUST and *VGUST should not be used with the *PVTVO option as they are not required (*EGUST and *VGUST also define gas properties at pressures other than the dew point pressure).

Example:

```
*MODEL *VOLATILE_OIL
```

*PVTVO	*BG									
**pressure	solution	solution	oil	gas	oil	gas	dry gas	dry gas		
** kpa	GOR	CGR	FVF	FVF	vis, cp	vis, cp	FVF	vis, cp		
<hr/>										
101.32	0.0	0.00000	1.0	1.35118	0.180	0.01300	1.35118	0.01300		
4926.30	21.8867	0.000027	1.1436	0.02779	0.17945	0.01470	0.02777	0.01473		
8372.75	41.5325	0.000033	1.2175	0.01597	0.16769	0.01553	0.01598	0.01553		
14576.30	83.2489	0.000059	1.3447	0.00899	0.14546	0.01773	0.00902	0.01756		
21469.18	143.3570	0.000119	1.5225	0.00617	0.12204	0.02159	0.00620	0.02068		
29740.60	254.5648	0.000264	1.8504	0.00472	0.09578	0.02856	0.00467	0.02507		
38012.04	530.0487	0.000656	2.7030	0.00420	0.06721	0.04125	0.00387	0.02973		
39816.28	801.7297	0.001026	3.6174	0.004144	0.06200	0.05026	0.00374	0.03097		

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the solution gas-oil ratios is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

The acceptable range of values for the oil content is:

	SI m³/m³	Field STB/scf	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E-1	0.0178	1.0E-1

The acceptable range of values for the formation volume factors is:

	SI m³/m³	Field Rb/STB	Lab cm³/cm³
min	0.5	0.5	0.5
max	5.0	5.0	5.0

The acceptable range of values for the gas expansion factors is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	1.0E-20	5.617E-20	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the oil viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	.0095	.0095	.0095
max	86,400.	86,400.	86,400.

The acceptable range of values for the gas viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	9.5E-5	9.5E-5	9.5E-5
max	86,400.	86,400.	86,400.

The acceptable range of values for the surface tension is:

	SI dyne/cm	Field lbf/ft	Lab dyne/cm
min	-1.1E-20	-6.852E-25	-1.1E-20

The acceptable range of values for the oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

A positive total hydrocarbon compressibility check is incorporated in IMEX for the condensate model. Its derivation by K.H. Coats can be found in “A Note on IMPES and some IMPES-based Simulation Models” SPEJ Journal 5 (3), Sept 2000.

We check that:

$$rs \cdot rv < 1$$

$$bg > rv \cdot bo$$

$$bo > rs \cdot bg$$

and

$$(bg - rv \cdot bo) \frac{d(rs)}{dp} > (1 - rs \cdot rv) \cdot \frac{dbo}{dp}$$

or in terms of table entries i and $i+1$ we check that:

$$rs(i) \cdot rv(i) < 1$$

$$bg(i) > rv(i) \cdot bo(i)$$

$$bo(i) > rs(i) \cdot bg(i)$$

$$[bg(i) - rv(i) \cdot bo(i)] \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)] >$$

$$[1 - rs(i) \cdot rv(i)] \cdot [bo(i+1) - bo(i)] / [p(i+1) - p(i)]$$

and

$$[bg(i+1) - rv(i+1) \cdot bo(i+1)] \cdot [rs(i+1) - rs(i)] / [p(i+1) - p(i)] >$$

$$[1 - rs(i+1) \cdot rv(i+1)] \cdot [bo(i+1) - bo(i)] / [p(i+1) - p(i)]$$

A warning is printed if this check is violated. However the run will continue.

Oil Compressibility as a Function of Pressure (Optional)

***COT**

PURPOSE:

*COT indicates the input of an oil compressibility table that is a function of both pressure (P) and bubble point pressure (P_b).

FORMAT:

***COT set_number**

<i>p</i>	<i>co</i>
:	:

DEFINITION:

set_number

PVT region that corresponds to the *co* table.

p

Pressure above the corresponding bubble point pressure in the PVT table. (kPa | psi | kPa | kg/cm²). The first entry of this table is the oil compressibility value at the bubble point pressure. All subsequent pressure entries must be greater than this bubble point pressure and must increase in a monotonic fashion.

co

Oil Compressibility entry at the corresponding pressure. (1/kPa | 1/psi | 1/kPa | 1/(kg/cm²)).

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be in the Component Properties section. *COT may be used when the oil compressibility is a function of both pressure and bubble point pressure. This corresponds to an oil formation volume factor (Bo) which has a nonlinear functional dependence on pressures above the bubble point pressure. The *COT keyword cannot be used with the *CO keyword, the *BOT keyword, or when *co* is entered in the PVT table. The *COT keyword cannot be used with the *GASWATER option (The oil phase is not modelled). The *COT keyword cannot be used with the *GASWATER_WITH_CONDENSATE option (The oil phase is always saturated with gas).

EXPLANATION:

The oil compressibility *co*, gives the slope of the *bo* curve divided by $bo(P_b)$ when the oil is above the bubble point pressure. For a typical black-oil system, *co* assumes a constant value for $P > P_b$. A weak compositional dependence can be exhibited by allowing *co* to vary with P_b . *co* is further allowed to vary in a nonlinear manner with P ($P > P_b$). If *co* is taken as constant, then

$$bo(P) = bo(Pb) \cdot [1 - co \cdot (P - Pb)]$$

If co is a function of bubble point pressure only, then a different value of co must be used for each bubble point pressure. If co is also a function of pressure, then co is integrated over the range of pressures between Pb and P and used in the equation above. This integration is performed numerically within IMEX.

Because of the integration step, it is not trivial to calculate co 's obtained from measured values of Bo above the bubble point into IMEX by hand. However, ModelBuilder can easily perform the calculations required to convert measured Bo above the bubble point into co above the bubble point for *COT table input.

Many *COT keywords, with each *COT keyword followed two columns of numbers (p vs. co entries), may appear in the Component Properties Data section. If more than one PVT region is specified, then each *COT keyword must be followed by an integer specifying the PVT region that the p vs. co table represents. If the number of PVT regions is only one, then it is not necessary to specify the *set_number*.

The minimum number of entries specified for each table is one (co at the bubble point). If the calculated block pressure is outside the range of data specified, then the last table entry value of co is used for the calculated block pressure.

The minimum number of *COT keywords required per PVT region is one. If more than one *COT keyword is defined per PVT region, then the bubble point pressure (the first pressure entry in the table) must be greater than the previous table's value and less than the next table's value. If more than one table is specified, then a linear interpolation is performed to determine the variation of co with p for the calculated block bubble point pressure. If the calculated block bubble point pressure falls outside the tables range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example:

```
*PVT 1    ** field units
** p        rs      bo       eq      viso     visq
  14.7     1.0    1.062    6.0     1.04    0.0080
 264.7    90.5   1.15    82.692   0.975   0.0096
 514.7   180.0   1.207   159.388   0.91    0.0112
1014.7   371.0   1.295   312.793   0.83    0.0140
2014.7   636.0   1.435   619.579   0.69    0.0189
2514.7   775.0   1.5      772.798   0.64    0.0208
3014.7   930.0   1.565   925.926   0.594   0.0228
4014.7  1270.0   1.695  1233.046   0.51    0.0268
5014.7  1618.0   1.827  1540.832   0.449   0.0309
9014.7  2984.0   2.36   2590.674   0.203   0.0470

*COT 1
** p          co
 14.7    2.00e-06
1000.0   1.90e-06
2000.0   1.75e-06
5000.0   1.70e-06
9000.0   1.68e-06
```

```

*COT 1
** p      co
3014.7  1.80e-06
4000.0  1.75e-06
5000.0  1.73e-06
9000.0  1.70e-06

*COT 1
** p      co
7000.0  1.10e-06
8000.0  1.09e-06
9000.0  1.09e-06

```

The acceptable range of values for oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

Bo Above the Bubble Point as a Function of Pressure (Optional)

*BOT

PURPOSE:

*BOT indicates the input of an Bo table that is a function of both pressure (P) and bubble point pressure (P_b).

FORMAT:

*BOT *set_number*

p *bo*
: :
:

DEFINITION:

set_number

PVT region that corresponds to the *bo* table.

p

Pressure above the corresponding bubble point pressure in the PVT table. (kPa | psi | kPa | kg/cm²). The first entry of this table is the oil formation volume factor value at the bubble point pressure. All subsequent pressure entries must be greater than this bubble point pressure and must increase in a monotonic fashion.

bo

Oil formation volume factor entry at the corresponding pressure. (m³/m³ | RB/STB | cm³/cm³).

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be in the Component Properties section. *BOT may be entered when the oil formation volume factor is a function of both pressure and bubble point pressure. This corresponds to an oil formation volume factor (*bo*) which has a nonlinear functional dependence on pressures above the bubble point pressure. The *BOT keyword cannot be used with the *COT keyword, *CO keyword or when *co* is entered in the PVT table. The *BOT keyword cannot be used with the *GASWATER option (The oil phase is not modelled). The *BOT keyword cannot be used with the *GASWATER_WITH_CONDENSATE option (The oil phase is always saturated with gas).

EXPLANATION:

The oil formation volume factor can exhibit a weak compositional dependence when the oil is above the bubble point. The *BOT table allows us to model this dependence varying with both bubble point pressure and block pressure.

Many *BOT keywords, with each *BOT keyword followed two columns of numbers (*p* vs. *bo* entries), may appear in the Component Properties Data section. If more than one PVT region is specified, then each *BOT keyword must be followed by an integer specifying the PVT region that the *p* vs. *bo* table represents. If the number of PVT regions is only one, then it is not necessary to specify the set_number.

The minimum number of entries specified for each table is two (*bo* at the bubble point and *bo* at a pressure above the bubble point). If the calculated block pressure is outside the range of data specified, then the last table entry value of *bo* is used for the calculated block pressure.

The minimum number of *BOT keywords per PVT region required is one. If more than one *BOT keyword is defined per PVT region, then the bubble point pressure (the first pressure entry in the table) must be greater than the previous table's value and less than the next table's value. If more than one table is specified, then a linear interpolation is performed to determine the variation of *bo* with *p* for the calculated block bubble point pressure. If the calculated block bubble point pressure falls outside the tables range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example:

```

*PVT 1    ** field units
** p        rs      bo       eg      viso     visg
  14.7      1.0    1.062    6.0     1.04    0.0080
 264.7     90.5   1.15    82.692   0.975   0.0096
 514.7    180.0   1.207   159.388   0.91    0.0112
1014.7    371.0   1.295   312.793   0.83    0.0140
2014.7    636.0   1.435   619.579   0.69    0.0189
2514.7    775.0   1.5      772.798   0.64    0.0208
3014.7    930.0   1.565   925.926   0.594   0.0228
4014.7   1270.0   1.695  1233.046   0.51    0.0268
5014.7   1618.0   1.827  1540.832   0.449   0.0309
9014.7   2984.0   2.36   2590.674   0.203   0.0470

*BOT 1
** p        bo
  14.7      1.062
 1000.0    1.051
 2000.0    1.041
 5000.0    1.009
 9000.0    0.966

*BOT 1
** p        bo
 3014.7    1.565
 4000.0    1.549
 5000.0    1.534
 9000.0    1.471

*BOT 1
** p        bo
 7000.0    2.000
 8000.0    1.980
 9000.0    1.960

```

Oil Viscosity Above the Bubble Point as a Function of Pressure (Optional)

***VOT**

PURPOSE:

*VOT indicates the input of an oil viscosity (Vo) table that is a function of both pressure (P) and bubble point pressure (P_b).

FORMAT:

*VOT *set_number*

p *Vo*

 : :

DEFINITION:

set_number

PVT region that corresponds to the *bo* table.

p

Pressure above the corresponding bubble point pressure in the PVT table. (kPa | psi | kPa | kg/cm²). The first entry of this table is the oil viscosity value at the bubble point pressure. All subsequent pressure entries must be greater than this bubble point pressure and must increase in a monotonic fashion.

Vo

Oil viscosity entry at the corresponding pressure. (mPa-s | cp | cp | mPa-s).

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be in the Component Properties section. *VOT may be entered when the oil viscosity is a function of both pressure and bubble point pressure. This corresponds to an oil viscosity (Vo) which has a nonlinear functional dependence on pressures above the bubble point pressure. The *VOT keyword cannot be used with the *CVO keyword. The *VOT keyword cannot be used with the *GASWATER option (The oil phase is not modelled). The *VOT keyword cannot be used with the *GASWATER_WITH_CONDENSATE option (The oil phase is always saturated with gas).

EXPLANATION:

The oil viscosity can exhibit a weak compositional dependence when the oil is above the bubble point. The *VOT table allows us to model this dependence varying with both bubble point pressure and block pressure.

Many *VOT keywords, with each *VOT keyword followed two columns of numbers (p vs. Vo entries), may appear in the Component Properties Data section. If more than one PVT region is specified, then each *VOT keyword must be followed by an integer specifying the

PVT region that the p vs. Vo table represents. If the number of PVT regions is only one, then it is not necessary to specify the *set_number*.

The minimum number of entries specified for each table is two (oil viscosity at the bubble point and oil viscosity at a pressure above the bubble point). If the calculated block pressure is outside the range of data specified, then the last table entry value of Vo is used for the calculated block pressure.

The minimum number of *VOT keywords required per PVT region is one. If more than one *VOT keyword is defined per PVT region, then the bubble point pressure (the first pressure entry in the table) must be greater than the previous table's value and less than the next table's value. If more than one table is specified, then a linear interpolation is performed to determine the variation of Vo with p for the calculated block bubble point pressure. If the calculated block bubble point pressure falls outside the tables range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example:

```
*PVT 1    ** field units
** p      rs     bo      eg      viso    visg
 14.7     1.0    1.062   6.0     1.04    0.0080
 264.7    90.5   1.15    82.692  0.975   0.0096
 514.7    180.0  1.207   159.388 0.91    0.0112
1014.7    371.0  1.295   312.793 0.83    0.0140
2014.7    636.0  1.435   619.579 0.69    0.0189
2514.7    775.0  1.5     772.798 0.64    0.0208
3014.7    930.0  1.565   925.926 0.594   0.0228
4014.7    1270.0 1.695   1233.046 0.51    0.0268
5014.7    1618.0 1.827   1540.832 0.449   0.0309
9014.7    2984.0 2.36    2590.674 0.203   0.0470
*VOT 1
** p      vo
 14.7     1.040
 1000.0   1.064
 2000.0   1.088
 5000.0   1.161
 9000.0   1.259
*VOT 1
** p      vo
 3014.7   0.594
 4000.0   0.618
 5000.0   0.642
 9000.0   0.764
*VOT 1
** p      co
 7000.0   0.326
 8000.0   0.350
 9000.0   0.375
```

The acceptable range of values for oil viscosity is:

	SI mPa-s	Field cp	Lab mPa-s	Mod. SI mPa-s
min	0.095	0.095	0.095	0.095
max	8.64E+4	8.64E+4	8.64E+4	8.64E+4

Bo above the Bubble Point as a Function of Pressure and STC Oil Density for API Tracking (Optional)

*BOTAPI

PURPOSE:

*BOTAPI indicates the input of a Bo table that is a function of pressure (P), bubble point pressure (P_b) and STC oil density in an API tracking model.

FORMAT:

```
*BOTAPI set_number apigrad_number
      p      bo
      :      :
```

DEFINITION:

set_number

PVT region that corresponds to the botapi table.

apigrad_number

APIGRAD table number for the set number.

p

Pressure above the corresponding bubble point pressure in the APIGRAD table. (kPa | psi | kPa | kg/cm²). The first entry of this table is the oil formation volume factor value at the bubble point pressure. All subsequent pressure entries must be greater than this bubble point pressure and must increase in a monotonic fashion.

bo

Oil formation volume factor entry at the corresponding pressure. (m³/m³ | RB/STB | cm³/cm³).

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be in the Component Properties section and is only used for API tracking problems. *BOTAPI may be entered when the oil formation volume factor is a function of pressure, bubble point pressure and API. This corresponds to an oil formation volume factor (Bo) which has a nonlinear functional dependence on pressures above the bubble point pressure and which can vary depending on the STC density. The *BOTAPI keyword cannot be used when *co* is entered directly in the APIGRAD table. When inputting a BOTAPI Bo table *apigrad_number* should vary fastest. All APIGRAD table numbers should be read in for the first PVT region before reading all APIGRAD table numbers for the second and subsequent PVT regions. BOTAPI tables for all APIGRAD tables and all PVT regions need to read in if this option is used.

EXPLANATION:

The oil formation volume factor can exhibit a weak compositional dependence when the oil is above the bubble point. The *BOTAPI table allows us to model this dependence varying with bubble point pressure, block pressure, and STC oil density.

Many *BOTAPI keywords, with each *BOTAPI keyword followed by two columns of numbers (p vs. bo entries), may appear in the Component Properties Data section. Each *BOTAPI keyword must be followed by an integer (*set_number*) specifying the PVT region that the p vs. bo table represents.

Following the *set_number*, an *apigrad_number* must appear, in API tracking models each PVT region must have at least two APIGRAD tables, the *apigrad* number associates the BOTAPI table to a particular (1st, 2nd, 3rd etc) APIGRAD table for each PVT region.

Two or more BOTAPI tables with identical values of *set_number* and *apigrad_number* refer to undersaturated bo in the same APIGRAD table but at different bubble points (saturation pressures).

All APIGRAD tables must be given BOTAPI tables.

The minimum number of entries specified for each table is two (bo at the bubble point and bo at a pressure above the bubble point). If the calculated block pressure is outside the range of data specified, then the last table entry value of bo is used for the calculated block pressure.

The minimum number of required *BOTAPI keywords per apigrad table number (for each PVT region) is one. If more than one *BOTAPI keyword is defined per apigrad table number, then the bubble point pressure (the first pressure entry in the table) must be greater than the previous table's value and less than the next table's value. If more than one table is specified, then a linear interpolation is performed to determine the variation of bo with p for the calculated block bubble point pressure in a simulation block.

The BOTAPI calculation performs this calculation in a BOTAPI table with an STC oil density less than the block value and in a BOTAPI table with an STC oil density greater than the simulation block value.

Finally the BOTAPI table calculation interpolates with respect to the block STC density to determine the bo of the oil at the specific STC oil density of the simulation block.

If the calculated block bubble point pressure falls outside the tables range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example:

```

*APIGRAD  56.4239 ** 25 api
** p      rs       bo     viso
 14.70    0.0000   1.0346   2.3982
 174.00   24.3247  1.0509   2.3709
 365.00   52.4846  1.0660   2.3297
 615.00   85.6892  1.0828   2.2562
 865.00  118.2507  1.0986   2.1701
1115.00  150.9454  1.1142   2.0771
1365.00  184.0502  1.1297   1.9812
1452.23  195.6845  1.1352   1.9478
1929.61  261.0615  1.1653   1.7661
2495.91  342.6808  1.2024   1.5622
3182.53  447.5103  1.2491   1.3433
4035.30  587.1534  1.3101   1.1170
5123.39  782.4795  1.3934   0.8916

*APIGRAD  53.6858 ** 33 API
** p      rs       bo     viso
 14.70    0.0000   1.0425   1.7622
 174.00   38.8883  1.0724   1.5043
 365.00   87.3174  1.1011   1.3908
 615.00  139.5046  1.1294   1.2735
 865.00  189.5810  1.1555   1.1709
1115.00  239.8258  1.1811   1.0777
1365.00  291.0484  1.2068   0.9924
1615.00  343.6380  1.2329   0.9142
1865.00  397.8353  1.2595   0.8427
2115.00  453.8270  1.2869   0.7774
2365.00  511.7849  1.3150   0.7178
2534.22  552.2113  1.3345   0.6805
2994.83  667.7158  1.3899   0.5899
3525.92  811.7237  1.4582   0.5033
4145.57  996.2812  1.5447   0.4220
4877.79  1241.3559 1.6582   0.3471
5755.27  1582.6362 1.8143   0.2793

*BOTAPI 1 1
** pressure      oil
** psia          FVF
** -----
 14.70    1.0346
 174.00   1.0334
 365.00   1.0321
 615.00   1.0304
 865.00   1.0287
1115.00   1.0272
1365.00   1.0257
1452.23   1.0252
1929.61   1.0225
2495.91   1.0195
3182.53   1.0163
4035.30   1.0126
5123.39   1.0085

```

```

*BOTAPI 1 1
** pressure      oil
** psia          FVF
** -----
   615.00      1.0828
   865.00      1.0804
  1115.00      1.0782
  1365.00      1.0761
  1452.23      1.0754
  1929.61      1.0717
  2495.91      1.0676
  3182.53      1.0631
  4035.30      1.0581
  5123.39      1.0525

*BOTAPI 1 1
** pressure      oil
** psia          FVF
** -----
   1365.00      1.1297
   1452.23      1.1288
   1929.61      1.1240
   2495.91      1.1187
   3182.53      1.1129
   4035.30      1.1066
   5123.39      1.0994

*BOTAPI 1 1
** pressure      oil
** psia          FVF
** -----
   3182.53      1.2491
   4035.30      1.2383
   5123.39      1.2265

*BOTAPI 1 1
** pressure      oil
** psia          FVF
** -----
   4035.30      1.3101
   5123.39      1.2954

*BOTAPI 1 2
** pressure      oil
** psia          FVF
** -----
   14.70        1.0425
   174.00       1.0410
   365.00       1.0394
   865.00       1.0352
  1115.00       1.0333
  1615.00       1.0297
  2115.00       1.0264
  2365.00       1.0248
  2994.83       1.0211
  3525.92       1.0182
  4877.79       1.0117
  5755.27       1.0080

```

```

*BOTAPI 1 2
** pressure      oil
** psia          FVF
** -----
   1615.00      1.2329
   1865.00      1.2286
   2115.00      1.2245
   2365.00      1.2207
   2534.22      1.2181
   2994.83      1.2117
   3525.92      1.2049
   4145.57      1.1976
   4877.79      1.1899
   5755.27      1.1817

*BOTAPI 1 2
** pressure      oil
** psia          FVF
** -----
   2115.00      1.2869
   2365.00      1.2822
   2534.22      1.2791
   2994.83      1.2713
   3525.92      1.2630
   4145.57      1.2544
   4877.79      1.2452
   5755.27      1.2354

*BOTAPI 1 2
** pressure      oil
** psia          FVF
** -----
   3525.92      1.4582
   4145.57      1.4439
   4877.79      1.4290
   5755.27      1.4134

*BOTAPI 1 2
** pressure      oil
** psia          FVF
** -----
   4877.79      1.6582
   5755.27      1.6342

```

Condensate Model - Undersaturated Eg, Bg, Zg Table

*EGUST, *BGUST, *ZGUST

PURPOSE:

*EGUST, *BGUST or *ZGUST indicates the input of a gas expansion factor, gas formation volume factor or gas super compressibility table for condensate undersaturated gas.

FORMAT:

*EGUST *set_number*
p Eg | Bg | Zg
: :
: :

DEFINITION:

set_number

PVT region that corresponds to the table.

p

Dew point pressure of a new saturated gas mixture formed by reducing the oil content of the original saturated gas. The first entry is the driest gas dew point pressure, the last entry is the original (unreduced) saturated gas dew point pressure (kPa | psi | kPa | kg/cm²).

Eg | Bg | Zg

Gas expansion factor (formation volume factor or super compressibility) of a new saturated gas mixture formed by reducing the oil content of the original saturated gas. The first entry is the Eg/Bg/Zg at the driest gas dew point pressure, the last entry is the Eg/Bg/Zg at the original (unreduced) saturated gas dew point pressure.

DEFAULT:

Optional keyword. No default. May be used with the *GASWATER_WITH_CONDENSATE or *VOLATILE_OIL option. If not present, values of Eg/Bg/Zg from the *PVTCOND table are used.

CONDITIONS:

This keyword must be in the Component Properties section. *EGUST/*BGUST/*ZGUST may be entered when the gas density is a function of both pressure and dew point pressure. This corresponds to an undersaturated gas density that has a nonlinear functional dependence on amount of condensate within the gas phase. This keyword can only be used with the *GASWATER_WITH_CONDENSATE or *VOLATILE_OIL option.

Many *EGUST (refers to BGUST and ZGUST as well) keywords, with each *EGUST keyword followed two columns of numbers (pdew vs. Eg entries), may appear in the Component Properties Data section for each PVT region. If more than one *EGUST keyword is defined per PVT region, then the saturated gas pressure (the last pressure entry in the table) must be greater than the previous table's value and less than the next table's value. It would be normal for the first entry in each table to be atmospheric pressure. This would allow the user to input a dry gas value.

EXPLANATION:

The gas expansion factor Eg, is ideally a near linear function of pressure describing how gas density varies with gas pressure. The use of the condensate model may require the modelling of the condensate compositional dependence when condensate in gas is modelled at pressures above the dew point pressure (undersaturated conditions). The *EGUST tables allows the user to input Eg values at dew point pressures below the saturated gas pressure

If more than one PVT region is specified, then each *EGUST keyword must be followed by an integer specifying the PVT region that the dew pressure vs. Eg table represents. If the number of PVT regions is only one, then it is not necessary to specify the set_number.

The minimum number of entries specified for each table is two (Eg at a dew pressure below the saturated condition and Eg at the dew pressure of the saturated condition). If the calculated dew point pressure is outside the range of data specified, then the last table entry value of Eg is used for the calculated dew pressure.

The minimum number of *EGUST keywords required per PVT region is one. If more than one *EGUST keyword is defined per PVT region, then the saturated dew point pressure (the last pressure entry in the table) must be greater than the previous table's last value and less than the next table's last value. If more than one table is specified, then a linear interpolation is performed to determine the variation of Eg with dew point pressure for the calculated block saturated gas pressure. If the calculated block saturated gas pressure falls outside the table's range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example 1: *BGUST (using all saturation pressures available from example in *PVTCOND keyword)

```
*BGUST 1
**press    gas
**psia    FVF
**
14.70      5.83237e-3  ** Dry Gas
515.00     5.86124e-3  ** Saturated
*BGUST 1
14.70      2.74702e-3  ** Dry Gas
515.00     2.71717e-3  ** Undersaturated
1015.00    2.72302e-3  ** Saturated
*BGUST 1
14.70      1.74664e-3  ** Dry Gas
515.00     1.71313e-3  ** Undersaturated
1015.00    1.71908e-3  ** Undersaturated
1515.00    1.71098e-3  ** Saturated
```

```

*BGUST 1
14.70    1.27707e-3    ** Dry Gas
515.00   1.25368e-3    ** Undersaturated
1015.00  1.25760e-3    ** Undersaturated
1515.00  1.25228e-3    ** Undersaturated
2015.00  1.23923e-3    ** Saturated
*BGUST 1
14.70    1.02302e-3    ** Dry Gas
515.00   1.01330e-3    ** Undersaturated
1015.00  1.01483e-3    ** Undersaturated
1515.00  1.01276e-3    ** Undersaturated
2015.00  1.00808e-3    ** Undersaturated
2515.00  1.00106e-3    ** Saturated
*BGUST 1
14.70    0.87555e-3    ** Dry Gas
515.00   0.87713e-3    ** Undersaturated
1015.00  0.87682e-3    ** Undersaturated
1515.00  0.87724e-3    ** Undersaturated
2015.00  0.87860e-3    ** Undersaturated
2515.00  0.88262e-3    ** Undersaturated
3015.00  0.89373e-3    ** Saturated
*BGUST 1
14.70    0.82551e-3    ** Dry Gas
515.00   0.83126e-3    ** Undersaturated
1015.00  0.83031e-3    ** Undersaturated
1515.00  0.83161e-3    ** Undersaturated
2015.00  0.83505e-3    ** Undersaturated
2515.00  0.84268e-3    ** Undersaturated
3015.00  0.85903e-3    ** Undersaturated
3265.00  0.87063e-3    ** Saturated
*BGUST 1
14.70    0.78619e-3    ** Dry Gas
515.00   0.79518e-3    ** Undersaturated
1015.00  0.79373e-3    ** Undersaturated
1515.00  0.79570e-3    ** Undersaturated
2015.00  0.80070e-3    ** Undersaturated
2515.00  0.81100e-3    ** Undersaturated
3015.00  0.83113e-3    ** Undersaturated
3265.00  0.84458e-3    ** Undersaturated
3515.00  0.85536e-3    ** Saturated
*BGUST 1
14.70    0.75438e-3    ** Dry Gas
515.00   0.76592e-3    ** Undersaturated
1015.00  0.76408e-3    ** Undersaturated
1515.00  0.76657e-3    ** Undersaturated
2015.00  0.77278e-3    ** Undersaturated
2515.00  0.78513e-3    ** Undersaturated
3015.00  0.80809e-3    ** Undersaturated
3265.00  0.82294e-3    ** Undersaturated
3515.00  0.83464e-3    ** Undersaturated
3765.00  0.84058e-3    ** Saturated

```

```

*BGUST 1
14.70    0.72856e-3   ** Dry Gas
515.00   0.74214e-3   ** Undersaturated
1015.00  0.73999e-3   ** Undersaturated
1515.00  0.74290e-3   ** Undersaturated
2015.00  0.75007e-3   ** Undersaturated
2515.00  0.76402e-3   ** Undersaturated
3015.00  0.78920e-3   ** Undersaturated
3265.00  0.80512e-3   ** Undersaturated
3515.00  0.81754e-3   ** Undersaturated
3765.00  0.82380e-3   ** Undersaturated
4006.47  0.82634e-3   ** Saturated

```

The acceptable range of values for Gas Expansion Factor is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³	Mod. SI m³/m³
min	0.0	0.0	0.0	0.0
max	1.0E+3	5,615.0	1.0E+3	1.0E+3

Condensate Model - Undersaturated Gas Viscosity Table

*VGUST

PURPOSE:

*VGUST indicates the input of a gas viscosity table for condensate undersaturated gas

FORMAT:

*VGUST *set_number*

<i>p</i>	<i>Vg</i>
:	:
:	:

DEFINITION:

set_number

PVT region that corresponds to the *co* table.

p

Dew point pressure of a new saturated gas mixture formed by reducing the oil content of the original saturated gas. The first entry is the driest gas dew point pressure, the last entry is the original (unreduced) saturated gas dew point pressure (kPa | psi | kPa | kg/cm²).

Vg

Gas viscosity of a new saturated gas mixture formed by reducing the oil content of the original saturated gas. The first entry is gas viscosity at the driest gas dew point pressure, the last entry is the gas viscosity at the original (unreduced) saturated gas dew point pressure.

DEFAULT:

Optional keyword. No default. May be used with the *GASWATER_WITH_CONDENSATE or *VOLATILE_OIL option. If not present, values of Vg from the *PVTCOND table are used.

CONDITIONS:

This keyword must be in the Component Properties section. *VGUST may be entered when the gas viscosity is a function of both pressure and dew point pressure. This corresponds to an undersaturated gas viscosity that has a nonlinear functional dependence on amount of condensate within the gas phase. This keyword can only be used with the *GASWATER_WITH_CONDENSATE or *VOLATILE_OIL option.

Many *VGUST keywords, with each *VGUST keyword followed two columns of numbers (*pdew* vs. *Vg* entries), may appear in the Component Properties Data section for each PVT region. If more than one *VGUST keyword is defined per PVT region, then the saturated gas pressure (the last pressure entry in the table) must be greater than the previous table's value and less than the next table's value. It would be normal for the first entry in each table to be atmospheric pressure. This would allow the user to input a dry gas value.

EXPLANATION:

The gas Viscosity, V_g , is ideally a function of pressure describing how gas viscosity varies with gas pressure. The use of the condensate model may require the modelling of the condensate compositional dependence when condensate in gas is modelled at pressures above the dew point pressure (undersaturated conditions). The *VGUST tables allows the user to input V_g values at dew pressures below the saturated gas pressure

If more than one PVT region is specified, then each *VGUST keyword must be followed by an integer specifying the PVT region that the dew pressure vs. V_g table represents. If the number of PVT regions is only one, then it is not necessary to specify the *set_number*.

The minimum number of entries specified for each table is two (V_g at a dew pressure below the saturated condition and V_g at the dew pressure of the saturated condition). If the calculated dew pressure is outside the range of data specified, then the last table entry value of V_g is used for the calculated dew pressure.

The minimum number of *VGUST keywords required per PVT region is one. If more than one *VGUST keyword is defined per PVT region, then the saturated dew point pressure (the last pressure entry in the table) must be greater than the previous table's last value and less than the next table's last value. If more than one table is specified, then a linear interpolation is performed to determine the variation of V_g with dew point pressure for the calculated block saturated gas pressure. If the calculated block saturated gas pressure falls outside the tables range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example: VGUST (using all saturation pressures available from example in *PVTCOND keyword)

```
*VGUST 1
**press  gas
**psia  vis,cp
** -----
14.70  0.01251  ** Dry Gas
515.00 0.012403 ** Saturated
*VGUST 1
14.70  0.01384  ** Dry Gas
515.00 0.01396  ** Undersaturated
1015.00 0.013943 ** Saturated
*VGUST 1
14.70  0.01590  ** Dry Gas
515.00 0.01648  ** Undersaturated
1015.00 0.01638  ** Undersaturated
1515.00 0.016523 ** Saturated
*VGUST 1
14.70  0.01894  ** Dry Gas
515.00 0.02018  ** Undersaturated
1015.00 0.01997  ** Undersaturated
1515.00 0.02025  ** Undersaturated
2015.00 0.020973 ** Saturated
```

*VGUST 1

14.70	0.02287	** Dry Gas
515.00	0.02473	** Undersaturated
1015.00	0.02442	** Undersaturated
1515.00	0.02483	** Undersaturated
2015.00	0.02585	** Undersaturated
2515.00	0.027793	** Saturated

*VGUST 1

14.70	0.02918	** Dry Gas
515.00	0.03169	** Undersaturated
1015.00	0.03129	** Undersaturated
1515.00	0.03183	** Undersaturated
2015.00	0.03314	** Undersaturated
2515.00	0.03555	** Undersaturated
3015.00	0.03954	** Undersaturated
3265.00	0.041883	** Saturated

*VGUST 1

14.70	0.03094	** Dry Gas
515.00	0.03361	** Undersaturated
1015.00	0.03319	** Undersaturated
1515.00	0.03376	** Undersaturated
2015.00	0.03514	** Undersaturated
2515.00	0.03769	** Undersaturated
3015.00	0.04191	** Undersaturated
3265.00	0.04438	** Undersaturated
3515.00	0.046243	** Saturated

*VGUST 1

14.70	0.03248	** Dry Gas
515.00	0.03527	** Undersaturated
1015.00	0.03483	** Undersaturated
1515.00	0.03543	** Undersaturated
2015.00	0.03687	** Undersaturated
2515.00	0.03954	** Undersaturated
3015.00	0.04395	** Undersaturated
3265.00	0.04655	** Undersaturated
3515.00	0.04851	** Undersaturated
3765.00	0.049483	** Saturated

*VGUST 1

14.70	0.03386	** Dry Gas
515.00	0.03675	** Undersaturated
1015.00	0.03629	** Undersaturated
1515.00	0.03691	** Undersaturated
2015.00	0.03840	** Undersaturated
2515.00	0.04116	** Undersaturated
3015.00	0.04573	** Undersaturated
3265.00	0.04844	** Undersaturated
3515.00	0.05048	** Undersaturated
3765.00	0.05148	** Undersaturated
4006.47	0.051893	** Saturated

The acceptable range of values for Gas viscosity is:

	SI MPa-s	Field cp	Lab MPa-s	Mod. SI MPa-s
min	0.095	0.095	0.095	0.095
max	8.64E+4	8.64E+4	8.64E+4	8.64E+4

Oil Viscosity Above the Bubble Point as a Function of Pressure and STC Oil Density for API Tracking (Optional)

***VOTAPI**

PURPOSE:

*VOTAPI indicates the input of an oil viscosity table that is a function of pressure (P), bubble point pressure (P_b) and STC oil density in an API tracking model.

FORMAT:

*VOTAPI *set_number apigrad_number*
 p *Vo*
 \vdots \vdots

DEFINITION:

set_number

PVT region that corresponds to the votapi table.

apigrad_number

APIGRAD table number for the set number.

p

Pressure above the corresponding bubble point pressure in the APIGRAD table. (kPa | psi | kPa | kg/cm²). The first entry of this table is the oil viscosity value at the bubble point pressure. All subsequent pressure entries must be greater than this bubble point pressure and must increase in a monotonic fashion.

Vo

Oil viscosity entry at the corresponding pressure. mPa-s | cp | cp | mPa-s).

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be in the Component Properties section and is only used for API tracking problems. *VOTAPI may be entered when the oil viscosity is a function of pressure, bubble point pressure and API. This corresponds to an oil viscosity which has a nonlinear functional dependence on pressures above the bubble point pressure and which can vary depending on the STC density. The *VOTAPI keyword cannot be used when oil viscosity above bubble point is entered in another manner in the same or another PVT region (for example using *CVO).

When inputting a VOTAPI table *apigrad_number* should vary fastest. All APIGRAD table numbers should be read in for the first PVT region before reading all APIGRAD table numbers for the second and subsequent PVT regions. VOTAPI tables for all APIGRAD tables and all PVT regions need to read in if this option is used.

EXPLANATION:

The oil viscosity can exhibit a weak compositional dependence when the oil is above the bubble point. The *VOTAPI table allows us to model this dependence varying with bubble point pressure, block pressure and STC oil density. When inputting a VOTAPI Vo table *apigrad_number* should vary fastest. All APIGRAD table numbers should be read in for the first PVT region before reading all APIGRAD table numbers for the second and subsequent PVT regions. VOTAPI tables for all APIGRAD tables and all PVT regions need to read in if this option is used.

Many *VOTAPI keywords, with each *VOTAPI keyword followed by two columns of numbers (*p* vs. *Vo* entries), may appear in the Component Properties Data section. Each *VOTAPI keyword must be followed by an integer (*set_number*) specifying the PVT region that the *p* vs. *Vo* table represents.

Following the *set_number*, an *apigrad_number* must appear, in API tracking models each PVT region must have at least two APIGRAD tables, the apigrad number associates the VOTAPI table to a particular (1st, 2nd, 3rd etc) APIGRAD table for each PVT region.

Two or more VOTAPI tables with identical values of *set_number* and *apigrad_number* refer to undersaturated Bo in the same APIGRAD table but at different bubble points (saturation pressures).

All APIGRAD tables must be given VOTAPI tables.

The minimum number of entries specified for each table is two (*Vo* at the bubble point and *Vo* at a pressure above the bubble point). If the calculated block pressure is outside the range of data specified, then the last table entry value of *Vo* is used for the calculated block pressure.

The minimum number of required *VOTAPI keywords per apigrad table number (for each PVT region) is one. If more than one *VOTAPI keyword is defined per apigrad table number, then the bubble point pressure (the first pressure entry in the table) must be greater than the previous table's value and less than the next table's value. If more than one table is specified, then a linear interpolation is performed to determine the variation of *Vo* with *p* for the calculated block bubble point pressure in a simulation block.

The VOTAPI calculation performs this calculation in a VOTAPI table with an STC oil density less than the block value and in a VOTAPI table with an STC oil density greater than the simulation block value.

Finally the VOTAPI table calculation interpolates with respect to the block STC density to determine the viscosity of the oil at the specific STC oil density of the simulation block.

If the calculated block bubble point pressure falls outside the tables range, then the table which is closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example:

```
*APIGRAD  56.4239 ** 25 api
** p          rs        bo      viso
  14.70      0.0000   1.0346   2.3982
 174.00     24.3247   1.0509   2.3709
 365.00     52.4846   1.0660   2.3297
 615.00     85.6892   1.0828   2.2562
 865.00    118.2507   1.0986   2.1701
1115.00    150.9454   1.1142   2.0771
1365.00    184.0502   1.1297   1.9812
1452.23    195.6845   1.1352   1.9478
1929.61    261.0615   1.1653   1.7661
2495.91    342.6808   1.2024   1.5622
3182.53    447.5103   1.2491   1.3433
4035.30    587.1534   1.3101   1.1170
5123.39    782.4795   1.3934   0.8916
*APIGRAD  53.6858 ** 33 API
** p          rs        bo      viso
  14.70      0.0000   1.0425   1.7622
 174.00     38.8883   1.0724   1.5043
 365.00     87.3174   1.1011   1.3908
 615.00    139.5046   1.1294   1.2735
 865.00    189.5810   1.1555   1.1709
1115.00    239.8258   1.1811   1.0777
1365.00    291.0484   1.2068   0.9924
1615.00    343.6380   1.2329   0.9142
1865.00    397.8353   1.2595   0.8427
2115.00    453.8270   1.2869   0.7774
2365.00    511.7849   1.3150   0.7178
2534.22    552.2113   1.3345   0.6805
2994.83    667.7158   1.3899   0.5899
3525.92    811.7237   1.4582   0.5033
4145.57    996.2812   1.5447   0.4220
4877.79   1241.3559   1.6582   0.3471
5755.27   1582.6362   1.8143   0.2793
*VOTAPI 1 1
** pressure      oil
** psia         vis,cp
** -----
      14.70      2.3982
     174.00      2.4339
     365.00      2.4762
     615.00      2.5307
     865.00      2.5843
    1115.00      2.6370
    1365.00      2.6889
    1452.23      2.7067
    1929.61      2.8029
    2495.91      2.9131
    3182.53      3.0416
    4035.30      3.1937
    5123.39      3.3767
```

```

*VOTAPI 1 1
** pressure      oil
** psia         vis,cp
** -----
    865.00      2.1701
   1115.00      2.2306
   1365.00      2.2905
   1452.23      2.3112
   1929.61      2.4231
   2495.91      2.5526
   3182.53      2.7050
   4035.30      2.8876
   5123.39      3.1102

*VOTAPI 1 1
** pressure      oil
** psia         vis,cp
** -----
    1452.23      1.9478
   1929.61      2.0555
   2495.91      2.1806
   3182.53      2.3287
   4035.30      2.5072
   5123.39      2.7262

*VOTAPI 1 1
** pressure      oil
** psia         vis,cp
** -----
    2495.91      1.5622
   3182.53      1.6912
   4035.30      1.8482
   5123.39      2.0432

*VOTAPI 1 1
** pressure      oil
** psia         vis,cp
** -----
    3182.53      1.3433
   4035.30      1.4822
   5123.39      1.6561

*VOTAPI 1 1
** pressure      oil
** psia         vis,cp
** -----
    4035.30      1.1170
   5123.39      1.2639

```

```

*VOTAPI 1 2
** pressure      oil
** psia         vis,cp
** -----
    14.70      1.7622
    174.00     1.7929
    615.00     1.8765
    865.00     1.9229
   1365.00     2.0140
   1615.00     2.0586
   1865.00     2.1026
   2365.00     2.1889
   3525.92     2.3806
   4877.79     2.5898
   5755.27     2.7181

*VOTAPI 1 2
** pressure      oil
** psia         vis,cp
** -----
    865.00     1.1709
   1115.00     1.2125
   1365.00     1.2537
   1865.00     1.3354
   2534.22     1.4429
   2994.83     1.5157
   3525.92     1.5985
   4877.79     1.8031
   5755.27     1.9313

*VOTAPI 1 2
** pressure      oil
** psia         vis,cp
** -----
   1615.00     0.9142
   1865.00     0.9482
   2115.00     0.9821
   2365.00     1.0158
   2534.22     1.0386
   2994.83     1.1003
   3525.92     1.1708
   4877.79     1.3471
   5755.27     1.4589

*VOTAPI 1 2
** pressure      oil
** psia         vis,cp
** -----
   2534.22     0.6805
   2994.83     0.7280
   3525.92     0.7826
   4145.57     0.8462
   4877.79     0.9210
   5755.27     1.0100

*VOTAPI 1 2
** pressure      oil
** psia         vis,cp
** -----
   4877.79     0.3471
   5755.27     0.3900

```

Reference Pressure for Gas-Oil Capillary Pressure (Optional)

*REFPST

PURPOSE:

*REFPST indicates the input of a reference pressure for the gas-oil (liquid) capillary pressure input.

FORMAT:

*REFPST *pst*

DEFINITION:

pst

Value for reference pressure for gas-oil (liquid) capillary pressure (kPa | psi | kPa | kg/cm²).

DEFAULT:

Optional keyword. No default.

CONDITIONS:

*REFPST needs to be entered when the surface tension is entered with the gas-oil PVT tables. With the surface tension, this reference pressure determines the gas-oil (liquid) capillary pressure.

EXPLANATION:

The calculation for interfacial effects on gas-oil capillary pressure is:

$$P_{cog} = \frac{\text{int t}}{\text{int tI}} * P_{cogl}$$

where:

- | | | |
|-------|---|--|
| intt | - | interfacial tension at current block pressure |
| inttI | - | interfacial tension at reference pressure, <i>pst</i> |
| Pcogl | - | gas-oil capillary pressure at reference pressure, <i>pst</i> |

For the *GASWATER model, the capillary pressure modified is *Pcwg* entered in the *SWT table, for the *GASWATER_WITH_CONDENSATE model, the capillary pressure modified is *Pclg* entered in the *SWT table.

The acceptable range of values for reference pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,039	1.0E+6	1.01E+4

Solvent PVT Table (Conditional)

*PPTS

PURPOSE:

*PPTS indicates the start of the solvent PVT table.

FORMAT:

```
*PPTS (*ES | *BS | *ZS) set_number
      p      rss      es | bs | zs      viss      omega_os
      :      :      :      :      :
```

DEFINITIONS:

set_number

Set_number for this particular PVT region. This is the number used with *PTYPE to assign *PPTS tables to grid blocks.

*ES

Keyword indicating that the solvent expansion factor will be used. If all *ES, *BS and *ZS are not present in the data set then this is the default.

*BS

Keyword indicating the solvent formation volume factor will be used instead of the solvent expansion factor.

*ZS

Keyword indicating that the solvent compressibility factor will be used instead of the solvent expansion factor. The reservoir temperature TRES must be specified before the *PPTS keyword in order to use this option.

p

Pressure (kPa | psi | kPa | kg/cm²) in oil phase.

rss

Solvent-water ratio (m³/m³ | scf/STB | cm³/cm³) at the given pressure *p*.

es

Solvent expansion factor (m³/m³ | scf/RB | cm³/cm³) at pressure *p*.

bs

Solvent formation volume factor (m³/m³ | RB/scf | cm³/cm³).

zs

Solvent compressibility factor (dimensionless) at pressure *p*.

viss

Solvent phase viscosity (mPa-s | cp | cp) at pressure *p*.

omega_os

Mixing parameter between oil and solvent. Its value is bounded between 0 and 1 representing the immiscible case and the completely miscible case. It is a dimensionless fraction. The last entry for *omega_os* must be greater than zero (if the *OMEGA_OST table is not used). When the *OMEGA_OST table is used, the values of *omega_os* are ignored and may all be set to zero or left blank. See *OMEGA_OST keyword for further information.

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *MISCG or *MISNCG.

If in a multiple PVT region run, a single *PVTS table is encountered it will be applied to all PVT regions, in this case, do not include a *set_number* on the *PVTS line.

If in a multiple PVT region run, multiple *PVTS tables are encountered, one must be defined for each PVT region. Set_numbers must be included.

EXPLANATION:

Typical PVT data curves used by this simulator are given by the solid lines in Figure 7(a), (b), (c), (d), and (e) in Appendix D. Note that they are the curves that would be obtained if there were unlimited gas present (saturated conditions).

The dotted lines in Figure 7(a), (b), and (d) indicate the usual curves obtained when the pressure exceeds the bubble point pressure (PB). The latter curves are assumed to be linear.

The gradient of the Rss curve above the bubble point is equal to zero. IMEX automatically uses a variable substitution technique to model undersaturated conditions (see theoretical outline in Appendix A).

The minimum pressure in the PVT table should be below any pressure constraint specified on a production well in the Well Data section. Normally a value of zero would be chosen. The maximum pressure in the table should be comfortably above any well injection pressure specified in the Well Data section. The formation volume factor and viscosity of water are calculated from functional relationships.

When the pseudo-miscible option is enabled a fourth mass conservation equation is used to simulate the solvent component. Solvent can dissolve in water but not in oil.

The gas mass conservation equation can be used to simulate chase gas injection. Obviously, the reservoir must be undersaturated and the solution gas should not vaporize during the simulation. If chase gas is not injected, the gas properties entered are for the gas-in-place.

For more information about the pseudo-miscible formulation, consult Todd and Longstaff, 1972, "The Development, Testing, and Application of a Numerical Simulator for Predicting Miscible Fluid Performance", Trans. AIME 253, p874. For complete details of the actual formulation, consult the IMEX Technical Manual.

Entries must be in order of increasing pressure

$$rss = \frac{(\text{volume of solvent at STC})}{(\text{volume of water at STC})}$$

$$es = \frac{(\text{volume of solvent at STC})}{(\text{volume of solvent at reservoir conditions})}$$

$$bs = \frac{(\text{volume of solvent at reservoir conditions})}{(\text{volume of solvent at STC})}$$

The default is *PVTS *ES. Use *BS to enter values for *bs* instead of *es* and *ZS to enter values of *zs* instead of *es*.

*PVTS				
** p	rss	es	viss	omega_os
14.7	0.0	4.480	0.011	0.0
500.0	0.0	178.501	0.012	0.0
1000.0	0.0	395.100	0.013	0.0
1200.0	0.0	491.304	0.014	0.0
1500.0	0.0	641.313	0.016	0.0
1800.0	0.0	790.077	0.018	0.0
2000.0	0.0	885.269	0.019	0.0
2302.3	0.0	1020.096	0.022	0.0
2500.0	0.0	1100.715	0.023	0.283
3000.0	0.0	1280.902	0.027	1.0
3500.0	0.0	1429.797	0.031	1.0
4000.0	0.0	1555.210	0.034	1.0
4500.0	0.0	1661.958	0.037	1.0
4800.0	0.0	1719.100	0.038	1.0

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+10	1.45E+9	1.0E+10	1.01E+8

The acceptable range of values for the solvent- water ratio is:

	SI m³/m³	Field scf/STB	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

The acceptable range of values for the solvent expansion factors is:

	SI m³/m³	Field Rb/STB	Lab cm³/cm³
min	1.0E-20	5.671E-20	1.0E-20
max	1.0E+3	5617	1.0E+3

The acceptable range of values for the formation volume factor is:

	SI m³/m³	Field Rb/STB	Lab cm³/cm³
min	0.5	0.5	0.5
max	5.0	5.0	5.0

The acceptable range of values for the solvent viscosity is:

	SI mPa·sec	Field cp	Lab mPa·sec
min	.0095	.0095	.0095
max	86,400.	86,400.	86,400.

The acceptable range of values for the omega factor (a fraction) is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

Extended Oil-Solvent Mixing Parameter Table (Optional)

*OMEGA_OST

PURPOSE:

*OMEGA_OST indicates the input of an extended Omega (oil-solvent mixing parameter) table. This table allows the user to define two separate types of Omega, one to be used in the oil-solvent density mixing rule (*omega_den*) and one to be used in the oil-solvent viscosity mixing rule (*omega_vis*). Sorm may also be altered in this table using the *sormadd* column. Finally, it is possible to use the *OMEGA_OST table to define tables that are functions of both pressure (*P*) and bubble point pressure (*Pb*).

FORMAT:

```
*OMEGA_OST set_number
  p      omega_den  omega_vis  sormadd
  :      :          :          :
```

DEFINITION:

set_number

PVT region that corresponds to the OMEGA_OST table.

p

Pressure (kPa | psi | kPa | kg/cm²). The first row of this table contains parameters at the bubble point pressure (*Pb*) equal to the first pressure in the table. All subsequent pressure entries must be greater than this pressure and must increase in a monotonic fashion. If *omega_den*, *omega_vis*, and *sormadd* are not functions of bubble point pressure, enter a single *OMEGA_OST table for each PVT region. In this case, the pressure in the first row of the table does not correspond to a bubble point pressure.

omega_den

Mixing parameter between oil and solvent for use in the oil-solvent density calculation. Its value is bounded between 0 and 1, representing the immiscible case and the completely miscible case. It is a dimensionless fraction. The last entry for *omega_den* in each table must be greater than zero.

omega_vis

Mixing parameter between oil and solvent for use in the oil-solvent viscosity calculation. Its value is bounded between 0 and 1, representing the immiscible case and the completely miscible case. It is a dimensionless fraction. The last entry for *omega_vis* in each table must be greater than zero.

sormadd

Sorm entered in the *SWT table defines the irreducible oil saturation (fraction) due to water blocking at a given water saturation. When non-zero *sormadd* is entered in the *OMEGA_OST table, the final *sorm* used by the

model is modified to include *sormadd*. Thus $sorm = sorm(Sw) + sormadd(P, Pb)$. *Sormadd* may be negative or positive; the final value of *sorm* will be bounded between 0 and 1.

DEFAULT:

Optional keyword. No default.

CONDITIONS:

This keyword must be in the Component Properties section. *OMEGA_OST may be entered whether or not its columns are functions of bubble point pressure (*Pb*). This allows the user to take advantage of separate mixing parameters for density and viscosity as well as the *sormadd* functionality without considering the *OMEGAS_OST table columns as function of *Pb*. When *OMEGA_OST is read, the *omega_os* entered in the last column of the *PVTS table is ignored and may be left out.

This keyword must be in the Component Property keyword group. It is optional, but must be used only with the *MISCG or *MISNCG PVT model.

EXPLANATION:

The *OMEGA_OST table allows the user to model more complex oil-solvent mixing rules and *sorm* functions than are available when using the *PVTS table (*omega_os*) and *SWT table (*sorm*).

The *OMEGA_OST table, when one table per PVT region is used (i.e. no dependence on *Pb*), can model separate mixing rules for oil-solvent density (Appendix E, equations E.2.8-E.2.11) and oil-solvent viscosity (Appendix E, equations E.2.12-E.2.15) and the effect of pressure on *sorm*.

The *OMEGA_OST tables, when multiple tables per PVT region are entered, can additionally include the effect of bubble point pressure. In this case the pressure entered in the first row of each table within a PVT region corresponds to different bubble point pressures (which must monotonically increase from table to table within a PVT region). Interpolation is done with respect to *OMEGA_OST table bubble point pressures (pressure in the first row of each table) and with respect to pressures within each table.

Many *OMEGA_OST keywords, with each *OMEGA_OST keyword followed by four columns of numbers, may appear in the Component Properties Data section. If more than one PVT region is specified, then each *OMEGA_OST keyword must be followed by an integer specifying the PVT region set number for that table. If the number of PVT regions is only one, then it is not necessary to specify the set_number.

The minimum number of entries specified within each table is two. If the calculated block pressure is outside the range of data specified, then the closest table value of *omega_den*, *omega_vis*, and *sormadd* is used.

The minimum number of *OMEGA_OST keywords per PVT region required is one. If more than one *OMEGA_OST keyword is defined per PVT region, then the bubble point pressure (the first pressure entry in the table) must be greater than the previous table's value and less than the next table's value. If more than one table is specified, then a linear interpolation is performed to determine the variation of parameters with *P* for the interpolated block bubble point pressure. If the calculated block bubble point pressure falls outside the tables range, then the table which is

closest to the calculated value is used. In other words, values are not extrapolated beyond the range of data specified.

Example:

```
*OMEGA_OST 1
** p    omega_den omega_vis    sormadd
 514.7      0.00      0.0      0.00
1014.7      0.00      0.0      0.05
2014.7      0.30      0.10     0.03
2514.7      0.65      0.30     0.01
3014.7      0.90      0.80    -0.01
4014.7      0.90      1.00    -0.03
5014.7      0.90      1.00    -0.01
9014.7      0.90      1.00     0.03

*OMEGA_OST 1 **(Second reference to PVT region 1)
**
          (Therefore Pb interpolation turned on)
**
          (Interpolation between Pb 514.7 and 1014.7)

** p    omega_den omega_vis    sormadd
1014.7      0.00      0.0      0.06
2014.7      0.00      0.10     0.04
2514.7      0.55      0.20     0.03
3014.7      0.95      0.70     0.04
4014.7      0.95      0.90     0.06
5014.7      0.95      1.00     0.07
9014.7      0.95      1.00     0.10
```

Densities (Required)

*DENSITY, *GRAVITY

PURPOSE:

*DENSITY indicates the input of a density (for a PVT region).

*GRAVITY indicates the input of a gravity (for a PVT region).

FORMAT:

```
*DENSITY *OIL | *GAS |*WATER | *SOLVENT value
*GRAVITY *GAS gas_gravity
```

DEFINITIONS:

*OIL

Keyword signaling that this is the oil phase density at STC.

*GAS

Keyword signaling that this is the gas phase density or gas phase gravity at STC.

*WATER

Keyword signaling that this is the water phase density at STC.

*SOLVENT

Keyword signaling that this is the solvent phase density at STC. (Required for *MISCG and *MISNCG).

value

Value of density (kg/m³ | lbm/ft³ | g/cm³).

gas_gravity

Value of gas specific gravity (Air = 1).

DEFAULTS:

Required keyword. No default values. If the API or GASWATER option is used, *DENSITY *OIL is not required.

CONDITIONS:

These keywords must be in the Component Property keyword group. *SOLVENT is required for *MISCG and *MISNCG. One of *DENSITY *GAS or *GRAVITY *GAS must be present in the data set. The *DENSITY *OIL keyword cannot be used with the *GASWATER option (The oil phase is not modelled).

EXPLANATION:

Note that if multiple PVT regions are used a *DENSITY keyword for oil and gas must be specified corresponding to each PVT region.

If a single *DENSITY keyword for water or solvent is encountered it will be applied to all PVT regions.

If multiple *DENSITY keywords for water or solvent are encountered, one must be defined for each PVT region.

The densities will be read in the same order that the PVT tables were specified i.e., the first *DENSITY keyword will correspond to the first PVT table and so on. STC means stock tank conditions (usually 14.7 psia, 60 F). If gas gravity is entered it is converted internally to kg/m³:

$$\text{gas_density (kg/m}^3) = \text{gas_gravity} * 1.2222$$

Each of the oil, gas, and water densities must be input. Solvent density must be input when using the pseudo-miscible option.

Example:

```
*DENSITY *OIL      38.53
*DENSITY *GAS      0.06864
*DENSITY *WATER    62.4
*DENSITY *SOLVENT   0.06248
```

The acceptable range of values for oil density is:

	SI kg/m³	Field lbm/ft³	Lab g/cm³
min	250.0	15.60	0.25
max	1500.0	93.64	1.5

The acceptable range of values for gas density is:

	SI kg/m³	Field lbm/ft³	Lab g/cm³
min	0.05	3.12E-3	5.0E-5
max	50.0	3.12	0.05

The acceptable range of values for gas gravity is:

	SI Air=1	Field Air=1	Lab Air=1
min	0.05	0.05	0.05
max	50.0	50.0	50.0

The acceptable range of values for water density is:

	SI kg/m³	Field lbm/ft³	Lab g/cm³
min	500.0	31.21	0.5
max	1500.0	93.64	1.5

The acceptable range of values for solvent density is:

	SI kg/m³	Field lbm/ft³	Lab g/cm³
min	0.001	6.24E-5	1.0E-6
max	100.0	6.24	0.01

Under-Saturated Oil Compressibility (Optional)

*CO

PURPOSE:

*CO indicates input of oil compressibility (for a PVT region). (Alternate method for inputting oil compressibility)

FORMAT:

*CO *oil_compress*

DEFINITIONS:

oil_compress

Compressibility of the oil phase when above the bubble point pressure (inverse pressure) (1/kPa | 1/psi | 1/kPa | 1/(kg/cm²)).

DEFAULTS:

Optional keyword. No default value.

CONDITIONS:

This keyword must be in the Component Property keyword group. This keyword must be omitted if oil compressibility is input in the PVT table as a function of pressure, or if the *COT or *BOT options are used.

If a single *CO keyword is encountered it will be applied to all PVT regions.

If multiple *CO keywords are encountered, one must be defined for each PVT region.

The *CO keyword cannot be used with the *GASWATER option (The oil phase is not modelled) or the *GASWATER_WITH_CONDENSATE option (The oil phase is always saturated with gas).

EXPLANATION:

oil_compress gives the slope of the *bo* curve divided by *bo(Pb)* when the oil is above the bubble point pressure. In IMEX this is taken as a constant value, independent of the bubble point pressure. Refer to Figure 7(b) in Appendix D.

$$bo(p) = bo(pb) \cdot [1 - co \cdot (p - pb)]$$

where *pb* is the bubble point pressure. The acceptable range of values for oil compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

Oil Phase Viscosity Pressure Dependence (Optional)

*cvo

PURPOSE:

*CVO signals the input of the pressure dependence of oil viscosity (for a PVT region).

FORMAT:

*CVO *cviso*

DEFINITIONS:

cviso

Pressure dependence of the viscosity curve for oil above the bubble point pressure (mPa·s/kPa | cp/psi | cp/(kg/cm²)).

DEFAULTS:

Optional. If *CVO is not present the default is zero.

CONDITIONS:

This keyword must be in the Component Property keyword group. The *CVO keyword cannot be used with the *GASWATER option (The oil phase is not modelled) or the *GASWATER_WITH_CONDENSATE option (The oil phase is always saturated with gas). The *CVO option cannot be used if the *VOT option is used.

If a single *CVO keyword is encountered it will be applied to all PVT regions.

If multiple *CVO keywords are encountered, one must be defined for each PVT region.

EXPLANATION:

In IMEX, *cviso* is assumed to be a constant value, independent of the bubble point pressure. Refer to Figure 7(b) in Appendix D. It is calculated by the equation:

$$viso(p) = viso + cviso \cdot (p - pb)$$

where *pb* is the bubble point pressure.

The acceptable range of values for oil viscosity pressure dependence is:

	SI mPa·s/kPa	Field cp/psi	Lab cp/(kg/cm ²)
min	0.0	0.0	0.0
max	10.0	68.9	0.102

Water Formation Volume Factor (Required) *BWI, *CW, *REFPW

PURPOSE:

*BWI indicates the input of the water formation volume factor (for a PVT region).

*CW indicates the input of water compressibility (for a PVT region).

*REFPW indicates the input of reference pressure (for a PVT region).

FORMAT:

*BWI	<i>bwi</i>
*CW	<i>cw</i>
*REFPW	<i>prw</i>

DEFINITIONS:

bwi

Water formation volume factor (m^3/m^3 | RB/STB | cm^3/cm^3) at reference pressure *prw*.

cw

Water compressibility ($1/\text{kPa}$ | $1/\text{psi}$ | $1/\text{kPa}$ | $1/(\text{kg}/\text{cm}^2)$).

prw

Reference pressure (kPa | psi | kPa | kg/cm^2) at which water formation volume factor *bwi* is calculated.

DEFAULTS:

Required keywords. No default values.

CONDITIONS:

These keywords must be in the Component Property keyword group.

If a single *BWI, *CW, or *REFPW keyword is encountered it will be applied to all PVT regions.

If multiple *BWI, *CW, or *REFPW keywords are encountered, one must be defined for each PVT region.

EXPLANATION:

The water phase formation volume factor, *bw*, is calculated according to the formula

$$bw = bwi \cdot [1 - cw \cdot (p - prw)]$$

where *p* is the oil-phase pressure.

$$bw = \frac{(\text{volume of water at reservoir conditions})}{(\text{volume of water at standard conditions})}$$

The acceptable range of values for water formation volume factor is:

	SI m³/m³	Field RB/STB	Lab cm³/cm³
min	0.5	0.5	0.5
max	1.5	1.5	1.5

The acceptable range of values for water compressibility is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-2

The acceptable range of values for reference pressure is:

	SI 1/kPa	Field 1/psia	Lab 1/Pa	Mod. SI 1/(kg/cm²)
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038	1.0E+6	1.0E+4

Water Phase Viscosity (Required)

*VWI, *CVW

PURPOSE:

*VWI signals the input of water viscosity (for a PVT region).

*CVW signals the input of cvw (for a PVT region).

FORMAT:

*VWI	<i>vwi</i>
*CVW	<i>cvw</i>

DEFINITIONS:

vwi

Viscosity of water phase at the reference pressure prw (mPa-s | cp | cp).

cvw

Pressure dependence of water viscosity (viscosity units/pressure units).

DEFAULTS:

Required keywords. Default for *VWI is 1.0 cp. Default for *CVW is 0.0.

CONDITIONS:

These keywords must be in the Component Property keyword group. If a single *VWI or *CVW keyword is encountered it will be applied to all PVT regions. If multiple *VWI or *CVW keywords are encountered, one must be defined for each PVT region.

Care must be made in the choice of CVW as it is constant over pressure. Values that are too large may cause the water viscosity to become unphysical when *REFPW is large compared to block or bottom-hole pressure.

Typical *CVW values for pure water:

@ 77 deg. F	= 0.0 cp/psi
@ 122 deg. F	= 1.3e-6 cp/psi
@ 167 deg. F	= 1.7e-6 cp/psi

EXPLANATION:

The water viscosity, vw, is calculated according to the formula:

$$vw = vwi + cvw \cdot (p - prw)$$

where *p* is the oil-phase pressure. *prw* is a reference pressure input in *REFPW.

The acceptable range of values for water viscosity is:

	SI mPa·s	Field cp	Lab cp
min	0.1	0.1	0.1
max	2.0	2.0	2.0

The acceptable range of values for the change in water viscosity is:

	SI mPa·s mPa·s/kPa	Field cp cp/psi	Lab cp cp/kPa	Mod. SI mPa·s mPa·s/(kg/cm ²)
min	0.0	0.0	0.0	0.0
max	1.0E-3	6.89E-3	1.0E-3	1.0E-1

PVT Type (Optional)

***PTYPE**

PURPOSE:

*PTYPE indicates the start of input of PVT region types. A PVT table set and if specified, a bubble point pressure vs. depth table is assigned to each grid block (see *PVT and *PBT).

ARRAY:

*PTYPE

DEFAULTS:

Optional keyword. The default is *PTYPE *CON 1, where 1 is a value for set_number. This corresponds to one PVT region only.

CONDITIONS:

This keyword must be in the Component Property keyword group, after all the PVT data tables have been entered.

EXPLANATION:

If the reservoir PVT properties differ from region to region, the reservoir can be subdivided further by assigning a PVT table to each subregion. If more than one set of PVT tables is input, then a set of tables must be assigned to each grid block.

*PTYPE 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.

Maximum Rate of Increase of Solution Gas Ratio or Bubble Point Pressure (Optional)

*DRSDT, *DPBDT

PURPOSE:

*DRSDT signals the input of the maximum rate of increase of solution gas ratio (for a PVT region). *DPBDT signals the input of the maximum rate of increase of bubble point pressure (for a PVT region).

*DRSDT and *DPBDT limits are not applied to reductions of solution gas ratio or bubble point pressure (respectively).

FORMAT:

*DRSDT (*FREEGAS) *value_drsdt*
or
*DPBDT (*FREEGAS) *value_dpbdt*

DEFINITIONS:

value_drsdt

Maximum rate of increase of solution gas ratio. ($\text{m}^3/\text{m}^3/\text{day}$ | $\text{scf}/\text{STB}/\text{day}$ | $\text{cm}^3/\text{cm}^3/\text{min}$ | $\text{m}^3/\text{m}^3/\text{day}$).

value_dpbdt

Maximum rate of increase of bubble point pressure. (kPa/day | Psi/day | kPa/minute | $\text{kg}/\text{cm}^2/\text{day}$).

*FREEGAS

If the *FREEGAS sub-keyword is present, the DRSDT or *DPBDT limit is only applied to blocks which contain free gas (gas saturation > 0.0).

If *FREEGAS is not present, the *DRSDT or *DPBDT limit is applied to all blocks, whether gas exists in the block or not, and if the *DRSDT or *DPBDT limit is exceeded in a block without free gas, free gas will immediately appear in that block.

DEFAULTS:

Optional. If *DRSDT or *DPBDT is not present in a data set, the maximum rate of solution of gas in oil is infinite.

CONDITIONS:

This keyword must be in the Component Property keyword group.

*DRSDT or *DPBDT has no effect on reductions of solution gas ratio or bubble point pressure (respectively).

The *DRSDT or *DPBDT keywords cannot be used with the API tracking options.

If a single *DRSDT or *DPBDT keyword is encountered, it will be applied to all PVT regions.

If multiple *DRSDT keywords are encountered, one must be defined for each PVT region.

If multiple *DPBDT keywords are encountered, one must be defined for each PVT region.

*DRSDT and *DPBDT cannot be used in different PVT regions. All PVT regions must either use *DRSDT or *DPBDT.

Then using *DRSDT or *DPBDT, the *FREEGAS sub-keyword must appear in either all or none of the PVT regions. If this is not true, the last PVT region defined will be used to determine whether or not the *FREEGAS option is used in all PVT regions.

value_drsdt or *value_dpbdt* cannot be less than zero. If these values are equal to zero or small, a gas phase can occur when undersaturated oil moves into cells of even more undersaturated oil (lower R_s). The *FREEGAS option may be used to prevent this from occurring.

*DPBDT can be used with the volatile oil option. *DRSDT can be used with the volatile oil option provided that dR_s/dP_b in the *PVTCOND or *PVTVO table is always positive.

EXPLANATION:

In IMEX, by default, when bubble point pressure or solution gas ratio rises, gas is allowed to dissolve in oil in an unrestricted manner. *DRSDT/*DPBDT can be used to limit the rate at which gas can dissolve in the oil.

In blocks with free gas, if *DRSDT or *DPBDT is exceeded, this will result in what is effectively a bubble point pressure lower than the block pressure.

In blocks without free gas, if *DRSDT or *DPBDT is exceeded, this will result in the appearance of a gas phase at the current bubble point pressure/solution gas ratio present in the block.

It is possible when using *DRSDT or *DPBDT that the material balance deteriorates over time when the timestep converges in a single Newton iteration. The material balance equations may have “just” converged when a *DRSDT limit has been applied after the first iteration. Rather than force a tighter MAXRES convergence tolerance to control this, the keyword MINCYC has been added in the NUMERICAL section to give the user control of the minimum number of Newton iterations per timestep. MINCYC 2 will overcome this issue.

Example:

```
*DRSDT  0.45 ** scf/stb/day
```

Maximum Rate of Increase of Oil Content or Dew Point Pressure (Optional)

*DRVDT, *DPDWDT

PURPOSE:

- *DRVDT signals the input of the maximum rate of increase of oil content (for a PVT region).
- *DPDWDT signals the input of the maximum rate of increase of dew point pressure (for a PVT region).
- *DRVDT and *DPDWDT limits are not applied to reductions of oil content or dew point pressure (respectively).

FORMAT:

*DRVDT (*FREEOIL) *value_drvdt*
or
*DPDWDT (*FREEOIL) *value_dpdwtdt*

DEFINITIONS:

value_drvdt

Maximum rate of increase of oil content. ($\text{m}^3/\text{m}^3/\text{day}$ | STB/scf/day | $\text{cm}^3/\text{cm}^3/\text{min}$ | $\text{m}^3/\text{m}^3/\text{day}$).

value_dpdwtdt

Maximum rate of increase of dew point pressure. (kPa/day | Psi/day | kPa/minute | $\text{kg}/\text{cm}^2/\text{day}$).

*FREEOIL

If the *FREEOIL sub-keyword is present, the *DRVDT or *DPDWDT limit is only applied to blocks which contain free oil (oil saturation > 0.0).

If *FREEOIL is not present, the *DRVDT or *DPDWDT limit is applied to all blocks, whether oil exists in the block or not, and if the *DRVDT or *DPDWDT limit is exceeded in a block without free oil, free oil will immediately appear in that block.

DEFAULTS:

Optional. If *DRVDT or *DPDWDT is not present in a data set, the maximum rate of solution of oil in gas is infinite.

CONDITIONS:

This keyword must be in the Component Property keyword group.

*DRVDT or *DPDWDT has no effect on reductions of oil content or dew point pressure (respectively).

The *DRVDT or *DPDWDT keywords cannot be used with the API tracking options.

If a single *DRVDT or *DPDWDT keyword is encountered, it will be applied to all PVT regions.

If multiple *DRVDT keywords are encountered, one must be defined for each PVT region.
If multiple *DPDWDT keywords are encountered, one must be defined for each PVT region.
*DRVDT and *DPDWDT cannot be used in different PVT regions. All PVT regions must either use *DRVDT or *DPDWDT.

When using *DRVDT or *DPDWDT, the *FREEOIL sub-keyword must appear in either all or none of the PVT regions. If this is not true, the last PVT region defined will be used to determine whether or not the *FREEOIL option is used in all PVT regions.

value_drvdt or *value_dpdwdt* cannot be less than zero. If these values are equal to zero or small, an oil phase can occur when undersaturated gas moves into cells of even more undersaturated gas (lower R_v). The *FREEOIL option may be used to prevent this from occurring.

*DRVDT or *DPDWDT can be used with the volatile oil option only. In addition dRv/dPdw in the *PVTCOND or *PVTVO table must always be positive.

EXPLANATION:

In IMEX, by default, when dew point pressure or oil content rises, oil is allowed to exist in gas in an unrestricted manner. *DRVDT/*DPDWDT can be used to limit the rate at which oil can move into the gas.

In blocks with free oil, if *DRVDT or *DPDWDT is exceeded, this will result in what is effectively a dew point pressure lower than the block pressure.

In blocks without free oil, if *DRVDT or *DPDWDT is exceeded, this will result in the appearance of a oil phase at the current dew point pressure/oil content present in the block.

It is possible when using *DRVDT or *DPDWDT that the material balance deteriorates over time when the timestep converges in a single Newton iteration. The material balance equations may have “just” converged when a *DRVDT limit has been applied after the first iteration. Rather than force a tighter MAXRES convergence tolerance to control this, the keyword MINCYC has been added in the NUMERICAL section to give the user control of the minimum number of Newton iterations per timestep. MINCYC 2 will overcome this issue.

Example:

```
*DRVDT 1.0d-10 ** stb/scf/day
```

Initial Gas-Oil Ratio (Conditional)

*GORINT

PURPOSE:

*GORINT signals the input of initial gas-oil ratio.

FORMAT:

*GORINT *gor*

DEFINITIONS:

gor

Initial gas-oil ratio used with *MISCG model (m^3/m^3 | scf/STB | cm^3/cm^3).

DEFAULTS:

Conditional keyword. No default value.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *MISCG.

EXPLANATION:

This keyword is used to account for the solution gas dissolved in the oil phase when running with chase- gas injection (*MISCG).

Note that mass conservation of initial solution gas is not accounted for in such simulation, hence it is assumed that pressure will not fall below saturation pressure of the initial solution gas.

The acceptable range of values for initial gas-oil ratio is:

	SI m^3/m^3	Field scf/STB	Lab cm^3/cm^3
min	0.0	0.0	0.0
max	1.0E+3	5,617	1.0E+3

Gas and Solvent Mixing Parameter (Conditional) *OMEGASG

PURPOSE:

*OMEGASG signals the input of gas and solvent mixing parameter.

FORMAT:

*OMEGASG *omega_sg*

DEFINITIONS:

omega_sg

Gas and solvent mixing parameter (fraction).

DEFAULTS:

Conditional keyword. No default value.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *MISCG or *MISNCG.

For models with multiple PVT regions, either a single *OMEGASG keyword can be used for all PVT regions or one *OMEGASG keyword must appear for each PVT region.

EXPLANATION:

When there is free gas (chase gas or original gas- in-place) this parameter determines the mixing of free gas with solvent. The value of *omega_sg* is bounded by 0 and 1 where 0 is the immiscible case and 1 is the completely miscible case as for *omega_os*. Typically *omega_sg* should be greater than or equal to the maximum value of *omega_os* that is entered in the *PVTS table.

Recall, for *MISCG, there is chase gas and the original gas-in-place can only be in solution. If *MISNCG, then there is no chase gas and the original gas-in-place can be in the form of free gas as well as solution gas.

The acceptable range of values for the gas and oil mixing parameter is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

Minimum Solvent Saturation (Conditional)

*MINSS

PURPOSE:

*MINSS signals input of minimum solvent saturation.

FORMAT:

*MINSS *min_sol_sat* (*SGTHRESH *sgval*)
(*SOTHRESH *soval*)
(*SMOOTHEND (*ON|*OFF))

DEFINITIONS:

min_sol_sat Minimum solvent saturation (fraction).

sgval Gas saturation threshold. Loss of miscibility if $Sg > Sgthresh$ and $Ss < Minss$.

soval Oil saturation threshold. Loss of miscibility between $So = SORM + soval$ and $So = SORM - soval$.

*SMOOTHEND

Enable or disable (*ON|*OFF) end point smoothing for loss of miscibility due to MINSS, SGTHRESH and/or SORM. Note SORM, is an optional part of the *SWT table.

DEFAULTS:

Conditional keyword. If *MINSS *min_sol_sat* *SGTHRESH *sgval* *SOTHRESH *soval* found, loss of miscibility smoothing is on, then

1. *MINSS has no default.
2. *MINSS is reset to 0.001 if a value less than 0.001 is entered.
3. *SGTHRESH is reset to 0.001 if a value less than 0.001 is entered.
4. If *SGTHRESH is not found in the data, a default value of 0.01 is applied.
5. If *SOTHRESH is not found in the data, it is set equal to *SGTHRESH

If *MINSS/*SGTHRESH loss of miscibility smoothing is off, then

1. *MINSS has no default.
2. If *SGTHRESH is not found in the data, a default value of 0.01 is applied.
3. *SOTHRESH is not used.

By Default, loss of miscibility end point smoothing (*SMOOTHEND *ON or *SMOOTHEND) is enabled.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *MISCG or *MISNCG.

EXPLANATION:

The minimum solvent saturation is the solvent saturation in the presence of gas below which mixing is not possible.

Sorm (part of the *SWT table) is the saturation of oil where mixing is not possible due to water blocking.

The acceptable range of values for the minimum solvent saturation and also for the gas saturation threshold is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	0.5	0.5	0.5

Polymer Adsorption Table (Conditional)

*PADSORP

PURPOSE:

*PADSORP signals start of polymer adsorption table.

TABLE:

*PADSORP	
<i>p_con</i>	<i>adsorp_level</i>
:	:

DEFINITIONS:

p_con Polymer concentration (kg/m³ | lb/STB | g/cm³).

adsorp_level Polymer adsorption level (kg/m³ | lb/STB | g/cm³).

DEFAULTS:

Conditional keyword. No default values.

This keyword must be in the Component Property keyword group. Required for *POLY.

EXPLANATION:

This is the table input of polymer adsorption as a function of concentration subject to the permeability dependent maximums and residual levels *PPERM as follows:

The *PADSORP table is read in, and *adsorp_level* is normalized by the last value in the table. In the example below, *adsorp_level* would be divided by 0.10 to create a normalized table.

When polymer adsorption is required in the simulation, the polymer concentration in a block is used in the normalized table to interpolate a value of normalized polymer adsorption.

The normalized polymer adsorption is then de-normalized to actual adsorption by multiplying the normalized value by the appropriate *max_ad* value from the *PPERM table.

Each block's maximum permeability (maximum of PERMI, PERMJ and PERMK) is used to determine the block's perm value which is used to interpolate the *max_ad* (and *res_ad*) value (all in the *PPERM table).

After this is done, and the polymer adsorption is determined for each block, the residual adsorption level (*res_ad* in *PPERM) for the block is used to adjust the adsorption level as follows:

If the calculated *adsorp_level* of a block is less than the *res_ad* for that block, we ensure that: if the *adsorp_level* was previously (last timestep) above *res_ad*, it goes no lower than *res_ad* and that if the *adsorp_level* was previously (last timestep) below *res_ad*, it goes no lower than the previous value.

For dual porosity systems, only one table is required.

Example:

```
*PADSORP
** p_con    adsorp_level
 0.0          0.0
 0.125        0.035
 0.25         0.075
 0.37         0.09
 0.50         0.10
```

NOTE:

Normally lab data for adsorption are expressed as mass of polymer adsorbed per mass of reservoir rock. To convert the lab values of adsorption to those required by IMEX multiply lab values by the density of the rock and by the factor

$$(1 - \phi)/\phi$$

The factor

$$(1 - \phi)/\phi$$

converts from rock volume to pore volume. Typical values of rock density vary between 2000 kg/m³ and 2500 kg/m³.

Polymer Viscosity Mixing (Conditional)

*PMIX

PURPOSE:

*PMIX indicates the viscosity mixing model for the polymer solution is being defined.

FORMAT:

*PMIX (*LINEAR)
(*NONLINEAR)
(*TABLE)
(*VELTABLE)

DEFINITIONS:

*LINEAR

Use linear mixing.

*NONLINEAR

Use nonlinear mixing

*TABLE

Use input table to calculate the polymer-water mixture viscosity. A dimensionless, two column table must follow:

<i>relative concentration</i>	<i>viscosity ratio</i>
:	:

The first column is the relative concentration which is the ratio of local polymer concentration to the reference polymer concentration (given by *PREFCONC). The second column is the ratio of polymer viscosity to water viscosity.

*VELTABLE

Use an input table to calculate the polymer-water mixture viscosity as a function of both polymer concentration and mixture velocity. The keyword *VWT followed by a velocity value (*VWT vel) must follow on a separate line. A dimensionless, two column table must follow the velocity value on a separate line. A new occurrence of "*VWT vel" signals the start of another dimensionless table at a new velocity:

*VWT *vel₁*

<i>relative concentration</i>	<i>viscosity ratio</i>
:	:

*VWT *vel₂*

<i>relative concentration</i>	<i>viscosity ratio</i>
:	:

***VWT** *vel_n*

<i>relative concentration</i>	<i>viscosity ratio</i>
:	:

The first column is the relative concentration which is the ratio of local polymer concentration to the reference polymer concentration (given by ***PREFCONC**). The second column is the ratio of polymer-water mixture viscosity to pure water viscosity.

vel₁ ... *vel_n*

Value of reservoir condition velocity. (m/day | ft/day | cm/min | m/day)

DEFAULTS:

Conditional keyword. No default.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with ***POLY**. If ***PMIX *TABLE** is used, the reference polymer viscosity (***PVISC**) becomes unnecessary and will be ignored if it is given. The relative concentration must start with zero and values can be no larger than one.

EXPLANATION:

When ***PMIX *LINEAR** is chosen, the polymer solution is calculated by the equation:

$$vsw(p) = \alpha \cdot visrp + (1 - \alpha) \cdot vsw(p)$$

When ***PMIX *NONLINEAR** is entered, the polymer solution is calculated by the equation:

$$vsw(p) = visrp^\alpha \cdot vsw(p)^{1-\alpha}$$

The variable *visrp* is the reference polymer solution viscosity (***PVISC**) and α is the relative concentration:

$$\alpha = cp / crp$$

Where *cp* is the local polymer concentration and *crp* is the reference polymer concentration (***PREFCONC**).

When ***PMIX *TABLE** is used, the viscosity of polymer solution is given by:

$$vsw(p) = visra(\alpha) \cdot vsw(p)$$

Corresponding to α , the *visra* is the viscosity ratio obtained from linear table interpolation or extrapolation.

When ***PMIX *VELTABLE** is encountered, a two dimensional interpolation is used. Water-polymer mixture velocity is calculated for each grid block. The velocity is divided by grid block porosity and multiplied by a correction factor (***FRWIDTHCOR**), or:

$$Vel_{cor} = FRWIDTHCOR * \text{Reservoir Condition Velocity}/\phi$$

Where *Vel_{cor}* is the corrected velocity used in the VELTABLE table, *FRWIDTHCOR* is the fracture width correction term entered on a block by block basis, Velocity is the average velocity in a block and ϕ is the block's porosity

This reservoir condition velocity (Vel_{cor}) is used to interpolate between the different *VWT tables. The viscosity ratios are then determined by additionally interpolating with respect to the relative polymer concentration.

If a single *VWT table is encountered, the single table is used for all values of velocity. If a velocity is less than the velocity on the first *VWT keyword, the first table will be used. If a velocity greater than the velocity on the last *VWT table, the last table will be used.

If a relative concentration is less than the lowest relative concentration found in a *VWT table, the lowest tabular value will be used. If a relative concentration is greater than the largest relative concentration found in a *VWT table, the largest tabular value will be used. The relative polymer concentration cannot be less than zero or greater than 1.0.

Example:

```
** Velocity Dependent Table at Velocity 0.10, 10.0 and 40.0
ft/day
*PMIX *VELTABLE
*VWT 0.10
0.0      1.00
0.2      1.50
0.6      2.00
1.0      5.00
*VWT 10.0
0.0      1.00
0.2      1.20
0.6      1.75
1.0      3.00
*VWT 40.0
0.0      1.00
0.2      1.10
0.6      1.40
1.0      2.00
```

Reference Polymer Viscosity (Conditional)

*PVISC

PURPOSE:

*PVISC signals the input of reference polymer viscosity.

FORMAT:

*PVISC *value*

DEFINITIONS:

value

Reference polymer solution viscosity (mPa-s | cp | cp).

DEFAULTS:

Conditional keyword. No default value.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *POLY or *POLYOW.

EXPLANATION:

The reference polymer viscosity is the polymer viscosity at the reference concentration (*PREFCONC).

The acceptable range of values for the polymer viscosity is:

	SI mPa·s	Field cp	Lab cp
min	0.5	0.5	0.5
max	10.0	10.0	10.0

Reference Polymer Concentration (Conditional)

*PREFCONC

PURPOSE:

*PREFCONC signals the input of reference polymer concentration.

FORMAT:

*PREFCONC *value*

DEFINITIONS:

value

Reference polymer concentration. It must be greater than or equal to the polymer injection concentration (kg/m^3 | lb/STB | g/cm^3)

DEFAULTS:

Conditional keyword. No default value.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *POLY or *POLYOW.

EXPLANATION:

The reference concentration is used to weight the polymer concentration for the viscosity mixing rule. (See equations E.6 and E.7 in Appendix E).

The value for *PREFCONC must be greater than zero since it is used to calculate the numerical shift, the tolerance (for Newtonian iteration), and the threshold for the threshold switching criteria.

The acceptable range of values for the polymer concentration is:

	SI kg/m^3	Field lb/STB	Lab g/cm^3
min	0.0	0.0	0.0
max	29.0	10.0	0.029

Polymer Permeability Table (Conditional)

*PPERM

PURPOSE:

*PPERM signals the start of table of absolute permeability dependent polymer properties.

TABLE:

*PPERM				
<i>perm</i>	<i>max_ad</i>	<i>res_ad</i>	<i>p_pore</i>	<i>rrf</i>
:	:	:	:	:

DEFINITIONS:

perm Absolute permeability (md | md | md)

max_ad Maximum adsorption capacity (kg/m³ | lb/STB | g/cm³).

res_ad Residual sorption level (kg/m³ | lb/STB) | g/cm³).

p_pore Polymer accessible pore volume (fraction).

rrf Residual resistance factor, >= 1.0.

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Component Property keyword group. Required with *POLY or *POLYOW.

EXPLANATION:

The maximum adsorption capacity is the capacity of the rock to absorb the polymer. The residual sorption level is the amount of polymer left in the rock after injected water washes some of the polymer away from the rock.

Example:

```
*PPERM
**  perm    max_ad   res_ad   p_pore   rrf
    10.0     0.3      0.15     0.95     1.2
  1000.0    0.2      0.10     1.0      1.2
```

Reference Seawater Viscosity (Conditional)

*SVISC

PURPOSE:

*SVISC signals the input of the reference Seawater viscosity.

FORMAT:

*SVISC *value*

DEFINITIONS:

value

Reference Seawater viscosity (mPa-s | cp | cp).

DEFAULTS:

Conditional keyword. Default is formation water viscosity of PVT region 1.

CONDITIONS:

This keyword must be in the Component Property keyword group. Only used with *BLACKOIL_SEAWATER or *OILWATER_SEAWATER Model.

EXPLANATION:

The reference seawater viscosity is the viscosity of 100% seawater at pressure *REFPW.

In the Seawater model, both the reference pressure *REFPW and the pressure dependence of Seawater viscosity *CVW are the values entered in PVT region 1.

Pure seawater viscosity (v_{sw}) is calculated as:

$$v_{sw} = v_{swi} + cvw \cdot (p - prw)$$

Where v_{swi} is the reference seawater viscosity enters via *SVISC, cw is the pressure dependence of water viscosity with respect to pressure entered via *CVW (for PVT region 1) and prw is the reference pressure of water entered via *REFPW (for PVT region 1)

The mixed seawater/formation water viscosity (vw) is calculated as:

$$vw = \alpha \cdot v_{sw} + (1 - \alpha) \cdot v_{fw}$$

Where α is the seawater volume fraction and v_{fw} is the formation water viscosity

The acceptable range of values for the seawater viscosity is:

	SI mPa·s	Field cp	Lab cp
min	0.5	0.5	0.5
max	10.0	10.0	10.0

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 adsorbed onto reservoir rock at a particular pressure. Adsorption is modeled using equations based on the Langmuir model (See *ADGMAXV).

FORMAT:

*ADS_WATERZONE (*ON)
 (*OFF)

DEFAULTS:

Optional keyword. Default is *OFF.

CONDITIONS:

This keyword must be in the Component Property keyword group. Only used with the adsorption model, which is turned on by the appearance of the keywords *ADGMAXV and *ADGCSTV in the Rock-Fluid data section.

EXPLANATION:

The basic Langmuir Isotherm for gas adsorption used in IMEX is:

$$Ads = (1 - \phi) \times ROCKDEN \times ADGMAXV \times \left(\frac{ADGCSTV \times P_{AD}}{1 + ADGCSTV \times P_{AD}} \right)$$

Where P_{AD} = block pressure, if the block pressure is less than or equal to ADGPCRIT and P_{AD} = ADGPCRIT, if the block pressure is greater than ADGPCRIT.

If ADS_WATERZONE is specified in the Component Properties section, the basic Langmuir equation in each block is multiplied by the gas saturation in a block, so that $Ads = Ads \times S_g$. This ensures that there is no gas adsorbed onto the rock in the water zone.

Rock-Fluid Data

Notes on Rock-Fluid Data

1. Multiple rock types are allowed in IMEX. Rock types are assigned to grid blocks using *RTYPE. Relative permeabilities of each rock type can be entered using tables.

The following keywords are used:

*RPT rock_type_number

followed by the tabular values for that rock type introduced by

*SWT and *SGT or *SLT.

2. Hysteresis effects are allowed on the oil-water capillary pressure, Pcow, the oil-gas capillary pressure, Pcog, the gas phase relative permeability, krg and the oil phase relative permeability, krow.

Capillary pressure hysteresis is modelled using an approach similar to that of Killough, 1976. To invoke Pcow hysteresis enter both the drainage and imbibition Pcow table columns and optionally epspc using *EPSPC. To invoke Pcow hysteresis enter both the drainage and imbibition Pcow table columns and optionally epspc using *EPSPCG. To invoke krg hysteresis enter the keyword *HYSKRG. To invoke krow hysteresis enter the keyword *HYSKRO.

Oil-water hysteresis can be modeled to couple capillary pressure hysteresis to relative permeability hysteresis which produces coupled, consistent, end-points for Pcow, Krow and Krw. To invoke the coupled oil-water hysteresis option, the user can either define the imbibition Pcow in the SWT table or define the SWTI table. HYSKRO and HYSKRW are used to assign the calculation method for Krow and Krw hysteresis. The keyword *EPSPC is still used when creating Pcow scanning curves.

3. Relative permeability tables may be scaled to different end points for each grid block. This is done by entering one or more of the following end points for each grid block.

SWCON - The connate water saturation

SWCRIT - The critical water saturation

SGCON – The connate gas saturation

SGCRIT - The critical gas saturation

SOIRW – Irreducible oil saturation (oil-water table)

SLCON – The connate liquid saturation (gas-liquid table)
SOIRG – The irreducible oil saturation (gas-liquid table) when
SCALING-STARS is used
SORW – The residual oil saturation (water-oil table)
SORG - The residual oil saturation (gas-liquid table)
SORMAX - The maximum residual oil saturation (water-oil table, imbibition)
KRWIRO - Water rel. perm. at irreducible oil (oil-water table)
KROCW – Oil rel. perm. at connate water (oil-water table)
KRGCL – Gas rel. perm at connate liquid (liq.-gas table)
KROGCG – Oil rel. perm at connate gas (liq.-gas table)
KRWRO – Water rel. perm. at residual oil (oil-water table)
KROCRW – Oil rel. perm. at critical water (oil-water table)
KRGRL – Gas rel. perm at residual liquid (liq.-gas table)
KROGCRG – Oil rel. perm at critical gas (liq.-gas table)
PCWMAX – Oil water capillary pressure at connate water
PGGMAX – Gas oil capillary pressure at connate liquid
JFWMAX – Oil water J function at connate water
JFGMAX – Gas-oil J function at connate liquid

4. Rock wettability may be changed by using the *OILWET keyword. The *OILWET keyword makes the rock oil wet and water becomes the phase with intermediate wettability.
5. When three phases are present the default in IMEX is to evaluate Kro using Stone's second model as normalized by Aziz and Settari. The Krow and Krog values which enter the Stone's second model formula are 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 Linear Isoperm Method and a segregated model. 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.
6. For relative permeabilities, smoothing is generally done for kr close to zero. Various smoothing options are available for these cases. The options may be specified using *SMOOTHEND and related keywords. The default is quadratic smoothing between kr=0 and the first nonzero table entry for kr.
7. Options are available to ensure smoothness and remove inflection points in the relative permeabilities and capillary pressures. These options are defaulted off. If poor timestep behavior occurs with consistently many Newton iterations per timestep and/or many timestep cuts, these options may improve performance. Use of these options may alter the shape of the relative permeability and capillary pressure curves.

8. The smoothing option discussed above can be also used in situations where actual rock data is sparse to generate curves which produce a best fit to analytical power law functions.
9. J Functions can be input in place of capillary pressures. See the keywords *JFUNC, *SRFTNW and *SRFTNG for details.
10. Non-Darcy flow in the reservoir can be modelled by using the keywords *NONDARCY and *NDARCYCOR. See these keywords for details. Non-Darcy flow in the well is modelled using the *TURB option on the *PERF or *PERFV keyword.

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:

Three phase relative permeabilities are computed using modified Stone's Model II (by default).

The relative permeability of water in a three phase system is the same as that in a two phase water-oil system, and is a function of S_w (water 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 S_g (gas saturation) only. Four methods are available for calculating the three phase oil relative permeability, k_{ro} . These are: Stone's method 1; Stone's method 2 as modified by Settari and Aziz (Aziz, K., and Settari, A., "Petroleum Reservoir Simulation", Applied Science Publishers Ltd., London, 1979); the Linear Isoperm method and the Segregated method.

The two phase relative permeability curves are entered in tabular form. The various saturation end points, such as connate water, are determined internally by scanning the krw table entries. Hysteresis effects upon water-oil capillary pressure, oil-gas capillary pressure, gas relative permeability and oil ($krow$) relative permeability can be included.

Relative Permeability Curves by Table (Required)

*RPT

PURPOSE:

*RPT indicates that this set of relative permeability curves will be defined by table entries.

FORMAT:

```
*RPT set_number (*OILWET)
      (*DRAINAGE |
       *IMBIBITION (*BOTH | *PCOW | *PCOG) )
      (*SCALING-STARS)
```

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.

*OILWET

Keyword indicating that the wetting phase is oil. If not present, the rock type is assumed to be water wet. Refer to the tutorial section for a further discussion of the oil wet option used in IMEX.

*DRAINAGE, *IMBIBITION

Hysteresis subkeywords for capillary pressure -- if *DRAINAGE is specified the drainage capillary pressure curve for both Pcow and Pcog is used initially; if *IMBIBITION is specified the imbibition curve for both Pcow and Pcog 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 gas and oil relative permeability. For the gas and oil relative permeability the drainage curve is always used initially.

*PCOW

Subkeyword 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

Subkeyword 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

Subkeyword 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 subkeyword

***SCALING-STARS**

This keyword allows the user to use the endpoint scaling option which is available in STARS. (1) This model does not use *SLCON as an independent endpoint. Instead *SOIRG is used. Input of *SOIRG in IMEX also triggers the use of this option. (2) In addition, IMEX duplicates the feature in STARS which sets connate and critical endpoints equal if one of the two endpoints is not read in and the table derived values are identical. Without *SCALING-STARS, the omitted endpoint is always read from the *SWT or *SLT/*SGT table.

The *NOSWC option cannot be used with *SCALING-STARS.

In the IMEX endpoint scaling option, the input of an *SWCON array would not change the value of *SLCON instead *SOIRG would be altered to maintain the original value of *SLCON. When *SCALING-STARS is enabled *SOIRG is kept constant and *SLCON is allowed to change. This duplicates the behavior in the STARS simulator.

In the IMEX endpoint scaling option, when an endpoint is not entered as an array, it is read for all blocks from the appropriate *RPT table, regardless of its value.

In the STARS scaling option, this is not the case for connate and critical/irreducible saturations when these values are identical in the relative permeability tables.

For example, if *SWCON is input and *SWCRIT is not input for a gridblock, and in the table active for that gridblock, the table value of Swc and Swcon are identical, *SWCRIT is set equal to *SWCON. If *SWCRIT is input and *SWCON is not input, *SWCON is set equal to *SWCRIT. In other words we assume that the user only really ever wanted one endpoint instead of two.

Of course if two endpoint arrays are read in (for example the user input both *SWCON and *SWCRIT) we override this behavior.

Similarly the same behavior occurs for *(SGCRIT - *SGCON), (*SOIRW - *SORW), and (*SORG - *SOIRG) pairs.

DEFUALTS:

Required keyword. No default. If not otherwise specified, the *RPT table is assumed to be a drainage table for water wet rock.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group. It should be immediately 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:

When three phases are present the default in IMEX is to evaluate Kro using Stone's second model as normalized by Aziz and Settari. The other models available are Stone's first model, the Linear Isoperm Method and a segregated model. Please see the entry under the *KROIL keyword for more information. These models calculate the three phase relative permeability curves from the two phase curves that are input.

There must be both a water-oil relative permeability table and a liquid-gas relative permeability table immediately following this card except when using the OILWATER option, the OILWATER option does not require the input of a gas-liquid table.

Set_numbers are used to assign different relative permeability curve sets to different rock types 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.

Each *RPT keyword must be immediately followed by a *SWT keyword and table, then a *SLT or *SGT keyword and table. Rock type dependent keywords are input after the SWT and SLT (*SGT) tables, they apply to the rock type defined above. After a new *RPT keyword is input a new set of tables and rock type dependent keywords can be read in.

The *OILWET option specifies oil as the wetting phase and water as the intermediate wetting phase. Thus for this option the water phase pressure is higher than the oil phase pressure.

Note that the *SCALING-OLD option is no longer supported.

Water-Oil Relative Permeability Table (Conditional) *SWT

PURPOSE:

*SWT indicates the start of the water-oil relative permeability table. In addition *SWT controls the use of a wide variety of relative permeability regression and optimization functions.

In its most basic form (to define the beginning of a table):

TABLE:

```
*SWT (*SMOOTHEND 'Subkeywords') (*SWTKRTHR krthr_value)
Sw      krw      krow      (Pcow)   (Pcowi | Sorm)
:       :       :       :       :
```

*SWT subkeywords provide the user with a number of unique features which can:

1. Change the shape of the curves as they go to zero.
2. Create new curves obtained from nonlinear regression of available data.
3. Allow the user to interact with the regression procedure to specify the range over which regression is performed.
4. Regenerate equally spaced tables using a user defined Interpolation interval.

See below for its most complete form:

DEFINITIONS:

(Basic Form)

TABLE:

```
*SWT (*SMOOTHEND 'Subkeywords') (*SWTKRTHR krthr_value)
Sw      krw      krow      (Pcow)   (Pcowi | Sorm)
:       :       :       :       :
```

End-Point Smoothing Options:

*SMOOTHEND

Optional keyword indicating what type of smoothing is to be used for the interval with kr=0 and kr>0 for krw and krow. These are the table intervals where water or oil has just become mobile. The keyword may be followed by a subkeyword as indicated below.

Subkeywords of the *SMOOTHEND Keyword:

*OFF | *ON | *QUAD | *CUBIC

If *SMOOTHEND is not specified or it has no subkeyword, the default is *SMOOTHEND *ON. This is equivalent to *SMOOTHEND *QUAD.

For the subkeyword *QUAD, a quadratic smoothing is used. For *CUBIC, a cubic smoothing is used. For *OFF, no smoothing (that is, linear interpolation) is used.

Critical End-Point Threshold Option:

*SWTKRTHR

Optional keyword to use krthr_value as a threshold value for krw and krow for end-point determination.

krthr_value

If a tabular value for *krw* or *krow* is less than or equal to *krthr_value* then it is set to zero. If *SWTKRTHR is not specified, then a default value of 5.0E-07 is used for *krthr_value*. Entered values of *krthr_value* are reset to lie within the range 5.0E-16 to 5.0E-07.

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.

Pcow

Water-oil capillary pressure (kPa | psi | kPa | kg/cm²) 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. This column is required if *Pcowi* or *Sorm* are to be entered i.e. the fourth column is always assumed to be *Pcow*.

Pcowi

Water-oil capillary pressure (kPa | psi | kPa | kg/cm²) 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. Hysteresis effects CANNOT be modelled when using the pseudo-miscible option.

Sorm

Irreducible oil saturation (fraction) due to water blocking at the given water saturation. (Only required for pseudo-miscible option.) There is no default value. If the pseudo-miscible option is used then the fifth column is assumed to be *Sorm* and capillary pressure hysteresis effects are not modelled.

DEFAULTS:

Conditional keyword. There are no defaults.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group, immediately after the *RPT keyword. *SWT is required with *RPT.

EXPLANATION:

Capillary pressure curves determine the height of the transition zones calculated by the *VERTICAL option. Nonzero capillary pressure curves also influence the width of the transition zones near sharp saturation fronts. This front widening mechanism tends to have a stabilizing effect during the displacement of less mobile fluids (oil) by more mobile fluids (water). The water-oil capillary pressure is defined as

$$P_{cow} = P_o - P_w$$

Since water is assumed to be the wetting phase, the capillary pressure should be monotonically declining with increasing water saturation. However, the P_{cow} value at $S_w=1$ can be nonzero. Capillary pressure need not be entered, unless you are using the pseudo-miscible option with non-zero S_{orm} (and even then P_{cow} can be explicitly defined as 0.0). Otherwise the default is zero.

Entries must be in order of increasing water saturation. The first table entry must be for $S_w = S_{wcon}$ (i.e. 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 8 in Appendix D).

THE *OILWET OPTION:

When the oil wet option is used the water-oil relative permeability table is altered. The S_w column contains the wetting-phase saturation (oil saturation). The K_{rw} column contains the wetting-phase relative permeability (oil relative permeability). The K_{row} column contains the nonwetting phase relative-permeability (water relative permeability). The P_{cow} column now contains the capillary pressure between the wetting and nonwetting phases. Thus for the oil wet system:

$$P_{water} = P_{oil} + P_{cow}$$

where:

P_{water} = water phase pressure

P_{oil} = oil phase pressure

P_{cow} = capillary pressure entered in the P_{cow} column.

By contrast for a water wet system:

$$P_{water} = P_{oil} - P_{cow}$$

Note that the values entered in the P_{cow} column are normally positive as the capillary pressure is still the positive difference in pressure between non-wetting water and wetting oil. Please refer to the tutorial section for a further discussion of the oil wet option used in IMEX.

THE *GASWATER OPTION:

When using the Gas Water option the *SWT table is only altered slightly. K_{row} is not read in, the P_{cow} column (if present is read as P_{cgw}) and the P_{cowi} column is not read. Thus a typical Gas Water *SWT table would be:

***SWT**

<i>Sw</i>	<i>krw</i>	<i>(Pcgw)</i>
:	:	:

It is important to ensure that the final value in the *SWT table, when using the *GASWATER option is *Sw* = 1.0, as the irreducible oil saturation must be 0.0. *Pcgw* is defined as:

$$Pcgw = Pg - Pw$$

THE *GASWATER_WITH_CONDENSATE OPTION:

When using the *GASWATER_WITH_CONDENSATE model, the *Pcow* column, if present, is read as *Pcgl* (capillary pressure between gas and liquid). If condensate appears in the reservoir as liquid its saturation is added to the water saturation when *Pcgl* is determined. For the condensate option, the output of *Pcgl* is sent to the *Pcog* output array (both for ascii and SR2 output). The *Pcow* output array is zero. The condensate option initializes the reservoir in fashion identical to the *GASWATER option. The *VOLATILE_OIL option does not alter the meaning of *Pcow*, capillary pressures are treated normally and initialization is handled in a fashion similar to the black oil model.

The acceptable range of values for water saturation is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

The acceptable range of values for water relative permeability is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

The acceptable range of values for oil relative permeability is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

The acceptable range of values for capillary pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	-1.0E+5	-14500.0	-1.0E+5	-1.0E+3
max	1.0E+5	14500.0	1.0E+5	1.0E+3

The acceptable range of values for irreducible oil saturation is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

General Form of the *SWT keyword:

TABLE:

```
*SWT (*SMOOTHEND (*QUAD | *CUBIC | *POWERQ |
                     *POWERC | *OFF | *ON))
(*SMOOTH (*ALL | *ALLPL | *ALLINT | *ALLIN |
           *NONE | *OFF | property_list)
(*PLAW pkrw pkrow ppcowd (ppcowi | plsorm))
(*DSWTI dswti_value) (*SWTKRTHR krthr_value)
  Sw      krw      krow      (Pcow      (Pcowi | Sorm))
  :       :       :       :       :
```

Uses of the General Form:

Definitions of individual subkeywords will follow, but it is useful to first highlight a few of the many tasks the *SWT keyword can perform in situations where rock data is sparse and the user wants to generate meaningful curves.

1. Using sparse data, generate curves which produce the best fit to an analytical power-law model over the entire range of data.
*SWT *SMOOTH *ALL or SWT *SMOOTH *ALLPL
2. Using Sparse data, generate curves which produce the best fit to an analytical power-law model over a smaller range of data which ensures matching of the original data at critical points and when Pcow -> 0.0 (or becomes constant).
*SWT *SMOOTH *ALLINT
3. Using no data, specify power law curves explicitly using input power law exponents and a two row relative perm. table (to define saturation end points and relative permeability end points)
*SWT *SMOOTH *PLAW water_exponent‘oil_exponent pcow_exponent’
4. Re-interpolate an existing table using equal spacing of approx. 0.01 to take advantage of faster table look-up procedures.
*SWT *SMOOTH *ALLIN *DSWTI 0.01
5. Alter the way the interpolation is performed in the table interval between kr=0 and the first nonzero entry for kr. For Relative permeability, this is at Swcrit (for Krw) and at Sorw (for Krow).
*SWT *SMOOTHEND *CUBIC

DEFINITIONS:

(General Form which includes all Regression Options)

The *SWT keyword can be subdivided into 5 subkeywords

- *SMOOTHEND - Smooth End-points of tables
- *SWTKRTHR - Water Table Critical End-point Threshold
- *SMOOTH- Regression Options
- *DSWTI - Table Regeneration Options
- *PLAW - Direct Power Law Input (Regression not Used)

*SWT (*SMOOTHEND ‘subkeywords’)
 (*SWTKRTHR *krthr_value*) (*SMOOTH‘subkeywords’)
 (*DSWTI *dswti_value*)
 (*PLAW ‘subkeywords’)

End-Point Smoothing Options:

***SMOOTHEND**

Optional keyword indicating what type of smoothing is to be used for the interval with kr=0 and kr>0 for krw and krow. These are the table intervals where water or oil has just become mobile. The keyword may be followed by a subkeyword as indicated below.

Subkeywords of the *SMOOTHEND Keyword:

*OFF | *ON | *QUAD | *CUBIC | *POWERQ | *POWERC

If *SMOOTHEND is not specified or it has no subkeyword, the default is *SMOOTHEND *ON. This is equivalent to *SMOOTHEND *POWERQ.

For the subkeyword *QUAD, a quadratic smoothing is used. For *CUBIC, a cubic smoothing is used. For *OFF, no smoothing (that is, linear interpolation) is used.

For *POWERQ and *POWERC end point smoothing is done using power law if the exponent is between 1.5 and 4.0. Otherwise, for *POWERQ, quadratic is used and for *POWERC, cubic is used for end point smoothing.

If the *SMOOTH regression option is not used, *POWERC And *POWERQ default to *CUBIC and *QUAD respectively.

Critical End-Point Threshold Option:

***SWTKRTHR**

Optional keyword to use *krthr_value* as a threshold value for *krw* and *krow* for end-point determination.

krthr_value

If a tabular value for *krw* or *krow* is less than or equal to *krthr_value* then it is set to zero. If *SWTKRTHR is not specified, then a default value of 5.0E-07 is used for *krthr_value*. Entered values of *krthr_value* are reset to lie within the range 5.0E-16 to 5.0E-07.

Regression Options:

***SMOOTH**

Perform regression on the relative permeability table entries and generate an equally spaced table which best fits the data over the specified range.

Valid Regression Option’subkeywords of the *SMOOTH Keyword are:

*ALL, *ALLPL, *ALLINT, *ALLLIN, *NONE, *OFF and ‘*property_list*’

If the *ALL subkeyword is used, or no subkeywords are Found, then all properties in the table are smoothed according to default actions. This tries power law first. For kr's, if the resultant power is greater than 4.0 then linear is used for that property. The smoothed ranges for *ALL are as follows:

Property Range

Krw	Swcrit to 1.0-Sorw
Krow	Swcon to 1.0-Sorw
Pcow	Swcon to Sw(Pcow=0.0)
Pcowi	Swcon to Sw(Pcow=0.0)
Sorm	Swcon to Max Sw in Table

where

Swcrit = Critical Saturation
Swcon = Connate Saturation
Sw(Pcow=0.0) = Lowest Sw in table with Pcow constant

***ALLPL**

Specifies that Power law is used for all properties in the table. If powers > 4.0 are calculated, they are used.

***ALLINT**

Specifies that Power law is used between a reduced range of end points the smoothed ranges for *ALLINT are as follows

Property Range

Krw	Swcrit to 1.0-Sorw
Krow	Swcrit to 1.0-Sorw
Pcow	Swcrit to Sw(Pcow=0.0)
Pcowi	Swcrit to Sw(Pcow=0.0)
Sorm	Swcrit to Max Sw in Table

where

Swcrit = Critical Saturation
Sw(Pcow=0.0) = Lowest Sw in table with Pcow constant

***ALLLIN**

Generates an evenly spaced table using linear interpolation for all properties.

If the *NONE or *OFF keyword is used then none of the table properties are smoothed. This is the default action if the *SMOOTH keyword is not specified.

property_list allows the user to customize how regression is applied to a list of properties.

For a *SWT table, valid properties to be regressed include:

*KRW	*KRWPL	*KRWINT	*KRWLIN
*KROW	*KROWPL	*KROWINT	*KROWLIN
*PCOW	*PCOWPL	*PCOWINT	*PCOWLIN
*PCOWI	*PCOWIPL	*PCOWIINT	*PCOWILIN
*SORM	*SORMLPL	*SORMINT	*SORMLIN

If *property_list* is used and a particular, existing property is not specified, then linear interpolation is used for that property.

The conventions for the above properties are the same as for *ALL, *ALLPL and *ALLINT and *ALLLIN.

Table Regeneration Option:

*DSWTI

When the Regression options (*SMOOTH) are not used IMEX does not regenerate equally spaced rel. perm. tables, i.e.:

Equally spaced tables not generated when,

1. the *SMOOTH keyword is not found,
2. the *SMOOTH *NONE keyword is found,
3. the *SMOOTH *OFF keyword is found.

When regression is employed, IMEX automatically creates equally spaced tables for the newly created curves.

The *DSTWI keyword can be used to define the number of points in the table.

NOTE:

*SMOOTH *ALLLIN and *DSWTI *dswti_value*

Can be used together to generate new equally spaced tables without using regression on table values. This can speed up relative permeability calculation times as equally spaced tables are interpolated more efficiently.

dswti_value

Generate an evenly space table with deltaSw approximately equal to *dswti_value*. Calculate deltaSw as follows:

$$\text{deltaSw} = (1 - \text{Swc} - \text{Sorw}) / \text{int}((1 - \text{Swc} - \text{Sorw})/\text{dswti} + \text{eps})$$

using the input *dswti_value* and where int(*x*) is the truncated integer value of *x*.

If the *DSWTI keyword is not specified, the default for *dswti_value* is calculated internally.

The minimum of half of the minimum table spacing and 0.016 is calculated:

1. If this value is greater than 0.001 then *dswti_value* is set to it.
2. If this value is less than 0.001 then *dswti_value* is set to 0.001.

Direct Input of Power Law Coefficients:

***PLAW**

Input power law exponents for table properties for cases with only end points input.

pkrw

Power law exponent for krw for cases with only end points input. A value of 0.0 results in a default value of 1.0.

pkrow

Power law exponent for krow for cases with only end points input. A value of 0.0 results in a default value of 1.0.

ppcowd

Power law exponent for Pcowd for cases with only end points input. A value of 0.0 results in a default value of 1.0. If it is not specified, then it is assumed that Pcowd is zero.

ppcowi

Power law exponent for Pcowi for cases with only end points input. A value of 0.0 results in a default value of 1.0. If it is not specified, then it is assumed that Pcowi is zero.

psorm

Power law exponent for Sorm for cases with only end points input. A value of 0.0 results in a default value of 1.0. If it is not specified, then it is assumed that Sorm is zero.

EXPLANATION:

In the regression option, power law or linear interpolation may be used for krw and krow and, if they exist, for Pcow and Pcowi. If *SMOOTH keyword is omitted, or *SMOOTH *NONE or *OFF is specified, no smoothing is done. Otherwise, the keywords

```
*SMOOTH ( *ALL | *ALLPL | *ALLINT | *ALLLIN |
           *NONE | *OFF | property_list )
```

determine for which properties power law and for which properties linear interpolation are to be used.

If *PLAW is specified, then all properties are assumed to be power laws using the specified exponents. End points and maximum values are determined from the following table. All other values, if they exist are ignored.

If *PLAW is not specified and power law correlations are specified, then there must be at two Sw's given such that the particular property is nonzero. The power for that property is then calculated using nonlinear least squares.

If the *SMOOTH keyword is specified, an equally spaced table is generated. Using a power law or linear interpolation as specified by its subkeywords. Within the simulator, this equally spaced table is used. Relative permeabilities are calculated using linear interpolation from this internal table except over the interval between kr=0 and kr>0. On this interval, the type of smoothing, if any, is controlled by the *SMOOTHEND keyword and optionally one of its subkeywords.

Examples:

- a) Generate power law correlations for Krw, Krow and Pcow using least squares fits of the following table:

*SWT	*SMOOTH			
** sw	krw	krow	pcow	
0.178350	0.0	0.991590	27.930	
:	:	:	:	
0.881490	0.490000	0.000000	-2.75	

- b) Generate power law correlations for Krw, Krow and Pcow (drainage) matching endpoints and maximum values for a table where Swcrit = Swcon and Sorw = Soirw:

1. Input the power law exponents for Krw, Krow and Pcowd on the *SWT keyword.

***SWT *SMOOTH *PLAW pkrw pkrow ppcowd**

2. Input the endpoints and values of Krwi, Krowc and minimum and maximum values of drainage oil water capillary pressure in the two row table.

Swcrit = Swcon	0.0	Krowc	Pcowmax
1 - Sorw	Krwi	0.0	Pcowmin

- c) Generate power law correlations for Krw, Krow and Pcow (drainage) matching endpoints and maximum values for a table where Swcrit > Swcon and Sorw = Soirw:

1. Input the power law exponents as in B above.

2. Input the endpoints and values of Krwi, Krowc and minimum and maximum values of oil water capillary pressure in the three row table.

Swcon	0.0	Krowc	Pcowmax
Swcrit	0.0	*INT	*INT
1 - Sorw	Krwi	0.0	Pcowmin

Note: Any value may be used instead of *INT, as Krow and Pcow at Sw = Swcrit are ignored When using *PLAW.

For the above (B and C), assuming *SMOOTH *ALL (the default) or *SMOOTH *ALLPL, we obtain:

$$Krw = Krwi \left(\frac{Sw - Swcrit}{1 - Swcrit - Sorw} \right)^{Pkrw}$$

$$K_{row} = K_{row}^W \left(\frac{1 - S_{orw} - S_w}{1 - S_{wcon} - S_{orw}} \right)^{P_{krow}}$$

$$P_{cow_D} = (P_{cow}(S_w = S_{wcon}) - P_{cow}(S_w = 1 - S_{orw})) \left(\frac{1 - S_{orw} - S_w}{1 - S_{wcon} - S_{orw}} \right)^{P_{pcowd}} \\ + P_{cow}(S_w = 1 - S_{orw})$$

Water-Oil Imbibition Relative Permeability Table *SWTI

PURPOSE:

*SWTI indicates the start of the water-oil imbibition relative permeability table.

TABLE:

*SWTI	(*KROWI_ONLY *KRWI_ONLY)	
<i>Sw</i>	<i>krowi</i>	<i>krwi</i>
:	:	:

SWTI table inherits parameters of subkeywords *SMOOTHEND and *SWTKRTHR from SWT.

SWTI table does not perform the *SMOOTH regression option supported by SWT table.

DEFINITIONS:

*KROWI_ONLY

Optional keyword to indicate the *SWTI table will include only Sw and Krowi (two columns).

*KRWI_ONLY

Optional keyword to indicate the *SWTI table will include only Sw and Krwi (two columns).

If *SWTI is not followed by a subkeyword, three columns are required - Sw, Krowi and Krwi.

Sw

Water saturation (fraction). There is no default value.

krowi

Imbibition relative permeability to oil (fraction) in the presence of the given water saturation. There is no default value.

krwi

Imbibition relative permeability to water (fraction) at the given water saturation. There is no default value.

DEFAULTS:

Conditional keyword. There are no defaults.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group. An *SWTI table can be defined for each rock type. If defined, the *SWTI table must appear after the *SWT table.

The water saturation column, Sw, must start from the connate water saturation, Swcon, the same first point of Sw on the *SWT table. Sw takes ascending points and must end at one minus the maximum residual oil saturation, 1-Sormax.

The Krowi column must start with the same value as the drainage Krow column of the *SWT table and must end with zero. Between the start and end points, Krowi values must be less than Krow.

The Krwi column must start with zero and can take values greater or less than drainage Krw of the *SWT table; however, imbibition and drainage curves should never cross. When the rock type does not trap oil, i.e. Sormx equals Sorw, the Krwi and Krw must meet at 1-Sorw; otherwise, the two curves should not meet at 1-Sormax.

EXPLANATION:

Defining the *SWTI table invokes relative permeability hysteresis. The *SWTI table also defines the value of Sormax. The Sormax defined must be consistent with the Sormax indicated by the imbibition capillary pressure Pcowi on *SWT table. When Sormax is given by table (*SWTI), the Sormax entered using the keyword *HYSKRO is ignored.

When *SWTI is defined, *HYSKRO and *HYSKRW can be used to assign the calculation method. Note that although *SWTI table may be entered with *KRWI_ONLY, oil relative permeability Krow hysteresis can still be activated with the Carlson method if the Sormax given is greater than Sorw.

The *SWTI table does not work when the *OILWET option is entered as a subkeyword of *RPT.

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.

In addition *SLT(*SGT) controls the use of a wide variety of relative permeability regression and optimization functions.

In its most basic form (to define beginning of table).

TABLE:

*SLT (*NOSWC) (*SMOOTHEND 'Subkeywords') (*SLTKRTHR krthr_value)

Sl	krg	krog	(Pcog (Pcogi))
:	:	:	:

*SGT (*NOSWC) (*SMOOTHEND 'Subkeywords') (*SLTKRTHR krthr_value)

Sg	krg	krog	(Pcog (Pcogi))
:	:	:	:

*SLT(*SGT) subkeywords provide the user with a number of unique features which can:

1. Change the shape of the curves as they go to zero.
2. Create new curves obtained from nonlinear regression of available data.
3. Allow the user to interact with the regression procedure to specify the range over which regression is performed.
4. Regenerate equally spaced tables using a user defined Interpolation interval.

See below for its most complete form:

DEFINITIONS:

*NOSWC

Keyword indicating that the total liquid residual saturation does not include the connate water saturation.

If the *NOSWC option is used, it is assumed that the connate water saturation is not part of the measured residual liquid saturation in the liquid-gas table. Hence, Sorg is set equal to the residual liquid saturation obtained from the *SLT (*SGT) table and Slcon equals irreducible oil (Soirg).

By default, it is assumed that the residual liquid saturation in the liquid-gas table was measured in the presence of connate water. Hence, Sorg is obtained by subtracting off Swcon from the residual liquid saturation obtained from the *SLT (*SGT) table. Slcon is comprised of connate water (Swcon) plus irreducible oil (Soirg). See the figures in the *SWCON definition for more details. The *NOSWC option cannot be used with *SCALING-STARS.

End-Point Smoothing Options:

***SMOOTHEND**

Optional keyword indicating what type of smoothing is to be used for the interval with kr=0 and kr>0 for krg and krog. These are the table intervals where gas or oil has just become mobile. The keyword may be followed by a subkeyword as indicated below.

Subkeywords of the *SMOOTHEND Keyword:

***OFF | *ON | *QUAD | *CUBIC**

If *SMOOTHEND is not specified or it has no subkeyword the default is *SMOOTHEND *ON. This is equivalent to *SMOOTHEND *QUAD.

For the subkeyword *QUAD, a quadratic smoothing is used. For *CUBIC, a cubic smoothing is used. For *OFF, no smoothing (that is, linear interpolation) is used.

Critical End-point Threshold Option:

***SLTKRTHR**

Optional keyword to use krthr_value as a threshold value for krg and krog for end-point determination.

krthr_value

If a tabular value for krg or krog is less than or equal to krthr_value, then it is set to zero. If *SLTKRTHR is not specified, then a default value of 5.0E-07 is used. Entered values of krthr_value are reset to lie within the range 5.0E-16 to 5.0E-07.

Sl

Total liquid saturation (fraction).

Sg

Gas saturation (fraction).

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.

Pcog

Gas-oil capillary pressure (kPa | psi | kPa | kg/cm²). 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 | kg/cm²). 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. *Pcog* hysteresis cannot be modelled when using the pseudo-miscible option.

DEFAULTS:

Conditional keyword. There are no defaults.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group, immediately after the *SWT keyword and table. Either *SLT or *SGT are required with *RPT.

EXPLANATION:

Capillary pressure curves determine the height of the transition zones calculated by the *VERTICAL option. Nonzero capillary pressure curves also influence the width of the transition zones near sharp saturation fronts. This front widening mechanism tends to have a stabilizing effect during the displacement of less mobile fluids (oil) by more mobile fluids (gas).

The gas-oil capillary pressure is defined as

$$Pcog = Pg - Po$$

Since liquid is assumed to be the wetting phase, the capillary pressure should be monotonically declining with increasing liquid saturation. However, the *Pcg* value at *Sl*=1 or *Sg*=0 can be nonzero. If *Pcog* is not entered, the capillary pressure defaults to zero.

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 (*krog* at *Sl*=1) should equal the first entered value of *Krow* in the water-oil relative permeability table (*krow* at connate water saturation (*Swcon*)).

If the *NOSWC option is used, it is assumed that the connate water saturation is not part of the measured residual liquid saturation in the liquid-gas table. Thus *Slcon* and *Sorg* do not depend on the connate water saturation derived from the oil-water curve or entered using the *SWCON keyword

When *NOSWC is not used, reading in *SWCON arrays will effect both oil-water and liquid-gas tables.

When *NOSWC is used, reading in *SWCON will effect only the oil-water tables. The gas-liquid table is assumed to have no connate water in place, thus the connate liquid (*Slcon*) is equal to the irreducible oil (*Soirg*).

See the *SWCON card for further explanation.

Examples:

See examples in the *SWT keyword section. See Figure 8 in Appendix D for typical relative permeability curves.

THE *OILWET OPTION:

When the oil wet option is used with the BLOCK_CENTER initialization option, the gas capillary pressure entered in the P_{cog} column is the capillary pressure between the intermediate wetting phase (water) and the nonwetting phase (gas). Thus gas phase pressure is calculated as:

$$P_{gas} = P_{oil} + P_{cow} + P_{cog}$$

where:

- P_{gas}* = gas phase pressure
P_{oil} = oil phase pressure
P_{cow} = capillary pressure from the water oil rel-perm table.
P_{cog} = capillary pressure from the gas liquid rel-perm table.

Compare this with the water wet case or the oil wet case when the DEPTH_AVE initialization option is used where:

$$P_{gas} = P_{oil} + P_{cog}$$

Please refer to the tutorial section for a further discussion of the oil wet option used in IMEX:

THE *GASWATER OPTION:

When using the Gas Water option the SGT/SLT table is only altered slightly. Krog is not read in and the P_{cog} column is not read. Thus a typical Gas Water SGT/SLT table would be:

```
*SLT (*NOSWC)
Sl      krg
:
*Sgt (*NOSWC)
Sg      krg
:
```

When using the Gas Water option, it is important to ensure that the value of Soirg (irreducible oil in the liquid-gas table) calculated from this table is always 0.0 and that the value of Slcon is equal to Swcon (or is equal to 0.0, if the *NOSWC option is used).

In a *SGT table this is accomplished by ensuring that the final value of Sg in the table is 1.0-Swcon (or 1.0, if the *NOSWC option is used). In a *SLT table this is accomplished by ensuring that the first value of Sl is equal to Swcon (or 0.0, if the *NOSWC option is used).

THE *GASWATER_WITH_CONDENSATE OPTION:

When using the Condensate option the P_{cog} column is not read. P_{cgl} (capillary pressure between gas and liquid) is read in on the SWT table in place of P_{cow} (See *SWT).

The acceptable range of values for liquid or gas saturation is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

The acceptable range of values for gas relative permeability is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

The acceptable range of values for oil relative permeability is:

	SI	Field	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

The acceptable range of values for capillary pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	-2.5E+5	-36250.0	-2.5E+5	-2.5E+3
max	2.5E+5	36250.0	2.5E+5	2.5E+3

General Form of the *SLT(*SGT) Keyword:

TABLE:

```
*SLT (*NOSWC) (*SMOOTHEND (*QUAD | *CUBIC | *POWERQ |
    *POWERC | *OFF | *ON))
    (*SMOOTH (*ALL | *ALLPL | *ALLINT | *ALLLIN |
        *NONE | *OFF | property_list))
    (*PLAW pkrg pkrog ppcog ppcogi)
    (*DSLTI dslti_value) (*SLTKRTHR krthr_value)
    Sl           krg           krog           (Pcog   (Pcogi))
    :             :             :             :       :
*SGT (*NOSWC) (*SMOOTHEND (*QUAD | *CUBIC | *POWERQ |
    *POWERC | *OFF | *ON))
    (*SMOOTH (*ALL | *ALLPL | *ALLINT | *ALLLIN |
        *NONE | *OFF | property_list))
    (*PLAW pkrg pkrog ppcog ppcogi)
    (*DSLTI dslti_value) (*SLTKRTHR krthr_value)
    Sg           krg           krog           (Pcog   (Pcogi))
    :             :             :             :       :
```

Uses of the General Form:

Definitions of individual subkeywords will follow, but it is useful to first highlight a few of the many tasks the *SLT(*SGT) keyword can perform in situations where rock data is sparse and the user wants to generate meaningful curves.

1. Using Sparse data, generate curves which produce the best fit to an analytical power-law model over the entire range of data.

*SLT *SMOOTH *ALL or SLT *SMOOTH *ALLPL

2. Using Sparse data, generate curves which produce the best fit to an analytical power-law model over a smaller range of data which ensures matching of the original data at critical points and when $P_{cog} \rightarrow 0.0$ (or becomes constant).
`*SLT *SMOOTH *ALLINT`
3. Using no data, specify power law curves explicitly using input power law exponents and a two row relative perm. table (to define saturation end points and relative permeability end points)
`*SLT *SMOOTH *PLAW‘gas_exponent‘oil_exponent‘pcog_exponent’`
4. Re-interpolate an existing table using equal spacing of approx. 0.01 to take advantage of faster table look-up procedures.
`*SLT *SMOOTH *ALLLIN *DSLTI 0.01`
5. Alter the way the interpolation is performed in the table interval between where $kr=0$ and the first nonzero entry in the table. For Relative permeability, this is at Sgcrit (for Krg) and at Sorg (for Krog).
`*SLT *SMOOTHEND *CUBIC`

DEFINITIONS:

(General Form which includes all Regression Options)

The `*SLT (*SGT)` keyword can be subdivided into 5 subkeywords

- `*SMOOTHEND` - Smooth End-points of tables
- `*SLTKRTHR` - Gas Table Critical End-point Threshold Option
- `*SMOOTH` - Regression Options
- `*DSLTI` - Table Regeneration Options
- `*PLAW` - Direct Power Law Input (Regression not Used)
- `*SLT (*SGT)` (`*SMOOTHEND ‘subkeywords’`)
 - (`*SLTKRTHR krthr_value`)
 - (`*SMOOTH‘subkeywords’`)
 - (`*DSLTI dshti_value`)
 - (`*PLAW ‘subkeywords’`)

End-Point Smoothing Options:

***SMOOTHEND**

Optional keyword indicating what type of smoothing is to be used for the interval with $kr=0$ and $kr>0$ for krg and krog. These are the table intervals where gas or oil has just become mobile. The keyword may be followed by a subkeyword as indicated below.

Subkeywords of the *SMOOTHEND Keyword:

`*OFF | *ON | *QUAD | *CUBIC | *POWERQ | *POWERC`

If `*SMOOTHEND` is not specified or it has no subkeyword the default is `*SMOOTHEND *ON`. This is equivalent to `*SMOOTHEND *POWERQ`.

For the subkeyword *QUAD, a quadratic smoothing is used. For *CUBIC, a cubic smoothing is used. For *OFF, no smoothing (that is, linear interpolation) is used.

For *POWERQ and *POWERC end point smoothing is done using power law if the exponent is between 1.5 and 4.0. Otherwise, for *POWERQ, quadratic is used and for *POWERC, cubic is used for end point smoothing.

If the *SMOOTH regression option is not used, *POWERC And *POWERQ default to *CUBIC and *QUAD respectively.

Critical End-point Threshold Option:

*SLTKRTHR

Optional keyword to use *krthr_value* as a threshold value for *krg* and *krog* for end-point determination.

krthr_value

If a tabular value for *krg* or *krog* is less than or equal to *krthr_value*, then it is set to zero. If *SLTKRTHR is not specified, then a default value of 5.0E-07 is used. Entered values of *krthr_value* are reset to lie within the range 5.0E-16 to 5.0E-07.

Regression Options:

*SMOOTH

Perform regression on the relative permeability table entries and generate an equally spaced table which best fits the data over the specified range.

Valid Regression Option Subkeywords of the *SMOOTH Keyword:

*ALL | *ALLPL | *ALLINT | *ALLLIN | *NONE | *OFF | *property_list*

If the *ALL subkeyword is used, or no subkeywords are found, then all properties in the table are smoothed according to default actions. This tries power law first. For kr's, if the resultant power is greater than 4.0 then linear is used for that property. The smoothed ranges for *ALL are as follows:

Property	Range
Krg	Sllow to 1.0-Sgcon
Krog	Slcrit to 1.0-Sgcon
Pcog	Sllow to Sl(Pcog=0.0)
Pcogi	Sllow to Sl(Pcogi=0.0)

where

Slcrit	=	Critical Liquid Saturation
Sllow	=	Lowest Sl in Table
Sl(Pcog=0.0)	=	Lowest Sl in table with Pcog constant
Sgcon	=	Connate Gas Saturation which is assumed equal to Critical Gas Saturation

***ALLPL**

Specifies that Power law is used for all properties in the table. If a power > 4.0 is calculated, the power is used.

***ALLINT**

Specifies that Power law is used between a reduced range of end points the smoothed ranges for *ALLINT are as follows

Property	Range
Krg	Slcrit to 1.0-Sgcon
Krog	Slcrit to Slmax
Pcog	Sllow to Sl(Pcog=0.0)
Pcogi	Sllow to Sl(Pcogi=0.0)

where

Slcrit	=	Critical Liquid Saturation
Sllow	=	Lowest Sl in Table
Sl(Pcog=0.0)	=	Lowest Sl in table with Pcog constant
Sgcon	=	Connate Gas Saturation which is assumed equal to Critical Gas Saturation

***ALLLIN**

Generates an evenly spaced table using linear interpolation for all properties.

If the *NONE or *OFF keyword is used then none of the table properties are smoothed. This is the default action if the *SMOOTH keyword is not specified.

property_list allows the user to customize how regression is applied to a list of properties.

For a *SLT(*SGT) table, valid properties to be smoothed include:

*KRG	*KRGPL	*KRGINT	*KRGLIN
*KROG	*KROGPL	*KROGIN	*KROGLIN
*PCOG	*PCOGPL	*PCOGINT	*PCOGLIN
*PCOGI	*PCOGIPL	*PCOGIINT	*PCOGILIN

If property_list is used and a particular, existing property is not specified, then linear interpolation is used for that property.

The conventions for the above properties are the same as for *ALL, *ALLPL and *ALLINT and *ALLLIN.

Table Regeneration Option:***DSLTI**

When the Regression options (*SMOOTH) are not used IMEX does not regenerate equally spaced rel. perm. tables, i.e.:

Equally spaced tables not generated when,

1. the *SMOOTH keyword is not found,

2. the *SMOOTH *NONE keyword is found,
3. the *SMOOTH *OFF keyword is found.

When regression is employed IMEX automatically creates equally spaced tables for the newly created curves.

The *DSTLI keyword can be used to define the number of points in the table.

NOTE:

SMOOTH *ALLLIN and *DSLTI *dslti_value

Can be used together to generate new equally Spaced tables without using regression on table values. This can speed up relative permeability calculation times as equally spaced tables are interpolated more efficiently.

dslti_value

Generate an evenly space table with deltaSl approximately equal to *dslti_value*. Calculate deltaS as follows:

$$\text{deltaSl} = (1 - S_{lc} - S_{gc}) / \text{int}((1 - S_{lc} - S_{gc}) / dslti + \text{eps})$$

Using the input *dslti_value* and where int(x) is the truncated integer value of x.

If the *DSLTI keyword is not specified, the default for *dslti_value* is calculated internally. The minimum of half of the minimum table spacing and 0.016 is calculated. If this value is greater than 0.001 then *dslti_value* is set to it. Otherwise, if *DSLTI is not specified, *dslti_value* is set to 0.001.

Direct Input of Power Law Coefficients:

***PLAW**

Input power law exponents for table properties for cases with only end points input.

pkrg

Power law exponent for krg for cases with only end points input. A value of 0.0 results in a default value of 1.0.

pkrog

Power law exponent for krog for cases with only end points input. A value of 0.0 results in a default value of 1.0.

ppcog

Power law exponent for P cog for cases with only end points input. A value of 0.0 results in a default value of 1.0. If it is not specified, then it is assumed that P cog is zero.

ppcogi

Power law exponent for P cog i for cases with only end points input. A value of 0.0 results in a default value of 1.0. If it is not specified, then it is assumed that P cog i is zero.

EXPLANATION:

For optional smoothing, power law or linear interpolation may be used for *krg* and *krog* and, if it exists, for *Pcog*. If *SMOOTH keyword is omitted, or *SMOOTH *NONE or *OFF is specified, no smoothing is done. Otherwise, the keywords

*SMOOTH (*ALL | *ALLPL | *ALLINT | *ALLLIN |
*NONE | *OFF | *property_list*)

determine for which properties power law and for which properties linear interpolation are to be used.

If *PLAW is specified, then all properties are assumed to be power laws using the specified exponents. End points and maximum values are determined from the following table. All other values, if they exist are ignored. If *PLAW is not specified and power law correlations are specified, then there must be at least two SI's or Sg's given such that the particular property is nonzero. The power for that property is then calculated using nonlinear least squares.

If the *SMOOTH keyword is specified, an equally spaced table is generated. Using a power law or linear interpolation as specified by its subkeywords. Within the simulator, this equally spaced table is used. Relative permeabilities are calculated using linear interpolation from this internal table except over the interval between kr=0 and kr>0. On this interval, the type of smoothing, if any, is controlled by the *SMOOTHEND keyword and optionally one of its subkeywords.

Examples:

A. Generate power law correlations for Krg, Krog and Pcog (drainage) matching endpoints and maximum values for a gas liquid (*SLT) table where Slcrit = Slcon and Sgcrit = Sgcon:

1. Input the power law exponents for Krg, Krog and Pco on the *SLT keyword.

*SLT *SMOOTH *PLAW pkrg pkrog ppcog

2. Input the endpoints and values of Krgcl and Krogcg and minimum and maximum values of drainage oil gas capillary pressure in the two row table.

Slcrit = Slcon	Krgcl	0.0	Pcogmax
1 - Sgcrit	0.0	Krogcg	Pcogmin

B. Generate power law correlations for Krg, Krog and Pcog (drainage) matching endpoints and maximum values for a gas liquid (*SLT) table where Slcrit > Slcon and Sgcrit = Sgcon:

1. Input the power law exponents for Krg, Krog and Pco as in A above.
2. Input the endpoints and values of Krgcl and Krogcg and minimum and maximum values of drainage oil gas capillary pressure in the three row table.

Slcon	Krgcl	0.0	Pcogmax
Slcrit	*INT	0.0	*INT
1 - Sgcrit	0.0	Krogcg	Pcogmin

Note: Any value may be used in place of *INT, as Krg and Pcog at Sl = Slcrit are ignored when using *PLAW.

For the above (A and B), assuming *SMOOTH *ALL (the default) or *SMOOTH *ALLPL, we obtain

$$Krog = Krogcl \left(\frac{Sl - Slcrit}{1 - Slcrit - Sgcrit} \right)^{Pkrog}$$

$$Krg = Krgcl \left(\frac{1 - Sgcrit - Sl}{1 - Slcon - Sgcrit} \right)^{Pkr}$$

$$\begin{aligned} P_{cog_D} &= (P_{cog}(Sl = Slcon) - P_{cog}(Sw = 1 - Sgcrit)) \left(\frac{1 - Sgcrit - Sl}{1 - Slcon - Sgcrit} \right)^{Ppcog} \\ &\quad + P_{cog}(Sw = 1 - Sgcrit) \end{aligned}$$

Capillary Pressure Hysteresis Parameters (Optional)

***EPSPC,**
***EPSPCG**

PURPOSE:

These keywords signal entry of the hysteresis parameters.

***EPSPC**

Denotes the entry of *epspc*. This determines the transition between the imbibition and drainage curves for oil-water capillary pressure.

***EPSPCG**

Denotes the entry of *epspcg*. This determines the transition between the imbibition and drainage curves for oil-gas capillary pressure.

FORMAT:

*EPSPC	<i>epspc</i>
*EPSPCG	<i>epspcg</i>

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. *epspc* cannot be less than 1.0d-05.

epspcg

Dimensionless real number which determines the transition between the imbibition and drainage curves for oil-gas capillary pressure. Typical values of *epspcg* lie between 0.05 and 0.1. *epspcg* cannot be less than 1.0d-05.

DEFAULTS:

The water-oil capillary pressure hysteresis effect is included if the fifth column (Pcowi) 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 (Pcogi) 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.

EXPLANATION:

The *epspc* and *epspcg* refers to the rock type set_number specified by *RPT (which must be defined before these keywords are used). If more than one rock type is present then *EPSPC and *EPSPCG must be specified before the next *RPT keyword. After the next *RPT

keyword, a new set of *EPSPC and *EPSPCG can be specified for the new rock type *set_number*.

When Pcow 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

$$pcow(Sw) = (f) \times pcow(Im,Sw) + (1-f) \times pcow(Dr,Sw);$$

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-Soirw. pcow remains on this scanning curve until Sw falls below Sw(hyst), where 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:

$$pcow(Sw) = (f) \times pcow(Dr,Sw) + (1-f) \times pcow(Im,Sw),$$

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), where pcow reverts to the imbibition curve. Let dSmax 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 values of 0.05 to 0.1 for epspc generally leads to a moderate transition. epspc cannot be less than 1.0d-05.

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

$$pcog(Sl) = (f) \times pcog(Im,Sl) + (1-f) \times pcog(Dr,Sl);$$

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:

$$pcog(Sl) = (f) \times pcog(Dr,Sl) + (1-f) \times pcog(Im,Sl),$$

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 leads to a moderate transition. $epspcg$ cannot be less than 1.0d-05.

Example:

```
*EPSPC 0.05 *EPSPCG 0.05
```

Oil-Water Relative Permeability Hysteresis Parameters (Optional)

***HYSKRO, *HYSKRW, *KILLOUGH_MOD_KRWCO**

PURPOSE:

These keywords assign the hysteresis method of oil-water relative permeabilities and signal the entry of associated parameters.

FORMAT:

```
*HYSKRO      Sormax
              (*CARLSON (Sormax) | *KILLOUGH (*HYEXO (hyexo)))
*HYSKRW      (*KILLOUGH | *KILLOUGH_MOD (*HYEXW hyexw))
*KILLOUGH_MOD_KRWCO   krwco
```

DEFINITIONS:

sormax

Sormax is the maximum possible imbibition residual oil saturation (Oil-water table). It is the end-point of the boundary imbibition branch of oil relative permeability Krow. This value is used to evaluate the shape and path of all imbibition scanning curves which leave the drainage curve at any saturation. Sormax denotes the oil trapping of the rock type. The difference Sormax–Sorw gives the maximum possible trapped oil saturation.

CARLSON

Denotes the use of Carlson's Krow hysteresis method. This subkeyword can be followed by a Sormax value if Sormax is not defined in a *SWT or *SWTI table.

KILLOUGH

Denotes the use of Killough's hysteresis method for relative permeability.

KILLOUGH_MOD

Denotes the use of the modified Killough hysteresis method for water relative permeability Krw.

HYEXO

Denotes the use of Killough's saturation interpolation method in Krow hysteresis.

HYEXW

Denotes the input of exponent *hyexw* in Krw hysteresis.

hyexo

Exponent of Killough's saturation interpolation equation .

hyexw

Exponent of Killough's Krw hysteresis method equation

krwco

Coefficient of the equation of Killough's Krw hysteresis method. Only used when Sormx equals Sorw.

DEFAULTS:

If the rock type Sormax value is greater than Sorw, and the same rock type does not have imbibition oil relative permeability in the *SWTI table, *HYSKRO *CARLSON is assumed for Krow hysteresis.

If the imbibition oil relative permeability Krowi is entered in the *SWTI table, but *HYSKRO is not provided, *HYSKRO *KILLOUGH is assumed for Krow hysteresis.

If the imbibition water relative permeability is entered in the *SWTI table but *HYSKRW is not provided, HYSKRW *KILLOUGH_MOD is assumed for Krw hysteresis.

If *HYSKRO *KILLOUGH *HYEXO is entered without exponent hyexo, the default *hyexo* is assumed (1.5).

For Krw hysteresis, if not specified, the default value of hyexw is 1.5 and the default value of krwco is 0.5.

CONDITIONS:

These keywords must be in the Rock-Fluid Data keyword group, after the *RPT keyword. Exponents *hyexo* and *hyexw* must be greater than one. The range of krwco is from zero to one.

EXPLANATION:

Entering *HYSKRO *Sormax* invokes the original Kro hysteresis which is not necessarily coupled with the hysteresis of capillary pressure Pcow.

To model both relative permeability and capillary pressure hysteresis in oil-water system the rigorously coupled hysteresis is recommended. To activate the coupled hysteresis, the maximum residual oil saturation, Sormax, must be given via imbibition capillary pressure Pcowi in the *SWT table or defined in the *SWTI table.

For hysteresis of water relative permeability Krw, *KILLOUGH_MOD uses an equation based on a modified Killough method. The modified equation addresses the problem where the scanning Krw curve crosses the boundary imbibition curve.

Choosing the proper values for exponents *hyexo* and *hyexw* is also important. Setting the exponents to 1 will produce straight lines for the scanning curves. The default value of 1.5 produces slightly concave scanning curves. Analyzing the input of the drainage and imbibition relative permeability curves is required to obtain the proper number. See Eq. 4 and Eq.8 of tutorial 'Relative Permeability and Capillary Pressure Hysteresis in Oil-Water System'.

Coefficient krwco is only used for modified Killough Krw when Sormax equals Sorw (Eq.9a of the tutorial). The default value of krwco is set to 0.5. This value can be adjusted by keyword *KILLOUGH_MOD_KRWCO. The small krwco makes the scanning curve close to drainage Krw and large krwco makes the scanning curve close to boundary imbibition Krw. The reasonable range of the coefficient is 0.2 to 0.8.

Refer to CMG tutorial 'Relative Permeability and Capillary Pressure Hysteresis in Oil-Water System' for further technical information.

Example:

If the *SWTI table defines both Krowi and Krwi:

```
HYKRO *KILLOUGH  
*HYKRW *KILLOUGH_MOD *HYEXW 2.0
```

Gas Relative Permeability Hysteresis Parameter (Optional)

*HYSKRG

PURPOSE:

This keyword denotes the hysteresis effect on the gas relative permeability is modelled and signals the entry of the hysteresis parameter.

FORMAT:

*HYSKRG *sgrmax*

DEFINITIONS:

sgrmax

The maximum possible imbibition residual gas saturation. It is the endpoint of the boundary imbibition branch which breaks off from the gas relative permeability drainage curve at the maximum possible krg (at $S_g = 1.0 - S_{lcon}$). This value is used to evaluate the shape and path of all imbibition curves which leave the drainage curve at any saturation. *sgrmax* must be greater than S_{gcrit} and less than $1.0 - S_{lcon}$. Even though S_{gcrit} is normally close to zero, *sgrmax* can be quite large, values of 0.3 to 0.4 are typical.

DEFAULTS:

If *HYSKRG is not entered the hysteresis effect on gas relative permeability is not included in the simulation.

CONDITIONS:

This keywords must be in the Rock-Fluid Data keyword group, after the *RPT keyword.

*HYSKRG must be followed by the maximum residual gas saturation *sgrmax*.

EXPLANATION:

The *sgrmax* refers to the rock type *set_number* specified by *RPT (which must be defined before these keywords are used). *HYSKRG can be entered per each rock type.

The hysteresis effect on krg is calculated as follows:

$$krg(S_g) = krg(Dr; S_g), \text{during drainage};$$

$$krg(S_g) = krg(Dr; S_g(\text{shifted})), \text{during imbibition};$$

where

$$S_g(\text{shifted}) = S_{gcrit} + (S_g - S_{grh})(S_{gh} - S_{gcrit}) / (S_{gh} - S_{grh}).$$

krg is assumed to lie on the drainage curve initially; S_{gh} is the value of S_g when the shift to imbibition occurs and S_{grh} is the value of S_{gcrit} corresponding to S_{gh} via Land's equation,

$$\frac{1}{S_{grmax} - S_{gcrit}} - \frac{1}{S_{gmax} - S_{gcrit}} = \frac{1}{S_{grh} - S_{gcrit}} - \frac{1}{S_{gh} - S_{gcrit}}$$

where $S_{gmax} = 1 - S_{lcon}$. S_{grmax} has the value of the user-entered parameter *sgrmax*.

Example:

*HYSKRG 0.4

Reduced Vertical Extent of P_{cow} Cycle for Trapped Oil Hysteresis (Conditional)

*DAMP-PCOW-TROIL

PURPOSE:

Allows linear reduction of P_{cow} values when Soh towards Sorw so as to make a “damped” P_{cow} hysteresis cycle.

FORMAT:

*DAMP-PCOW-TROIL

DEFAULTS:

Conditional keyword. By default the option is off.

CONDITIONS:

Keyword only effective when the rock type traps oil ($S_{omax} > S_{orw}$).

DEFINITION:

In the trapped oil hysteresis option for P_{cow} , all P_{cow} scanning cycles start from the historically attained maximum oil saturation, S_{oh} , and end at the residual oil saturation of the imbibition process, S_{orh} .

If Soh is very close to Sorw (And therefore Sorh), and the difference between the imbibition capillary pressures at Sorh and the drainage capillary pressure at Soh is large, a small change in water saturation may result in very large change in capillary pressure.

This keyword provides an option to reduce the capillary pressure difference which might result in a pressure oscillation. The reduction factor linearly varies from 1 to 0 as the historical attained oil, S_{oh} , changes from $1-S_{wcon}$ to S_{orw} . Please see the tutorial “The Trapped Oil Hysteresis Option for P_{cow} and K_{row} (Oil-Water System)” for discussions and graphs.

Oil-Trapping Relationship (Optional)

***OILTRAP**

PURPOSE:

*OILTRAP assigns the oil-trapping relationship that is used to determine the residual oil saturation of imbibition from the historical-maximum-attained oil saturation.

FORMAT:

```
*OILTRAP ( *LAND |
    *LINEAR |
    *TABLE
    Sohy   Sorh
    :       :     )
```

DEFINITIONS:

***LAND**

Denotes the use of Land's correlation.

***LINEAR**

Denotes the use of a linear relationship.

***TABLE**

Denotes the use of a user-provided relationship, defined by the following table.

Sohy

Historical-maximum-attained oil saturation.

Sorh

Residual oil saturation of imbibition process.

DEFAULTS:

Optional keyword. If the keyword is not defined and the oil-trapping relationship is required, the default Land's correlation will be used.

CONDITIONS:

Keyword is only effective when the rock type traps oil (i.e. Sormax > Sorw).

If a table is used to define the relationship, the first column, Sohy, must start from the residual oil saturation Sorw, and increase to 1-Swcon, the maximum attainable oil saturation. The second column, Sorh, must also start from Sorw, and increase to the maximum residual oil saturation, Sormax.

EXPLANATION:

The oil-trapping relationship determines the amount of trapped oil when a block starts imbibition from its initial equilibrium. Using a different relationship will change the amount of oil recovery from the reservoir transition zone. Land's correlation is based on gas-water data, and is therefore considered a strong water wetting relationship. Land's correlation is expressed as:

$$S_{orh} = S_{orw} + \frac{1}{1/(S_{ormax} - S_{orw}) - 1/(1 - S_{wcon} - S_{orw}) + 1/(S_{ohy} - S_{orw})}$$

The linear relationship is:

$$S_{orh} = S_{orw} + (S_{ohy} - S_{orw}) \frac{(S_{ormax} - S_{orw})}{(1 - S_{wcon} - S_{orw})}$$

Compared with Land's correlation, the linear relationship results in lower Sorh (less trapped oil) and is thus considered a less-water-wetting relationship.

Method for Evaluating 3-Phase Kro (Optional)

***KROIL**

PURPOSE:

*KROIL introduces subkeywords which indicate how Kro is evaluated. The options are: Stone's first model; Stone's second model as modified by Aziz and Settari; 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-Swcon-So.

FORMAT:

```
*KROIL  ( *STONE2 (*SWSG | *SO) |
           *STONE1 (*SWSG | *SO) |
           *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 a 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 for Krow and Krog is *SWSG. *STONE2 *SWSG is the default when *KROIL appears with no following subkeyword.

***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 - Swcon - So

for use in the computation of kro with Stone's second model. This can be used to model Krow and Krog entered as functions of oil saturation only.

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, but should only appear once. *KROIL is not rock type dependent and applies to all tables. If *KROIL appears more than once, the last occurrence overwrites previous occurrences.

EXPLANATION:

In STONE'S SECOND MODEL for kro as modified by Aziz and Settari, the oil relative permeability is computed as

$$kro = krocw * \left\{ \left(\frac{krow}{krocw} + krw \right) \left(\frac{krog}{krocw} + krg \right) - krw - krg \right\}$$

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 IMEX, 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 - Swcon - 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. IMEX 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(1 - So) and krog(Sg = 1 - Swcon - So) as krog(1 - So) and specify *KROIL *STONE2 *SO. 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

$$kro = \frac{Sostar * krow * krog}{krocw * (1 - Swstar) * (1 - Sgstar)}$$

where

$$Sostar = \frac{(So - Som)}{(1 - Swcon - Som)}$$

$$Swstar = \frac{(Sw - Swcon)}{(1 - Swcon - Som)}$$

$$Sg_{star} = \frac{Sg}{(1 - Sw_{con} - Som)}$$

for the "minimal" value Som of the oil saturation, IMEX uses the linear function of Sg proposed by Fayers and Matthews (SPEJ April 1984, pp. 224-232):

$$Som(Sg) = (1 - a(Sg)) * Sorw + a(Sg) * Sorg,$$

where

$$a(Sg) = \frac{Sg}{(1 - Sw_{con} - Sorg)}$$

As in Stone's second model, when the subkeyword *SO is specified, krow(1-So) and krog(1-Swcon-So) are used in the formula; otherwise, krow(Sw) and krog(Sg) are used.

The LINEAR ISOPERM model was proposed by L.E. Baker (paper SPE 17369, 1988). In this method, kro(Sg,Sw) is defined in the region $Sw_{con} < Sw < 1 - Sorw$, $0 < Sg < (1 - Sorw - Sw) * (1 - Sorg - Sw_{con}) / (1 - Sorw - Sw_{con})$, 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

$$kro(Sg=0, Sw) = krow(Sw) = 0.2 \quad \text{for } Sw = Sw_2$$

and

$$kro(Sg, Sw_{con}) = krog(Sg) = 0.2 \quad \text{for } Sg = Sg_2,$$

then all points (Sg,Sw) on the line segment joining (0,Sw₂) to (Sg₂,Sw_c),

$$(Sg, Sw) = (0, Sw_2) + b * (Sg_2, Sw_{con} - Sw_2), \quad 0 < b < 1,$$

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

$$kro = \frac{(Sg * krog + (Sw - Sw_{con}) * krow)}{(Sg + Sw - Sw_{con})}$$

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 *SWSG but looked up as in *SO). This model gives similar results as STONE2 on the IMEX templates. However, on a large field problem, significant differences were noted both in results and in CPU times.

This model was initially proposed by L.E Baker in SPE/DOE Paper (17369) presented at the 6-th Symposium on Enhanced Oil Recovery, Tulsa, Oklahoma.

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

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 have been entered.

This keyword can also be in the Well and Recurrent Data section as *RTYPE can be altered during the run.

If saturation or relative permeability value end point array entry by rock type input is used, then the *RTYPE array definition must be placed before any arrays (*SWCON etc) are defined.

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.

When *RTYPE is redefined repeatedly in recurrent data, the effected block saturation end points are individually reset to the new curve values. However, after an end point is read in explicitly (say using the *SWCON keyword) this behavior changes.

After an end point (Swcon, Swcrit, Soirw, Slcon, Sorg, Sorw, Sgcon, Krwi, Krocw, Krgcl, Krogcg, Pcwmax, Pcgmax, JFwmax, or JFgmax, Krwro, Krocw, Krgrl, Krogcr) is read in explicitly the individual end point is always honored, even when the rock types change. End points not explicitly defined are reset, but the user must explicitly alter an end point that has been entered using one of the end point array keywords (*SWCON, *SWCRIT, *SOIRW, *SLCON, *SORW, *SORG, *SGCRIT, *SGCON, *KRWI, *KROCW, *KRGCL, *KROGCG, *PCWMAX, *PCWMIN, *PCWIMIN, *PCGMAX, *PCGMIN, *PCGIMIN, *JFWMAX, *JFWMIN, *JFGMAX, *JFGMIN, *KRWRO, *KROCRW, *KRGRL, *KROGCRG).

This procedure is different from the one used on the *PERF family of cards when rock type and end points are read in. On the *PERF family of cards all values are reset to the values defined. End points are always reset.

Saturation Endpoints for each Grid Block (Optional)

*SWCON, *SWCRIT, *SOIRW, *SORW, *SORMAX, *SGCON, *SGCRIT, *SLCON, *SORG, *SOIRG

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.

ARRAY:

*SWCON
*SWCRIT
*SOIRW
*SORW
*SORMAX
*SGCON
*SGCRIT
*SLCON
*SORG
*SOIRG (SCALING-STARS only)

DEFINITIONS:

***SWCON**

Connate water saturation of a grid block. *SWCON is the first entry in the *SWT table. 1.0-*SWCON is the maximum oil saturation possible in the *SWT table.

***SWCRIT**

Critical water saturation of a grid block. *SWCRIT is the saturation at which water first becomes mobile in the two phase oil-water system. It is always equal to or greater than *SWCON.

***SOIRW**

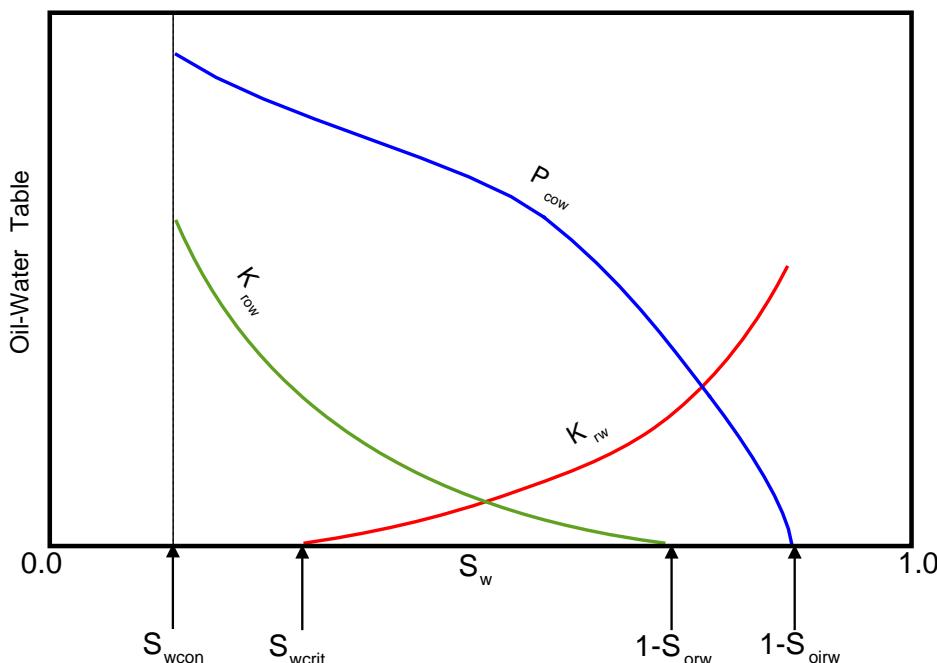
Irreducible oil saturation of a grid block. 1-*SOIRW is the last water saturation entry in the *SWT table. *SOIRW is the minimum oil saturation possible in the *SWT table. 1-*SOIRW is the maximum water saturation in the *SWT table.

***SORW**

Residual oil saturation of a grid block. *SORW 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 *SOIRW.

*SORMAX

Maximum residual oil saturation of a grid block. Imbibition end point of the water-oil hysteresis model. *SORMAX is the oil saturation at which oil first becomes immobile for a boundary/primary imbibition process (imbibition starting at oil saturation equals to $1-S_{wcon}$). Its value is always between $1-S_{wcon}$ and *SORW. *SORMAX is only effective for blocks which have Krow hysteresis or trapped oil hysteresis enabled. Please see the tutorial “Trapped Oil Hysteresis for Pcow and Krow” for discussions and graphs.



*SGCON

Connate gas saturation of a grid block. *SGCON is the minimum gas saturation in the two phase gas-liquid system. It is the first entry in a *SGT table and $1-*SGCON$ is the last entry in a *SLT table.

*SGCRIT

Critical gas saturation of a grid block. *SGCRIT is the gas saturation at which gas first becomes mobile. It is always equal to or greater than *SGCON.

*SLCON

Connate liquid saturation of a grid block in the gas-liquid table. *SLCON includes connate water (*SWCON) unless the *NOSWC option is employed. It is the minimum liquid saturation in the two phase gas-liquid system and the first entry in the *SLT table and one minus the last entry in the *SGT table.

*SORG

Residual oil saturation of a grid block in the two phase gas liquid system. It does not include connate water, thus *SORG + *SWCON = residual liquid saturation. If the *NOSWC option is active *SORG = residual liquid saturation.

*SOIRG

(*SCALING-STARS only)

Irreducible oil saturation of a grid block in the two phase gas liquid system. It does not include connate water, thus *SOIRG + *SWCON = connate liquid. This array is only input when using the *SCALING-STARS option. It cannot be used with *SLCON (will trigger an error), and if found in a data set, it will automatically enable the *SCALING-STARS option (if not already enabled).

In the IMEX endpoint scaling option, the input of an *SWCON array would not change the value of *SLCON instead *SOIRG would be altered to maintain the original value of *SLCON. When *SCALING-STARS is enabled *SOIRG is kept constant and *SLCON is allowed to change. This duplicates the behavior in the STARS simulator.

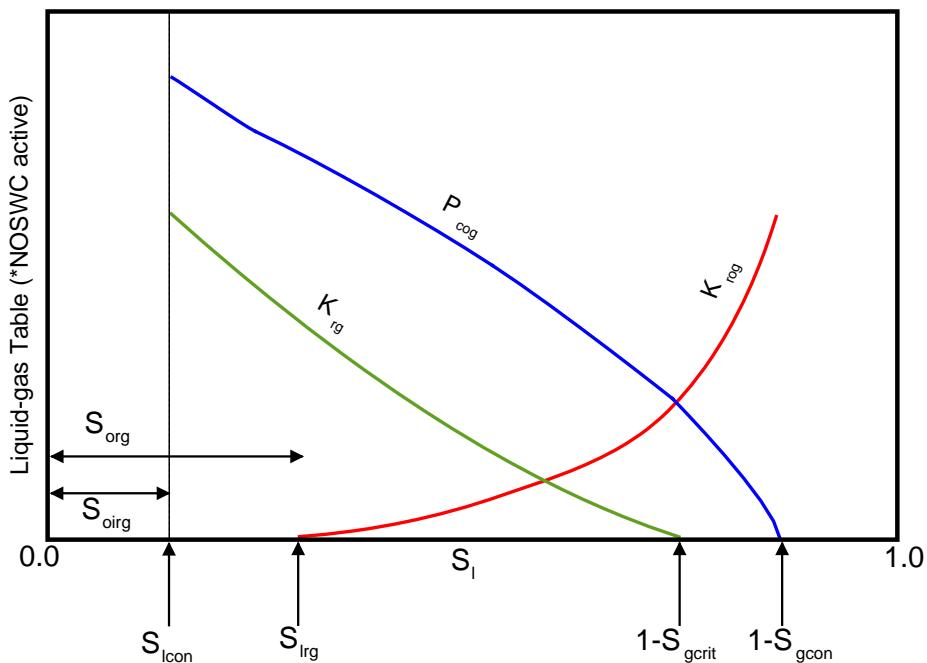
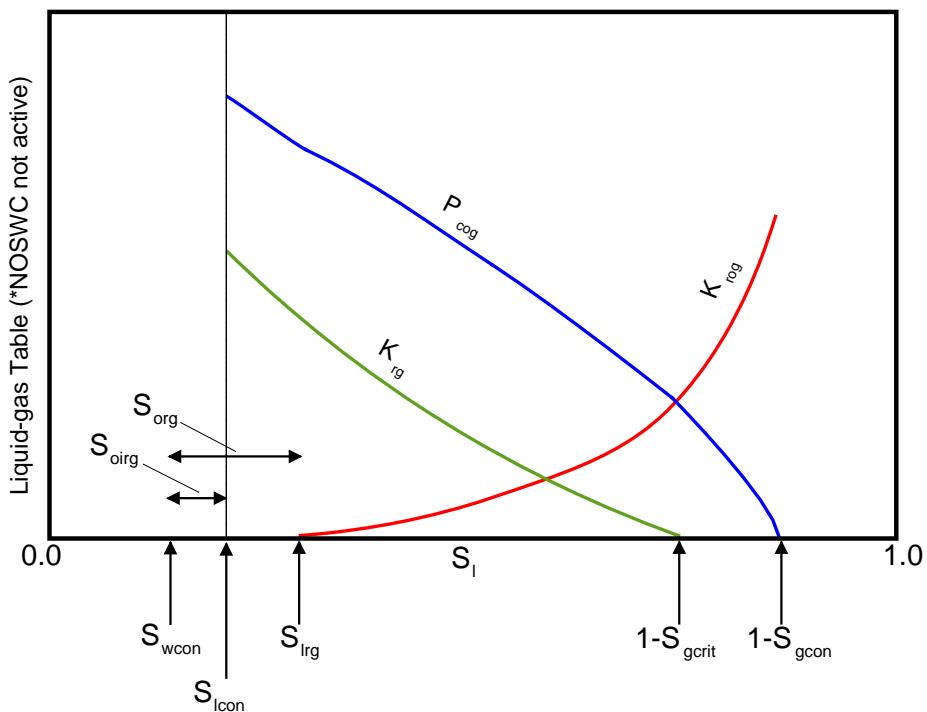
In the IMEX endpoint scaling option, when an endpoint is not entered as an array, it is read for all blocks from the appropriate *RPT table, regardless of its value.

In the STARS scaling option, this is not the case for connate/irreducible and critical saturations when these values are identical in the relative permeability tables.

For example, if *SWCON is input and *SWCRIT is not input for a gridblock, and in the table active for that gridblock, the table value of Swc and Swcon are identical, *SWCRIT is set equal to *SWCON. If *SWCRIT is input and *SWCON is not input, *SWCON is set equal to *SWCRIT. In other words we assume that the user only really ever wanted one endpoint instead of two.

Of course if two endpoint arrays are read in (for example the user input both *SWCON and *SWCRIT) we override this behavior.

Similarly the same behavior occurs for (*SGCRIT - *SGCON), (*SOIRW - *SORW), and (*SORG - *SOIRG) pairs.



where S_{lrg} is the residual liquid in the gas-liquid table.

$S_{oирg}$ is the irreducible oil in the gas liquid table.

If *NOSWC is not active:

$$S_{lrg} = S_{org} + S_{wcon}$$

$$S_{oирg} = S_{lcon} - S_{wcon}$$

If *NOSWC is active:

$$S_{lrg} = S_{org}$$

$$S_{oирg} = S_{lcon}$$

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:

This keyword can be in the Rock-Fluid Data keyword group after all the two phase relative permeability tables have been entered and/or in the Well and Recurrent Data section as all of the end points can be altered during the run.

The use of these end point keywords changes how the *RTYPE keyword in the Well and Recurrent Data section chooses end point values for tables being reassigned to new grid blocks. Please see *RTYPE for an explanation of how the end point keywords in the Well and Recurrent Data section interact.

If saturation end point array entry by rock type input is used, then the *RTYPE array definition must be placed before any end point arrays (*SWCON etc) are defined.

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.

These values override the values of the saturation end points. 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 block's saturation end point if it has been overridden by these keywords (see *RTYPE Keyword). Without the use of these keywords, if a blocks *RTYPE is changed, its saturation endpoints will be changed as well.

In addition to the standard array reading options, the eight saturation end point arrays can be entered according to rock type. The format used when using this entry technique is:

"Saturation End Point Array" *RTYPE "Rock Type Table number" "End Point Value"

For example, if the value of connate water saturation for table number 3 is to be changed to 0.12, the format would be:

*SWCON *RTYPE 3 0.12

This end point array reading format searches all blocks whose rock type is 3 and assigns the connate water saturation of those blocks to the value of 0.12. The result is identical to the result one would obtain if an *IJK array reading option was used to explicitly change each block whose *RTYPE was 3 to 0.12.

Multiple occurrences of this format are valid, for example:

```
*SWCON *RTYPE 1 0.13
*SWCON *RTYPE 2 0.10
*SWCON *RTYPE 3 0.20
```

In addition it is possible to define a saturation end point array using a standard array reading option and override the input values for specific rock types using this format, for example:

```
*SWCON *IJK 1:5 1:10 1 0.10
*SWCON *RTYPE 3 0.15
```

Initially all block's connate water saturation are set to the saturation end points defined by each rock type's relative permeability table (the default), the keywords above override blocks in the IJK range 1:5 1:10 1 with the connate water saturation of 0.10 and further set the connate water saturation of all blocks in rock type 3 to 0.15.

SCALING

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:

$$K_{rw}, S'_w = S'_{wcrit} + (S_w - S_{wcrit}) * \left(\frac{1 - S'_{oirw} - S'_{wcrit}}{1 - S_{oirw} - S_{wcrit}} \right)$$

$$K_{row}, S''_w = S'_{wcon} + (S_w - S_{wcon}) * \left(\frac{1 - S'_{orw} - S'_{wcon}}{1 - S_{orw} - S_{wcon}} \right)$$

$$P_{cow}, S'''_w = S'_{wcon} + (S_w - S_{wcon}) * \left(\frac{1 - S'_{oirw} - S'_{wcon}}{1 - S_{oirw} - S_{wcon}} \right)$$

S_w 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:

$$K_{rog}, S'_1 = S'_{lrg} + (S_1 - S_{lrg}) * \left(\frac{1 - S'_{lrg} - S'_{gcon}}{1 - S_{lrg} - S_{gcon}} \right)$$

$$K_{rg}, S''_1 = S''_{lcon} + (S_1 - S_{lcon}) * \left(\frac{1 - S''_{lcon} - S''_{gcrit}}{1 - S_{lcon} - S_{gcrit}} \right)$$

$$P_{cog}, S'''_1 = S'''_{lcon} + (S_1 - S_{lcon}) * \left(\frac{1 - S'''_{lcon} - S'''_{gcon}}{1 - S_{lcon} - S_{gcon}} \right)$$

$$S_l = S_o + S_w$$

$S_{lrg} = S_{wcon} + S_{org}$, if the *NOSWC option is not used.

$S_{lrg} = S_{org}$, if the *NOSWC option is used.

In the latter case S_{org} includes all residual liquid in the liquid-gas system.

S'_1, S''_1, S'''_1 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_{lcon} , 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.

K_{rw}	$S_{wcrit} \leq S_w \leq 1 - S_{oirw}$,	thus K_{rw} is scaled by S_{wcrit} and S_{oirw}
K_{row}	$S_{wcon} \leq S_w \leq 1 - S_{orw}$,	thus K_{row} is scaled by S_{wcon} and S_{orw}
P_{cow}	$S_{wcon} \leq S_w \leq 1 - S_{oirw}$,	thus P_{cow} is scaled by S_{wcon} and S_{oirw}
K_{rog}	$S_{lrg} \leq S_l \leq 1 - S_{gcon}$,	thus if *NOSWC is not active, K_{rog} is scaled by S_{wcon} , S_{org} and S_{gcon} . If *NOSWC is active K_{rog} is scaled by S_{org} and S_{gcon}
K_{rg}	$S_{lcon} \leq S_l \leq 1 - S_{gcrit}$,	thus K_{rg} is scaled by S_{lcon} and S_{gcrit}
P_{cog}	$S_{lcon} \leq S_l \leq 1 - S_{gcon}$,	thus P_{cog} is scaled by S_{lcon} and S_{gcon}

THE *OILWET OPTION:

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 nonwetting liquid phase.

When the oil wet option is active the column which normally contains water saturation should now contain the saturation of the wetting phase (oil). The K_{rw} column should contain the wetting phase relative permeability (oil relative permeability). The K_{row} column contains the nonwetting phase relative permeability (water relative permeability).

The meanings of the columns of the gas-liquid tables are not altered.

The use of the oil-wet option also changes the meaning of the user specified grid block specific end point arrays

S_{wcon}	= Connate wetting phase (oil) saturation
S_{wcrit}	= Critical wetting phase (oil) saturation
S_{oirw}	= Irreducible nonwetting phase (water) saturation
S_{orw}	= Residual nonwetting phase (water) saturation
S_{ormax}	= Maximum residual nonwetting phase (water) saturation
S_{org}	= Residual nonwetting phase (water) saturation
S_{lcon}	= Connate liquid saturation which is equal to connate oil (S_{wcon}) plus irreducible water saturation (S_{oorg}) if the *NOSWC option is not used, and equal to irreducible water saturation (S_{oorg}) alone, if the *NOSWC option is used

Please refer to the Tutorial section for a discussion of the IMEX oil wet option.

Endpoint Relative Permeability Value for Each Grid Block (Optional)

*KRWIRO, *KROCW, *KRGCL, *KROGCG, *KRWRO, *KROCRW,
*KRGRL, *KROGCRG

PURPOSE:

These keywords specify the endpoint values of the relative permeabilities for each grid block to scale the relative permeabilities of the associated table. Please refer to the tutorial section ‘Two and Three Point Scaling of Relative Permeability Values’ for a description of the scaling method.

ARRAY:

*KRWIRO
*KROCW
*KRGCL
*KROGCG
*KRWRO
*KROCRW
*KRGRL
*KROGCRG

DEFINITIONS:

Maximum Endpoint Values

***KRWIRO**

Relative permeability value of water at the irreducible oil saturation in the oil-water table (the last row in the table).

***KROCW**

Relative permeability value of oil at the connate water saturation in the oil-water table (the first row in the table).

***KRGCL**

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).

***KROGCG**

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).

Critical Endpoint Values

***KRWRO**

Relative permeability value of water at the residual oil saturation (Sorw) in the oil-water table.

***KROCRW**

Relative permeability value of oil at the critical water saturation (Swcrit) in the oil-water table.

***KRGRL**

Relative permeability value of gas at the residual liquid saturation (Slrg) in the gas-liquid table (See *SWCON Keyword for definition of Slrg).

***KROGCRG**

Relative permeability value of oil at the critical gas saturation (Sgcrit) in the gas-liquid table.

DEFAULTS:

Optional keywords. The default is to use the values from the relative permeability tables corresponding to the *RTYPE region to which the grid block belongs.

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, providing the critical endpoint also exists in the table. 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:

For the critical endpoint value scaling to be enabled it is crucial that the connate/irreducible endpoint saturations in the table be different from the critical/residual endpoint saturations.

That is: Sorw > Soirw, Swcrit > Swcon, Slrg > Slcon, and Sgcrit > Sgcon.

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

These keywords can be used in the Rock-Fluid Data keyword group after all the two phase relative permeability tables have been entered. Keywords can also be used in the recurrent data section.

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 tables. For critical endpoint values, the unassigned blocks will remain unspecified and will not be referenced.

For example, one can use *KRWRO to only assign Krwro values for blocks of rock type 1, leaving other blocks unspecified. Thus the blocks of rock type 1 will undergo three-point Krw scaling and the rest of the blocks will undergo two-point Krw scaling.

If entry by rock type input is used, then the *RTYPE array definition must be placed before any end point arrays are defined.

EXPLANATION:

The endpoint 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.

Once defined, these values remain in effect until explicitly overridden by another instance of these keywords. Changing a block's *RTYPE will change the relative permeability table of the block, but will not change the maximum relative permeability value of a block if the array has been defined for that block

In addition to the standard array reading options, the eight relative permeability end point value arrays can be entered according to rock type. The format used when using this entry technique is:

“End Point Value Array” *RTYPE “Rock Type Table Number” “End Point Value”

For example, if the value of *KRWIRO for table number 3 is to be changed to 0.80, the format would be:

*KRWIRO *RTYPE 3 0.80

This end point value array reading format searches all blocks whose rock type is 3 and assigns the water relative permeability at irreducible oil of those blocks to the value of 0.8. The result is identical to the result one would obtain if an *IJK array reading option was used to explicitly change each block whose *RTYPE was 3 to 0.80.

Multiple occurrences of this format are valid, for example:

*KRWIRO *RTYPE 1 0.70

*KRWIRO *RTYPE 2 0.90

*KRWIRO *RTYPE 3 1.00

In addition it is possible to define relative permeability end point array using a standard array reading option and override the input values for specific rock types using this format, for example:

*KRWIRO *IJK 1:5 1:10 1 0.90

*KRWIRO *RTYPE 3 0.80

Initially all block's water relative permeability at irreducible oil are set to the values defined by each rock type's relative permeability table (the default), the keywords above override blocks in the IJK range 1:5 1:10 1 with the water relative permeability at irreducible oil of 0.90 and further set the water relative permeability at irreducible oil of all blocks in rock type 3 to 0.80.

THE *OILWET OPTION:

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.

The use of the oil-wet option also changes the meaning of the user specified grid block specific end point arrays. For example, the *KRWIRO should be read as ‘maximum wetting phase relative permeability’, and the *KRWRO should be read as ‘wetting phase relative permeability at critical non-wetting phase saturation.’ Please refer to the Tutorial section for a discussion of the IMEX oil wet option.

Endpoint Capillary Pressure Values for Each Grid Block

(Optional)

*PCWMAX, *PCWMIN, *PCWIMIN, *PCGMAX, *PCGMIN, *PCGIMIN,
*JFWMAX, *JFGMIN, *JFWMAX, *JFGMIN

PURPOSE:

These keywords specify the endpoint values of the capillary pressure of each grid block. The table curves associated to the block is scaled based on these end points.

ARRAY:

*PCWMAX
*PCWMIN
*PCWIMIN
*PCGMAX
*PCGMIN
*PCGIMIN
*JFWMAX
*JFWMIN
*JFGMAX
*JFGMIN

DEFINITIONS:

*PCWMAX, *PCWMIN

Oil-water capillary pressure values at connate water saturation and irreducible oil saturation ($Sw=1-Soilw$) respectively. These keywords are used to represent gas-water capillary pressure if the GASWATER option is used and gas-liquid capillary pressure if the GASWATER_WITH_CONDENSATE option is used. Do not use if the J-Function option is specified.

*PCWIMIN

Oil-water imbibition capillary pressure value at the maximum residual oil saturation ($Sw=1-Sormax$). Keyword is used to represent gas-water capillary pressure if the GASWATER option is used and gas-liquid capillary pressure if the GASWATER_WITH_CONDENSATE option is used. Note that J-Function does not model imbibition capillary pressure.

*PCGMAX, *PCGMIN

Oil-gas capillary pressure values at the connate liquid saturation and connate gas saturation ($Sl=1-Sgcon$) respectively, in the gas-liquid system. Keywords have no effect if the GASWATER or GASWATER_WITH_CONDENSATE options are used. Do not use if the J-Function option is specified.

***PCGIMIN**

Oil-gas imbibition capillary pressure value at the connate gas saturation ($S_l=1-S_{gcon}$) in the gas-liquid system. Keyword has no effect if the GASWATER or GASWATER_WITH_CONDENSATE options are used. Note that J-Function does not model imbibition capillary pressure.

***JFWMAX, *JFWMIN**

Oil-water J-Function values at the connate water saturation and irreducible oil saturation ($1-S_{oilw}$) respectively. Keywords are used to represent Gas-water J-Function if the GASWATER option is used and gas-liquid J-Function if the GASWATER_WITH_CONDENSATE option is used. Only use if the J-Function option is specified.

***JFGMAX, *JFGMIN**

Oil-gas J-Function value at the connate liquid saturation and connate gas saturation ($1-S_{gcon}$) respectively, in the gas-liquid system. Keywords have no effect if the GASWATER or GASWATER_WITH_CONDENSATE options are used. Use only if the J function option is specified.

DEFAULTS:

Optional keywords. The default is to use the values from the relative permeability tables corresponding to the *RTYPE region to which the grid block belongs.

CONDITIONS:

These keyword can be in the Rock-Fluid Data keyword group after all the two phase relative permeability tables have been entered or in the recurrent data section.

Any or all of the above keywords may be omitted. If keywords are used to specify only part of the reservoir blocks, the rest of the blocks will be filled with default values. In the recurrent data section, it is not necessary to define all values, only those altered.

If entry by rock type input is used, then the *RTYPE array definition must be placed before any end point arrays are defined.

EXPLANATION:

The endpoint 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 keywords provide the capability to scale capillary pressure or J-function values at their end-points. There are two different ways to do this. The default is maximum factor scaling; optionally, two-point scaling is available.

When maximum factor scaling is used, only the P_c or J-Function values at the maximum end will be used. The minimum end points are only effective when two-point scaling is used. Refer to keyword *PCSCALE_2PT for the scaling method.

Capillary pressure end-points can be altered recurrently. Once defined these values remain in effect until explicitly overridden by another instance of these keywords. Changing a block's *RTYPE will change the relative permeability table of the block, but will not change the maximum capillary pressure value of a block if these arrays have been defined.

Examples:

Next keyword assigns maximum P_{cow} to 100.0 psia over all the reservoir blocks.

```
*PCWMAX *CON 100.0
```

Assume the reservoir has two rock types. Following keywords assign the maximum oil-water J-function to 6.0 for rock type 1 and 3.0 for rock type 2.

```
*JFWMAX *RTYPE 1 6.0
```

```
*JFWMAX *RTYPE 2 3.0
```

The following keywords will perform two-point scaling for imbibition capillary pressure P_{cowi} curves:

```
*PCSCALE_2PT
```

```
*PCWIMIN *CON -75.0
```

The above will not cause the scaling of drainage P_{cow} to change because none of the drainage-curve end points are assigned.

THE *OILWET OPTION:

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.

The use of the oil-wet option also changes the meaning of the user specified grid block specific maximum capillary pressure arrays. Please refer to the Tutorial section for a discussion of the IMEX oil wet option.

Three Point Saturation Scaling Option (Optional) *3PTSCALING

PURPOSE:

This keyword enables the three point scaling option for saturation endpoints.

FORMAT:

*3PTSCALING (*ON | *OFF)

DEFINITIONS:

*ON

Turn on the three point saturation scaling feature.

*OFF

Turn off the three point saturation scaling feature.

DEFAULTS:

Optional keyword. If *3PTSCALING is not include in the data set, the default is to not use three point scaling. If *3PTSCALING appears but is not followed by *ON or *OFF, *ON is assumed.

CONDITIONS:

This keyword must appear in the Rock-Fluid keyword group and applies to all rock types.

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_{rg} and K_{rog} , as described under the keyword explanation for “Saturation Endpoints for each Grid Block” is altered.

In the default “two point” scaling, relative permeability curves were scaled between two saturation endpoints. These were:

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 *3PTSCALING 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}	Scaled between S_{wcrit} and $(1-S_{orw})$, if block S_w is less than or equal to $(1-S_{orw})$, or between $(1-S_{orw})$ and $(1-S_{oirw})$, if S_w is greater than $(1-S_{orw})$.
K_{row}	Scaled between S_{wcon} and S_{wcrit} , if block S_w is less than or equal to S_{wcrit} , or between S_{wcrit} and $(1-S_{orw})$, if S_w is greater than S_{wcrit} .
K_{rg}	Scaled between S_{lcon} and S_{lrg} , if block S_l is less than or equal to S_{lrg} , or between S_{lrg} and $(1-S_{gcrit})$, if S_l is greater than S_{lrg} .

K_{rog}	Scaled between S_{lrg} and $(1-S_{gcrit})$, if block S_l is less than or equal to $(1-S_{gcrit})$ or between $(1-S_{gcrit})$ and $(1-S_{gcon})$, if S_l is greater than $(1-S_{gcrit})$.
-----------	---

It is important to ensure that when using three point scaling that 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 will only be done in the blocks affected and does not disable *3PTSCALING globally.

More generally the scaling from blocks 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 Krw (Sat_{ts})

$$Sat = S_w$$

For two point scaling:

$$Slow_{ts} = S'_{wcrit}, Slow_{bs} = S_{wcrit}, Supp_{ts} = 1 - S'_{oirw}, Supp_{bs} = 1 - S_{oirw}$$

For three point scaling:

$$\begin{aligned} \text{When } Sat_{bs} < 1 - S_{orw}: \quad Slow_{ts} = S'_{wcrit}, Slow_{bs} = S_{wcrit}, Supp_{ts} = 1 - S'_{orw}, \\ & Supp_{bs} = 1 - S_{orw} \end{aligned}$$

$$\begin{aligned} \text{When } Sat_{bs} > 1 - S_{orw}: \quad Slow_{ts} = 1 - S'_{orw}, Slow_{bs} = 1 - S_{orw}, Supp_{ts} = 1 - S'_{oirw}, \\ & Supp_{bs} = 1 - S_{oirw} \end{aligned}$$

For Krow (Sat_{ts})

$$Sat = S_w$$

For two point scaling:

$$Slow_{ts} = S'_{wcon}, Slow_{bs} = S_{wcon}, Supp_{ts} = 1 - S'_{orw}, Supp_{bs} = 1 - S_{orw}$$

For three point scaling:

$$\begin{aligned} \text{When } Sat_{bs} < S_{wcrit}: \quad Slow_{ts} = S'_{wcon}, Slow_{bs} = S_{wcon}, Supp_{ts} = S'_{wcrit}, Supp_{bs} = S_{wcrit} \\ \text{When } Sat_{bs} > S_{wcrit}: \quad Slow_{ts} = S'_{wcrit}, Slow_{bs} = S_{wcrit}, Supp_{ts} = 1 - S'_{orw}, \\ & Supp_{bs} = 1 - S_{orw} \end{aligned}$$

For Krog (Sat_{ts})

$$Sat = S_l$$

For two point scaling:

$$Slow_{ts} = S'_{lrg}, Slow_{bs} = S_{lrg}, Supp_{ts} = 1 - S'_{gcon}, Supp_{bs} = 1 - S_{gcon}$$

For three point scaling:

$$\begin{aligned} \text{When } Sat_{bs} < 1 - S_{gcrit}: \quad Slow_{ts} = S'_{lrg}, Slow_{bs} = S_{lrg}, Supp_{ts} = 1 - S'_{gcrit}, \\ & Supp_{bs} = 1 - S_{gcrit} \end{aligned}$$

$$\begin{aligned} \text{When } Sat_{bs} > 1 - S_{gcrit}: \quad Slow_{ts} = 1 - S'_{gcrit}, Slow_{bs} = 1 - S_{gcrit}, Supp_{ts} = 1 - S'_{gcon}, \\ & Supp_{bs} = 1 - S_{gcon} \end{aligned}$$

For Krg (Sat_{ts})

$$Sat = S_l$$

For two point scaling:

$$Slow_{ts} = S'_{lcon}, Slow_{bs} = S_{lcon}, Supp_{ts} = 1 - S'_{gcrit}, Supp_{bs} = 1 - S_{gcrit}$$

For three point scaling:

$$\text{When } Sat_{bs} < S_{lrg}: \quad Slow_{ts} = S'_{lcon}, Slow_{bs} = S_{lcon}, Supp_{ts} = S'_{lrg}, Supp_{bs} = S_{lrg}$$

$$\text{When } Sat_{bs} > S_{lrg}: \quad Slow_{ts} = S'_{lrg}, Slow_{bs} = S_{lrg}, Supp_{ts} = 1 - S'_{gcrit},$$

$$Supp_{bs} = 1 - S_{gcrit}$$

Two-Point Capillary Pressure Scaling (Optional) *PCSCALE_2PT

PURPOSE:

This keyword activates two-end-point capillary pressure scaling for both oil-water and oil-gas capillary pressure.

FORMAT:

*PCSCALE_2PT (*ON | *OFF)

DEFINITIONS:

*ON

Turns on two-point capillary pressure scaling.

*OFF

Turns off two-point capillary pressure scaling.

DEFAULTS:

Optional keyword. If *PCSCALE_2PT is not include in Rock-Fluid data, two-point Pc scaling will not be applied. If * PCSCALE_2PT appears but is not followed by *ON or *OFF, *ON is assumed.

CONDITIONS:

This keyword applies to all rock types and must appear in the Rock-Fluid keyword group.

EXPLANATION:

The default capillary pressure scaling is maximum factor scaling. For a capillary-pressure branch, if P_c is the block capillary pressure and P_{ct} is the calculated table value, the maximum factor scaling is:

$$P_c = P_{ct} \frac{P_{c_{max}}}{P_{c_{t, max}}}$$

where $P_{c_{max}}$ and $P_{c_{t, max}}$ are the capillary pressures at the connate water saturation for the block and table respectively.

When two-point capillary pressure scaling is on, users can assign both end-point P_c values for reservoir blocks. The capillary pressure calculated for the block will be scaled from the table value, as follows:

$$P_c = P_{c_{min}} + (P_{ct} - P_{c_{t, min}}) \frac{(P_{c_{max}} - P_{c_{min}})}{(P_{c_{t, max}} - P_{c_{t, min}})}$$

where subscripts max and min denote the maximum and minimum capillary pressure values. Once activated, two-point scaling applies for both oil-water and oil-gas capillary pressures and also applies for both drainage and imbibition branches if applicable.

Block maximum and minimum values are read from keywords *PCWMAX, *PCWMIN, *PCWIMIN, *PCGMAX, *PCGMIN, and *PCGIMIN.

Alternatively, if J-Function is specified in a *SWT table, the above scaling expressions are applicable for J-Function scaling, where P_c and P_{c_t} can be replaced by JF and JF_t to denote J-Function at block and table respectively. When J-Function is used, array keywords *JFWMAX, *JFWMIN, *JFGMAX, and *JFGMIN are effective.

Rock-Fluid Table-based Three-point Relative Permeability Scaling (Optional)

***TBLBAS_3PTKR**

PURPOSE:

This keyword switches the special treatment of three-point Kr scaling into table-based instead of the default block-based.

FORMAT:

***TBLBAS_3PTKR (*ON | *OFF)**

DEFINITIONS:

***ON**

Turn on the table-based three-point Kr scaling.

***OFF**

Turn off the table-based three-point Kr scaling.

DEFAULTS:

Optional keyword. If *TBLBAS_3PTKR is not include in the Rock-Fluid data, the default is to use the block end-points based three-point relative permeability scaling. If *TBLBAS_3PTKR appears but is not followed by *ON or *OFF, *ON is assumed.

CONDITIONS:

This keyword must appear in the Rock-Fluid keyword group and applies to all rock types.

EXPLANATION:

Table-based three-point Kr scaling differs from the default block-based scaling only at the handling of the next two special cases:

1. Table is two-point type however the block has the 3rd point:
Table-based treatment will retrograde to two-point Kr scaling and the block 3rd Kr point will be ignored.
2. Both table and block are three point type, however three-point saturation scaling option, *3PTSCALING, is off:
Table-based treatment follows the *3PTSCALING setting. Two-point saturation scaling will be applied and the table critical or residual saturation (e.g. Sorw in view of Krw) will be honored as the division point for Kr scaling.

For details of two- or three-point saturation and relative permeability scaling, please refer the keyword *3PTSCALING and tutorial ‘Two and Three Point Scaling of Relative Permeability Values’.

Extrapolation of Krw outside of the *SWT table to 1.0 (Optional)

***KRWEXTRAP**

PURPOSE:

This keyword enables the linear extrapolation of water relative permeability from the last water saturation in the oil-water table to a saturation of 1.0 and a value of Krw of 1.0.

FORMAT:

***KRWEXTRAP (*ON | *OFF)**

DEFINITIONS:

***ON**

Turn on the water relative permeability extrapolation feature.

***OFF**

Turn off the water relative permeability extrapolation feature.

DEFAULTS:

Optional keyword. If *KRWEXTRAP is not include in the data set, the default is to not use option. If *KRWEXTRAP appears but is not followed by *ON or *OFF, *ON is assumed.

CONDITIONS:

This keyword must appear in the Rock-Fluid keyword group and applies to all rock types. When using this option to linearly extrapolate K_{rw} to 1.0 at a water saturation of 1.0, from the last value of K_{rw} in the *SWT table, the following conditions must be true.

$$S_{orw} = S_{oirw}$$

$$S_{oirw} > 0.00$$

EXPLANATION:

When the *KRWEXTRAP option is used it is assumed that the table entered is valid for the oil zone, and so that S_{oirw} as read from the table or from endpoint scaling arrays is greater than 0.00. Implicit in the *KRWEXTRAP option is that $S_{oirw} = 0.0$, even though the table does not extend that far. In order to resolve this inconstancy, we expect the user to define the table such that S_{orw} must equal S_{oirw} (as read from the table or from endpoint scaling arrays). We will warn the user if this is not so. A maximum of twenty warning will be printed to the output file.

When a water saturation which is greater than the oil zone maximum water ($1-S_{oirw}$) is encountered, say in the water zone, K_{rw} will be extrapolated linearly beyond the end of the table from its value at $(1-S_{oirw})$ to the value of 1.0 at $S_w = 1.0$. Capillary pressures will be held constant at the value from the table at a saturation of $1-S_{oirw}$ to a saturation of $S_w = 1.0$.

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. This keyword can be used to account for the effects of a tilted water-oil contact (See Tutorial section).

ARRAY:

***PCOW_SHIFT
*JFW_SHIFT**

DEFINITIONS:

***PCOW_SHIFT**

*PCOW_SHIFT specifies the amount that a grid blocks water-oil capillary pressure curve is shifted up or down from the table value. PCOW_SHIFT is used to represent Gas-water capillary pressure if the GASWATER option is used and Gas-liquid capillary pressure if the GASWATER_WITH_CONDENSATE option is used. Do not use if the J Function option is specified.

***JFW_SHIFT**

*JFW_SHIFT specifies the amount that a grid blocks water-oil J Function curve is shifted up or down from the table value. JFW_SHIFT is used to represent the Gas-water J Function if the GASWATER option is used and the Gas-liquid J Function if the GASWATER_WITH_CONDENSATE option is used. Only use if the J Function option is specified.

DEFAULTS:

Optional keyword. The default is to zero.

CONDITIONS:

This keyword can be in the Rock-Fluid Data keyword group after all the two phase relative permeability tables have been defined. This keyword may only be used if the DEPTH_AVE initialization option is used.

EXPLANATION:

The PCOW_SHIFT keyword 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.

These values are added to the value of Pcow or water J functions. The shift is applied after the maximum capillary pressure/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 can be used to model tilting water-oil or water-gas contacts (see the Tutorial section on how to model tilting contacts).

Non-Darcy Flow in Reservoir (Optional)

*NONDARCY

PURPOSE:

*NONDARCY signals the use of the non-Darcy reservoir flow option.

FORMAT:

```
*NONDARCY ( *GEERTSMA |
    *FG1 |
    *FG2 |
    *GENERAL wt_ndarcy
    { Coefficient Table } )
```

DEFINITIONS:

*GEERTSMA

*GEERTSMA signals the use of Geertsma's (1974) correlation for gas phase non-Darcy flow.

*FG1

*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.

Background:

For any phase the Forchheimer equation can be written as

$$-\nabla \phi_p = \frac{\mu_p}{K K_{rp}} + \beta_p \rho_p |\bar{U}_p| \bar{U}_p$$

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:

$$-\nabla \phi_p = \frac{\mu_p}{K K_{rp}} (1 + F_o_p) \bar{U}_p$$

with

$$F_o_p = \beta_p K K_{rp} \rho_p |\bar{U}_p| / \mu_p$$

F_o is referred to as the Forchheimer Number.

The correlations for β_p are all of the form (K in md, and β_p in 1/ft.)

$$\beta_p = \frac{\alpha_p}{(K K_{rp})^{N1_p} (\phi S_p)^{N2_p}}$$

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:

*NODARCY *GENERAL		0.50		
gas	α_g	$N1_g$	$N2_g$	$Forch_{maxg}$
water	α_w	$N1_w$	$N2_w$	$Forch_{maxw}$
oil	α_o	$N1_o$	$N2_o$	$Forch_{maxo}$

For models with gas and water phases present (no oil or solvent) the form of the coefficient tables is:

*NODARCY *GENERAL		0.50		
gas	α_g	$N1_g$	N_g	$Forch_{maxg}$
water	α_w	$N1_w$	$N2_w$	$Forch_{maxw}$

For models with solvent phase present the form of the coefficient tables is:

*NODARCY *GENERAL		0.50		
gas	α_g	$N1_g$	$N2_g$	$Forch_{maxg}$
water	α_w	$N1_w$	$N2_w$	$Forch_{maxw}$
oil	α_o	$N1_o$	$N2_o$	$Forch_{maxo}$
solvent	α_s	$N1_s$	$N2_s$	$Forch_{maxs}$

For models with oil and water phases present (no gas or solvent) the form of the coefficient tables is:

*NODARCY *GENERAL		0.50		
water	α_w	$N1_w$	$N2_w$	$Forch_{maxw}$
oil	α_o	$N1_o$	$N2_o$	$Forch_{maxo}$

Note: A value of $N2_p = 0.0$ will eliminate the dependence of β_p on $(\phi \times S_p)$ and allow a correlation for β_p of the form:

$$\beta_p = \frac{\alpha_p}{(K K_{rp})^{N1_p}}$$

This form is common when modelling non Darcy flow in fractures.

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.

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 N_{1_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.

*NDARCYCOR may be redefined in recurrent data.

Fracture Velocity Width Correction for Polymer–Water Mixture Viscosity (Conditional)

***FRWIDTHCOR**

PURPOSE:

Allows the input of block by block correction factors for the modification of the velocity used to interpolate in the *PMIX *VELTABLE table.

ARRAY:

***FRWIDTHCOR**

DEFAULTS:

Conditional keyword. By default all values are 1.0.

CONDITIONS:

Used only when *PMIX *VELTABLE is input.

DEFINITION:

This array allows the user to modify the water velocity used to interpolate in the *PMIX *VELTABLE polymer–water mixture viscosity table.

It is envisioned that this correction factor would be used in blocks which are attempting to represent fractures. It is assumed that these simulated fracture blocks are small, on the order of 0.2 meter wide (for example), but are not small enough to accurately model the velocities in a fracture which (we will assume) has an actual fracture width of 0.1 millimeters.

The user might ensure that the actual fracture KA and the simulated fracture block KA (and hence the flow) are similar, but the velocity in the simulated fracture block would be lower by the ratio of the real fracture area to the simulated fracture block area (in this example 2,000.0 times lower)

A value of FRWIDTHCOR of 2,000.0 could be used to multiply the calculated velocity in a simulated fracture block by 2,000.0 before interpolating in the *PMIX *VELTABLE table to find the velocity dependent viscosity of the water–polymer mixture.

A value of FRWIDTHCOR = 0.0 in a grid block sets the velocity used in the *PMIX *VELTABLE viscosity calculation to 0.0. This is a convenient way to limit the region where the polymer – water viscosity calculation is a function of velocity.

A value of FRWIDTHCOR = 1.0 in a grid block sets the velocity used in the *PMIX *VELTABLE viscosity calculation to the intrinsic block velocity.

Pressure Gradient Threshold Arrays

*PTHRESHI, *PTHRESHJ,

*PTHREHK

PURPOSE:

Defines pressure gradients on interblock flow boundaries below which flow is not initiated.

ARRAY:

*PTHRESHI
*PTHRESHJ
*PTHREHK

DEFINITIONS:

*PTHRESHI

Pressure gradient threshold for the flow between block I,J,K and I+1,J,K.
(kPa/m | Psi/ft | kPa/cm | kg/cm²/m).

*PTHRESHJ

Pressure gradient threshold for the flow between block I,J,K and I,J+1,K.
(kPa/m | Psi/ft | kPa/cm | kg/cm²/m).

*PTHREHK

Pressure gradient threshold for the flow between block I,J,K and I,J,K+1.
(kPa/m | Psi/ft | kPa/cm | kg/cm²/m).

The pressure gradient threshold is a one time switch. Once it is exceeded between two blocks in a specific direction, the interblock connection remains open to flow ever more.

A zero value of the thresholds indicate the connection is always open to flow.

Note that the thresholds are defined on connections/interfaces between blocks and not on the blocks themselves.

DEFAULTS:

The defaults for all values are 0.0. This allows flow between connections without a pressure gradient to be overcome.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group. This keyword may be introduced in this section (not in recurrent data) at a restart to model injection followed by production in a fracture clean-up simulation (See Tutorial on Using the Fracture Clean-up Model).

EXPLANATION:

The pressure gradient thresholds are used to approximately model yield stress. Once the pressure gradient for flow has been exceeded on an inter-block connection, the threshold for that connection is not checked again.

The pressure gradient which initiates flow is corrected for the initial (usually time = 0.0) pressure in the reservoir. This prevents flow due to the difference in head in the reservoir.

When the pressure gradient threshold keywords are introduced on a restart, the initial reservoir pressure used to correct the pressure gradient is the pressure in the reservoir at the date of the first restart when the pressure gradient thresholds were introduced.

Leverett J Function Option

*JFUNC

PURPOSE:

Activates the Leverett J Function option. This option allows the user to replace water-oil and/or oil-gas capillary pressures entered with Leverett J Functions. When the *GASWATER model option is used, the entered water-gas capillary pressure is replaced by a water-gas J Function. When the *GASWATER_WITH_CONDENSATE model option is used, the entered liquid-gas capillary pressure is replaced by a liquid-gas J Function.

FORMAT:

*JFUNC (*BOTH | *WATER | *GAS)

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 which must precede all *RPT keywords.

*BOTH

Both water-oil and oil-gas capillary pressures are interpreted as J functions for all rock types defined (Default).

*GAS

Oil-gas capillary pressures are interpreted as J functions. Water-oil capillary pressures are still interpreted as capillary pressures.

*WATER

Water-oil capillary pressures are interpreted as J functions. Oil-gas capillary pressures are still interpreted as capillary pressures.

When using the *GASWATER model option, use of any of the *JFUNC, *JFUNC *BOTH, *JFUNC *WATER, or *JFUNC *GAS keywords will enable the interpretation of water-gas capillary pressures as J functions.

When using the *GASWATER_WITH_CONDENSATE model option, use of any of the *JFUNC, *JFUNC *BOTH, *JFUNC *WATER, or *JFUNC *GAS keywords will enable the interpretation of liquid-gas capillary pressures as J functions.

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.

CONDITIONS:

This keyword must be in the Rock-Fluid Data section and must be placed before the first *RPT keyword.

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 capillary pressure in the relative permeability table reading options (*SWT, *SGT, *SLT). When the *GASWATER model option is used, water-gas capillary pressure is interpreted as water-gas J function. Water-gas surface tension is entered using the *SRFTNW keyword.

When the *GASWATER_WITH_CONDENSATE model option is used, liquid-gas capillary pressure is interpreted as liquid-gas J function. Liquid-gas surface tension is entered using the *SRFTNW keyword.

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, normally $\sqrt{K_x * K_y}$ unless $\sqrt{K_x * K_y}$ is zero, then the largest of K_x , K_y , and K_z is used (md)
$J_{(SW)}$	Dimensionless J function entered in the *SWT table. This replaces the capillary pressure column for P_{cow} (or P_{cwg})
$J_{(Sg)}$ or $J_{(SI)}$	Dimensionless J function entered in the *SGT or *SLT table. This replaces the capillary pressure column for P_{cog}
*SRFTNW	Oil-water or water-gas 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:

$$P_{cow} = J_{(SW)} * SRFTNW * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

For the water-gas capillary pressure:

$$P_{cwg} = J_{(Sw)} * SRFTNW * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

Similarly for the oil-gas table:

$$P_{cog} = J_{(Sg)} * SRFTNG * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

or

$$P_{cog} = J_{(Sl)} * SRFTNG * \sqrt{\left(\frac{\phi}{K}\right)} * CF$$

If oil-gas surface tension is defined as a function of pressure by entering an appropriate nonzero column in one of the PVT table entry options and by entering a single reference pressure (*REFPST) to be used in all PVT regions, then the P_{cog} calculations for each block is corrected by:

$$\frac{PVT\ Table\ Surface\ tension\ (function\ of\ P\ and\ PVT\ region)}{PVT\ Table\ Surface\ tension\ at\ Reference\ pressure\ (function\ of\ *REFPST\ and\ PVT\ region)}$$

Note: Although there is a single reference pressure for surface tension which is shared by all PVT tables. Each table may have a different surface tension at the reference pressure.

J Function Surface Tension Arrays

*SRFTNW, *SRFTNG

PURPOSE:

Defines Surface tensions to be used in the J Function Option.

ARRAY:

*SRFTNW
*SRFTNG

DEFINITIONS:

*SRFTNW

In the calculation of P_{cow} or P_{cwg} (P_{clg} - condensate option) 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

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 keywords which are used only when the J function option is active. When the J function option is in use, the default values are 1.00. This implies 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.

When using the *GASWATER or *GASWATER_WITH_CONDENSATE option, only *SRFTNW can be used to enter surface tension.

Scale Deposition Table (Optional)

*SCLDPS

PURPOSE:

*SCLDPS indicates the start of a seawater option scale deposition table. It allows the user to enter scale deposition as a function of seawater volume fraction for individual scale deposition set numbers. These sets can then be applied to individual wells and/or well layers using *SCLtbl-WELL and SCLtbl-LAYER as described in the Recurrent Data section.

TABLE:

*SCLDPS	<i>table_set_number</i>
Seawater fraction	Precipitation
:	:

DEFINITIONS:

table_set_number

The set number of the table. Each well or layer of a well can be assigned to a deposition table through this set number. The first set number must be 1 and each additional deposition table must have a set number which increments by 1.

Seawater fraction

Seawater volume fraction. Seawater fraction must be no less than zero and no greater than one. In addition the seawater fraction must monotonically increase down the table's rows (fraction).

Precipitation

Amount of scale deposition per unit flow rate of water flow though a well perforation (kg/m^3 | lb/bbl | g/cm^3 | kg/m^3).

DEFAULTS:

Optional keyword. There are no defaults.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group.

EXPLANATION:

This table defines the relationship between the seawater volume fraction and the amount of scale precipitated per unit flow rate of water produced.

The amount of scale deposited on a perforation over a timestep is calculated as the product of the amount of scale precipitated per unit flow rate of water and the water flow through the perforation. If the *BLACKOIL_SEAWATER or *OILWATER_SEAWATER options are used, the amount precipitated per unit flow rate of water is also a function of seawater volume fraction produced. Without the seawater injection (tracking) options, the tables will always use the 0.0 volume fraction seawater value of precipitation to work out the amount deposited over a timestep.

Scale Damage Table (Optional)

*SCLDMG

PURPOSE:

*SCLDMG indicates the start of a seawater option scale damage table. It allows the user to enter scale well damage as a function of scale deposited per perforation length for individual scale damage set numbers. These sets can then be applied to individual wells and/or well layers using *SCLTBL-WELL and SCLTBL-LAYER as described in the Recurrent Data section.

TABLE:

*SCLDMG <i>table_set_number</i>	
<i>Scale Deposited</i>	<i>Damage Factor</i>
:	:

DEFINITIONS:

table_set_number

The set number of the table. Each well or layer of a well can be assigned to a damage table through this set number. The first set number must be 1 and each additional deposition table must have a set number which increments by 1.

Scale Deposited

The amount of scale deposited per perforation length (kg/m | lb/ft| g/cm| kg/m).

Damage Factor

The layer or well index multiplier which represents a reduction in productivity caused by the amount of scale deposited (fraction)

DEFAULTS:

Optional keyword. There are no defaults.

CONDITIONS:

This keyword must be in the Rock-Fluid Data keyword group.

EXPLANATION:

When seawater and formation water flow simultaneously into a well, scale is precipitated. The amount of scale deposited is a function of rate of water flow through the perforations and the relative amounts of formation water and seawater. The amount of scale which is deposited accumulates through the simulation (unless removed using the *SCLRMV-WELL keyword in the Recurrent Data section). This table refines the relationship between the cumulative scale deposited per perforation length and the well damage.

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 P_{cog} and P_{cow} using the following formula (for each subdomain's lowest block).

A scale factor (PCSCALE) is calculated for both oil-water and oil-gas capillary pressure in the lowest subdomain block.

$$\text{PCSCALE} = 1.00 - \text{*SD_REINF}$$

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	Lab
min	0.0	0.0	0.0
max	1.0	1.0	1.0

Adsorption Parameters for each Gridblock (Optional)

***ADGMAXV, *ADGCSTV, *ROCKDEN, *ADGPCRIT**

PURPOSE:

These keywords specify the Langmuir Isotherm parameters used to model the adsorption/desorption of gas to/from rock. This feature can be used to model simple adsorption effects in shale gas and CBM problems.

ARRAY:

***ADGMAXV
*ADGCSTV
*ROCKDEN
*ADGPCRIT**

DEFINITIONS:

***ADGMAXV**

Specifies the maximum volume of adsorbed gas per unit mass of rock (sm³/kg | sft³/lbm | scm³/gm).

***ADGCSTV**

Specifies the inverse pressure parameter for the Langmuir Isotherm (kPa⁻¹ | Psi⁻¹ | kPa⁻¹ | (kg/cm²)⁻¹).

***ROCKDEN**

Specifies the intrinsic rock density, where intrinsic density = bulk density / (1- ϕ) (kg/m³ | lbm/ft³ | g/cm³).

***ADGPCRIT**

Specifies the critical adsorption pressure, for pressures above ADGPCRIT, ADGPCRIT is used instead of block pressure in the Langmuir Isotherm equation (kPa | Psi | kPa | kg/cm²).

DEFAULTS:

Optional keyword. The default is to not model gas adsorption. Only ADGMAXV and ADGCSTV are required input to model adsorption. If not input, ROCKDEN is defaulted to 2650.0 kg/m³. If not input, ADGPCRIT is defaulted to 10¹⁰ kPa, rendering the Langmuir Isotherm independent of ADGPCRIT.

CONDITIONS:

These keywords can be in the Rock-Fluid Data keyword group only.

Adsorption is not enabled unless both ADGMAXC and ADGCSTV arrays are specified. The run will terminate if only one of the two required arrays are input.

If adsorption array entry by rock type input is used, then the *RTYPE array definition must be placed before any adsorption arrays (*ADGMAXC etc) are defined.

EXPLANATION:

The adsorption 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.

In addition to the standard array reading options, the four adsorption arrays can be entered according to rock type. The format when using this input technique is:

“Adsorption Array Name” *RTYPE “Rock Type Table number” “Adsorption Array Value”

For example, if the value of ADGMAXV for blocks corresponding to table number 3 is to be changed to 0.06, the format would be:

*ADGMAXV *RTYPE 3 0.06

This adsorption array reading format searches all blocks whose rock type is 3 and assigns the ADGMAXV of those blocks to the value of 0.06. The result is identical to the result one would obtain if an *IJK array reading option was used to explicitly change each block whose *RTYPE was 3 to 0.06.

Multiple occurrences of this format are valid, for example:

*ADGMAXV *RTYPE 1 0.05

*ADGMAXV *RTYPE 2 0.04

*ADGMAXV *RTYPE 3 0.00

In addition, it is possible to define an adsorption array using a standard array reading option and override the input values for specific rock types using this format, for example:

*ADGMAXV *IJK 1:5 1:10 1 0.05

*ADGMAXV *RTYPE 3 0.06

The keywords above initially set blocks in the IJK range 1:5 1:10 1 to the ADGMAXV value of 0.05, the second line resets the ADGMAXV of all blocks in rock type 3 to 0.06.

LANGMUIR ADSORPTION ISOTHERM

The basic Langmuir Isotherm for gas adsorption used in IMEX is:

$$Ads = (1 - \phi) \times ROCKDEN \times ADGMAXV \times \left(\frac{ADGCSTV \times P_{AD}}{1 + ADGCSTV \times P_{AD}} \right)$$

Where P_{AD} = block pressure, if the block pressure is less than or equal to ADGPCRIT and $P_{AD} = ADGPCRIT$, if the block pressure is greater than ADGPCRIT. The term, $(1 - \phi) \times ROCKDEN$, converts intrinsic density ROCKDEN to a bulk density.

If the Langmuir maximum gas term is given in terms of gas volume per bulk volume, then the bulk rock density must be used to obtain ADGMAXV in terms of gas volume per unit mass of rock.

If ADS_WATERZONE is specified in the Component Properties section, the basic Langmuir equation in each block is multiplied by the gas saturation in a block, so that $Ads = Ads \times S_g$. This ensures that there is no gas adsorbed onto the rock in the water zone.

Phase Segregation Model Modifier (Optional)

***PHS_SEG_MODL**

PURPOSE:

This keyword modifies the behavior of the *TRANSFER 1 (see Reservoir Description), complete phase segregation model.

FORMAT:

***PHS_SEG_MODL (*ORIGINAL | *MODIFIED)**

DEFINITIONS:

***ORIGINAL**

Maintains the original behavior of the *TRANSFER 1 complete phase segregation model.

***MODIFIED**

Modifies the behavior of the *TRANSFER 1 complete phase segregation model.

DEFAULTS:

Optional keyword. If *PHS_SEG_MODL is not include in the data set, the default is to use the original behavior of the TRANSFER 1 option.

CONDITIONS:

This keyword must appear in the Rock-Fluid keyword group and only has an effect if the TRANSFER 1 complete segregation model is used with the *DUALPOR or *DUALPERM option.

EXPLANATION:

When the *TRANSFER 1 option is used, a pseudo capillary term is calculated which reflects the difference in pressure between the fracture and the matrix, assuming complete gravity segregation of fluids. The pressure drop calculation uses DKFRAC to represent the region of vertical capillary continuity within each block (i.e. the spacing between horizontal fractures). As long as DKFRAC is less than DK (the block thickness) DKFRAC is used directly.

When DK is less than DKFRAC, the height used in calculating the pressure drop is limited to DK. This assumption is reasonable as a matrix block in a dual porosity model is the maximum region of vertical capillary continuity in a block (there is no matrix-matrix flow between blocks areally or vertically). It may be reduced by DKFRAC less than DK, but not made larger than DK.

When *PHS_SEG_MODL *MODIFIED is used, this limitation is no longer enforced and the DKFRAC used in the pressure drop calculation may be larger than the block thickness (DK).

This feature is useful when the user is required to modify the grid thickness to be lower than DKFRAC, but does not want to additionally alter the spacing between vertical fractures used in the gravity segregation model.

When the *ORIGINAL option is used, the height of the capillary continuous region will change with DK, when DK is less than DKFRAC. When the *MODIFIED option is used, the height of the capillary continuous region is independent of DK when DK is less than DKFRAC.

Initial Conditions

Notes on Initial Conditions

IMEX currently provides seven different ways to initialize the reservoir.

1. *USER_INPUT, formerly *VERTICAL *OFF, which is still accepted by IMEX. Pressure, water saturation and oil saturation are specified for each grid block. Gas saturations are then determined by subtraction. Use the following keywords for this option:

*PRES or *PREST, *PB or *PBT, *RST, *SW and *SO (*API or *APIT if the API tracking model is used, *PDEW, *PDEWT, *RVT, if the *VOLATILE_OIL or *GASWATER_WITH_CONDENSATE model is used).

2. *VERTICAL *BLOCK_CENTER *WATER_OIL formerly *VERTICAL *ON, which is still accepted by IMEX, with gas-oil contact depth set outside the reservoir. In this approach the water-oil contact depth, together with a reference pressure at a reference depth are specified. Bubble point pressure for each grid block or as a table versus depth must be assigned by the user. Saturations for entire grid blocks are set equal to their values at the grid block center.

Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and saturations.

For this approach use the following keywords:

*DWOC, *REFDEPTH, *REFPRES, *PB or *PBT*/RST (*API or *APIT if the API tracking model is used, *PDEW or *PDEWT/*RVT if the *VOLATILE_OIL model is used).

3. *VERTICAL *BLOCK_CENTER *WATER_GAS, formerly *VERTICAL *ON, which is still accepted by IMEX, with water-oil and gas/oil contact depths set equal. This is 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. Saturations for entire grid blocks are set equal to their values at the grid block center.

Gravity-capillary equilibrium calculations are performed to calculate all grid block pressures and saturations.

For this approach use the following keywords:

*DWGC, *REFDEPTH, *REFPRES, *PB or *PBT*/RST (*PDEW or *PDEWT/*RVT – *GASWATER_WITH_CONDENSATE model).

4. *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS, formerly *VERTICAL *ON, which is still accepted by IMEX. The water-oil contact and gas-oil contact depth, together with a reference pressure at a reference depth must be specified. Saturations for entire grid blocks 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.

For this approach use the following keywords:

*DWOC, *DGOC, *REFDEPTH, *REFPRES, *PB or *PBT/*RST (*API or *APIT if the API tracking model is used, *PDEW or *PDEWT/*RVT if the *VOLATILE_OIL model is used).

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. Note that the *GASWATER and *GASWATER_WITH_CONDENSATE options require the use of the *WATER_GAS initialization option.

5. *VERTICAL *DEPTH_AVE *WATER_OIL. Gravity initialization for an undersaturated 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 (2) above (*VERTICAL *BLOCK_CENTER *WATER_OIL).

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 this approach use the following associated keywords:

*DWOC, *REFDEPTH, *REFPRES, *PB or *PBT/*RST (*API or *APIT if the API tracking model is used, *PDEW or *PDEWT/*RVT if the *VOLATILE_OIL model is 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 (4) above (*VERTICAL *BLOCK_CENTER *WATER_OIL_GAS).

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 this approach use the following associated keywords:

*DWOC, *DGOC, *REFDEPTH, *REFPRES, *PB or *PBT/*RST (*API or *APIT if the API tracking model is used, *PDEW or *PDEWT/*RVT if the *VOLATILE_OIL model is 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.

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.

In order to properly account for the effects of gas-water capillary pressure (Pcgw), the *TRANSZONE subkeyword must be used with *VERTICAL *DEPTH_AVE *WATER_GAS.

Note that the *GASWATER and *GASWATER_WITH_CONDENSATE options require the use of the *WATER_GAS initialization option.

For this approach use the following associated keywords:

*DWGC, *REFDEPTH, *REFPRES, *PB or *PBT/*RST (*PDEW or *PDEWT/*RVT – *GASWATER_WITH_CONDENSATE model).

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.

User Input or Vertical Equilibrium Selection (Required)

*USER_INPUT, *VERTICAL

PURPOSE:

*USER_INPUT and *VERTICAL 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.

*USER_INPUT

Specify pressure and saturations of each grid block. Use this subkeyword only when the pressure and saturations 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 IMEX.

*BLOCK_CENTER

Sub option of *VERTICAL. Assign block saturations as the saturation prevailing at the block center.

*DEPTH_AVE

Sub option of *VERTICAL. Assign block saturations as averages 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.

***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 using the *OUTPRN *RES *ALL keywords in the Input/Output Control section.

***NOEQUIL**

Do not add a phase pressure correction when a *DEPTH_AVE option is used.

***TRANZONE**

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 1.0 and Swcon (It is assumed for the *WATER_GAS option that *SOIRW and *SORW are both equal to 0.0). Note that *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 the *WATER_GAS subkeyword, then *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.

DEFAULTS:

One of *USER_INPUT or *VERTICAL is required. No default.

If *VERTICAL is entered without any subkeyword, *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS 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.

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 conditions of the reservoir are to be determined.

***USER_INPUT**

Under *USER_INPUT (formerly *VERTICAL *OFF, which is still accepted by IMEX), pressure and saturations at each grid block must be specified by the user under the *PRES (or *PREST), *SW, *SO keywords. Gas saturations are then determined by subtraction.

***VERTICAL *BLOCK_CENTER**

Under *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS (formerly *VERTICAL *ON, which is still accepted by IMEX), the water-oil-contact depth (*DWOC), the gas-oil-contact depth, together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. Bubble point pressures for the reservoir must also be specified by using either *PBT/*RST or *PB. *API or *APIT must also be specified if the API tracking model is used (*API-INT).

When using the *GASWATER; *PBT/*RST or *PB need not be input.

To use this option to initialize a gas reservoir, set *DGOC = *DWOC or use *VERTICAL *BLOCK_CENTER *WATER_GAS as outlined below.

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 will be identical from those obtained using *VERTICAL *BLOCK_CENTER *WATER_OIL.

Under *VERTICAL *BLOCK_CENTER *WATER_GAS, the water-gas contact depth (*DWGC) together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. Bubble point pressures for the reservoir must also be specified by using either *PBT/*RST or *PB unless the *GASWATER or *GASWATER_WITH_CONDENSATE option is used.

When using the *GASWATER or *GASWATER_WITH_CONDENSATE *MODEL option, *PBT/*RST or *PB need not be input. In addition, the *WATER_GAS initialization option and the *DWGC keyword must be used to define the gas water contact.

Under *VERTICAL *BLOCK_CENTER *WATER_OIL (formerly *VERTICAL *ON, which is still accepted by IMEX, with the gas-oil contact depth set outside the reservoir), the water-oil contact (*DWOC), together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. Bubble point pressures for the reservoir must also be specified by using either *PBT/*RST or *PB. *API or *APIT must also be specified if the API tracking model is used (*API-INTOW).

*VERTICAL *DEPTH_AVE

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. Bubble point pressures for the reservoir must also be specified by using either *PBT/*RST or *PB. *API or *APIT must also be specified if the API tracking model is used (*API-INTOW). This option 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 (*TRANZONE | *NOTRANZONE), the water-gas contact depth (*DWGC) together with a reference pressure (*REFPRES) at a reference depth (*REFDEPTH) must be specified. Bubble point pressures for the reservoir must also be specified by using either *PBT/*RST or *PB, unless the *GASWATER or *GASWATER_WITH_CONDENSATE options are used. When using the *GASWATER or *GASWATER_WITH_CONDENSATE *MODEL option, *PBT/*RST or *PB need not be input. In addition, the *WATER_GAS initialization option and the *DWGC keyword must be used to define the gas water contact.

This option 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.

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. Bubble point pressures for the reservoir must also be specified by using either *PBT/*RST or *PB. *API or *APIT must also be specified if the API tracking model is used (*API-INT). 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.

When using the *GASWATER_WITH_CONDENSATE model, *PDEW or *PDEWT/*RVT also must be entered. When using the *VOLATILE_OIL model, *PDEW or *PDEWT/*RVT must also be entered.

See the notes in the beginning of this section for more information.

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
```

Initial Oil Phase Reservoir Pressure (Conditional)

*PRES

PURPOSE:

*PRES indicates the input of reservoir pressures (kPa | psi | kPa | kg/cm²).

ARRAY:

*PRES

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. Either *PRES or *PREST is required with *USER_INPUT. *PRES and *PREST cannot both appear in a data set.

EXPLANATION:

The *PRES keyword defines the initial reservoir block pressures (of the oil phase). Any of the array reading options can be used.

The acceptable range of values for block pressures is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm ²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

Initial Pressure vs. Depth (Conditional)

*PREST

PURPOSE:

*PREST indicates the input of pressure (oil phase) in tabular format as a function of depth.

TABLE:

*PREST	<i>set_number</i>
<i>depth</i>	<i>P</i>
:	:

DEFINITIONS:

set_number

Set number for this set of pressure vs. depth table. This is the number of the initialization region used with *PTYPE or *ITYPE to assign initial pressures to grid blocks. The *set_number* must correspond to the *set_number* specified with *PVT or *NREGIONS and must increase monotonically.

depth

Depth (m | ft | cm)

P

Pressure (kPa | psi | kPa | kg/cm²)

DEFAULTS:

No default values.

CONDITIONS:

Required keyword when using the *USER_INPUT initialization option unless the *PRES keyword is input. This keyword must be in the Initial Conditions keyword group and must consist of two columns of numerical entries. The number of *PREST keywords must be equal to the number of PVT regions in the Component Properties section, if *NREGIONS is not defined, or *NREGIONS, if *NREGIONS is defined. Each initialization region can have at most one Pressure vs. Depth table.

EXPLANATION:

The *PREST keyword defines the initial pressure as a function of depth for an initialization region. Values of pressure are then assigned to each grid block by use of the *PTYPE or *ITYPE keyword.

Examples:

```
*PREST 1
** depth Pressure
 3000.0  1200.0
 3200.0  1300.0
```

The acceptable range of values for pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

The acceptable range of values for depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

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 set equal to the number of PVT regions and ITYPE must not be read, as PTYYPE is used to assign initialization regions (as in IMEX 2012.10 and earlier).

CONDITIONS:

This keyword must be in the INITIAL CONDITIONS keyword group, and must precede all keywords which refer to *nreg*, namely the following:

*ITYPE, *GOC_PC, *WOC_PC, *PBT/*RST, *PDEWT/*RVT, *APIT, *REFDEPTH,
*REFPRES, *GOC_SW, *WOC_SW, *DWOC, *DGOC, *DWGC, *DATUMDEPTH,

Specification of Initialization Regions (Optional)

*ITYPE

PURPOSE:

*ITYPE introduces an array of 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 and “*NREGIONS 1” has been read, it is assumed that all grid blocks are in initialization region 1. If no blocks are assigned to a properly defined initialization region, an informational message will be issued, and the empty initialization region is associated with the PVT properties of PVT region 1.

If *NREGIONS is not read, then the number of initialization regions is equal to the number of PVT regions, reading the ITYPE keyword in this situation will trigger an error (PTYPE is used to assign initialization regions to blocks).

CONDITIONS:

If it appears, this keyword must be in the INITIAL CONDITIONS keyword group, and must follow the *NREGIONS keyword.

Note: Within each initialization region there must only be a single PVT region. If more than one PVT region is found within an initialization region, the simulation will stop.

Example:

For a $5 \times 5 \times 4$ 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
```

Initial Bubble Point Pressure (Conditional)

*PB

PURPOSE:

*PB indicates the input of bubble point pressure (kPa | psi | kPa | kg/cm²) in array format.

ARRAY:

*PB

DEFAULTS:

Required keyword unless using PVT models discussed in CONDITIONS or *PBT/*RST. No default values.

If zero is entered, then the block bubble-point pressure is initialized to the block pressure if any of the *VERTICAL options are used. If *USER_INPUT initialization is used *PB must be explicitly set, and input of zero does not default to block pressure.

If the fluid model is *MODEL *OILWATER, a fixed bubble-point pressure is used. The bubble-point pressure is set to the first pressure value in the *PVT table if *PB is not found.

When using the *BLOCK_CENTER or DEPTH_AVE vertical equilibrium models with PVT models which allow a gas phase to form (i.e. not *OILWATER etc). Setting PB CON 0.0 is a special flag which sets every block in the reservoir to its bubble point (i.e. Pb = P, where P is the pressure determined from the *BLOCK_CENTER or DEPTH_AVE vertical equilibrium).

If multiple initialization regions are specified (either using *PTYPE or *NREGIONS and *ITYPE) and the initial bubble point pressure is set with *PB and *VERTICAL *BLOCK_CENTER is enabled, the bubble point pressure used to calculate pressures at the woc's and goc's is the value for the grid block closest to the WOC and the GOC. If bubble point pressure varies as a function of depth, use *PBT/*RST instead of *PB.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is required unless *MODEL *OILWATER, *MODEL *GASWATER, *MODEL *GASWATER_WITH_CONDENSATE or *PBT/*RST keywords are used.

If the *GASWATER or *GASWATER_WITH_CONDENSATE models are used, it is not necessary to define *PB or *PBT/*RST.

EXPLANATION:

The *PB keyword defines the initial bubble point pressure of each grid block. Any of the array reading options may be used.

The acceptable range of values for bubble point pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm ²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

Initial Bubble Point Pressure vs. Depth (Conditional)

*PBT

PURPOSE:

*PBT indicates the input of bubble point pressure in tabular format as a function of depth.

TABLE:

*PBT	<i>set_number</i>
<i>depth</i>	<i>Pb</i>
:	:

DEFINITIONS:

set_number

Set number for this set of bubble-point pressure vs. depth table. This is the number of the initialization region used with *PTYPE or *ITYPE to assign initial bubble point pressures to grid blocks. The *set_number* must correspond to the *set_number* specified with *PVT or *NREGIONS and must increase monotonically.

depth

Depth (m | ft | cm)

Pb

Bubble point pressure (kPa | psi | kPa | kg/cm²)

DEFAULTS:

Required keyword unless using the models discussed in CONDITIONS or *PB. No default values.

This keyword is to be used only when the bubble-point pressure is a function of depth. If the bubble-point pressure is constant, use *PB instead.

If the fluid model is *MODEL *OILWATER, a fixed bubble-point pressure is used. The bubble-point pressure is set to the first pressure value in the *PVT table, if *PBT/*RST is not specified.

CONDITIONS:

Required unless *MODEL *OILWATER, *PB, *RST or the two gas-water options are used. This keyword must be in the Initial Conditions keyword group and must consist of two columns of numerical entries. The number of *PBT keywords must be equal to the number of PVT regions in the Component Properties section, if *NREGIONS is not defined, or *NREGIONS, if *NREGIONS is defined. Each initialization region can have at most one Pb vs. Depth table.

If the *GASWATER or *GASWATER_WITH_CONDENSATE options are used, it is not necessary to define *PB or *PBT/*RST.

EXPLANATION:

The *PBT keyword defines the initial bubble point pressure as a function of depth for a PVT region. Values of bubble point pressure are then assigned to each grid block by use of the *PTYPE or *ITYPE keyword.

Examples:

```
*PBT 1
** depth Pb
 3000.0 1200.0
 3200.0 1300.0
```

The acceptable range of values for bubble point pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

The acceptable range of values for depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Note that the reservoir depth and the depth for the woc and goc must fall within the depths specified in the table.

Initial Solution Gas Ratio vs. Depth (Conditional)

*RST

PURPOSE:

*RST indicates the input of solution gas ratio in tabular format as a function of depth.

TABLE:

*RST	<i>set_number</i>
<i>depth</i>	<i>Rs</i>
:	:

DEFINITIONS:

set_number

Set number for this set of solution gas ratio vs. depth table. This is the number of the initialization region used with *PTYPE or *ITYPE to assign initial bubble point pressures to grid blocks (converted from solution gas ratio). The *set_number* must correspond to the *set_number* specified with *PVT or *NREGIONS and must increase monotonically.

depth

Depth (m | ft | cm)

Rs

Solution Gas Ratio (m³/m³ | scf/STB | cm³/cm³)

DEFAULTS:

Required keyword unless using the models discussed in CONDITIONS or *PB. No default values.

This keyword is to be used only when the initial solution gas ratio is a function of depth. If the solution gas ratio is constant, use *PB instead.

If the fluid model is *MODEL *OILWATER, a fixed bubble-point pressure is used. The bubble-point pressure is set to the first pressure value in the *PVT table, if *PBT/*RST is not specified.

CONDITIONS:

Required unless *MODEL *OILWATER, *PB, *PBT or the two gas-water options are used. This keyword must be in the Initial Conditions keyword group and must consist of two columns of numerical entries. The number of *RST keywords must be equal to the number of PVT regions in the Component Properties section, if *NREGIONS is not defined, or *NREGIONS, if *NREGIONS is defined. Each initialization region can have at most one Rs vs. Depth table.

*RST cannot be used with any of the API tracking models, use *PBT instead.

If the *GASWATER or *GASWATER_WITH_CONDENSATE options are used, it is not necessary to define *PB or *PBT/*RST.

In order to use this option Rs must increase monotonically with bubble point pressure.

EXPLANATION:

The *RST keyword defines the initial solution gas ratio as a function of depth for an initialization region. Values of bubble point pressure (converted from solution gas ratio) are then assigned to each grid block by use of the *PTYPE or *ITYPE keyword.

Examples:

```
*RST 1
** depth Rs
 3000.0  500.0
 3200.0  400.0
```

The acceptable range of values for bubble point pressure (calculated from Rs) is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

The acceptable range of values for depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Note that the reservoir depth and the depth for the woc and goc must fall within the depths specified in the table.

Initial Dew Point Pressure (Conditional)

*PDEW

PURPOSE:

*PDEW indicates the input of dew point pressure (kPa | psi | kPa | kg/cm²) in array format.

ARRAY:

*PDEW

DEFAULTS:

Required keyword only when using *MODEL *GASWATER_WITH_CONDENSATE or *MODEL *VOLATILE_OIL, unless the *PDEWT/*RVT option is used instead. No default values.

If zero is entered, then the block dew point pressure is initialized to the block pressure if any of the *VERTICAL options are used. If *USER_INPUT initialization is used, *PDEW must be explicitly set, and input of zero does not default to block pressure.

If multiple initialization regions are specified and the initial dew point pressure is set with *PDEW and *VERTICAL *BLOCK_CENTER is enabled, the dew point pressure used to calculate pressures at the gas water contact is the value for the grid block closest to the gas water contact. If dew point pressure varies as a function of depth, use *PDEWT/*RVT instead of *PDEW.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is required only when *MODEL *GASWATER_WITH_CONDENSATE or *MODEL *VOLATILE_OIL is used. *PDEW cannot be used when the *PDEWT/*RVT option is used.

EXPLANATION:

The *PDEW keyword defines the initial dew point pressure of each grid block. Any of the array reading options may be used.

The acceptable range of values for dew point pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

Initial Dew Point Pressure vs. Depth (Conditional) *PDEWT

PURPOSE:

*PDEWT indicates the input of dew point pressure in tabular format as a function of depth.

TABLE:

*PDEWT	<i>set_number</i>
<i>depth</i>	<i>Pdew</i>
:	:

DEFINITIONS:

set_number

Set number for this set of dew point pressure vs. depth table. This is the initialization region number used with *PTYPE or *ITYPE to assign initial dew point pressures to grid blocks. The *set_number* must correspond to the *set_number* specified with *PVT or *NREGIONS and must increase monotonically.

depth

Depth (m | ft | cm)

Pdew

Dew point pressure (kPa | psi | kPa | kg/cm²)

DEFAULTS:

Required keyword when using *MODEL *GASWATER_WITH_CONDENSATE or *MODEL *VOLATILE_OIL unless the *PDEW/*RVT option is used instead. No default values.

This keyword to be used only when the dew point pressure is a function of depth. If the dew point pressure is constant, use *PDEW instead.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is required only when *MODEL *GASWATER_WITH_CONDENSATE or *MODEL *VOLATILE_OIL option is used. *PDEWT cannot be used when the *PDEW/*RVT option is used. The number of *PDEWT keywords must be equal to the number of PVT regions in the Component Properties section, if *NREGIONS is not defined, or *NREGIONS, if *NREGIONS is defined. Each initialization region can have at most one Pdew vs. Depth table.

EXPLANATION:

The *PDEWT keyword defines the initial dew point pressure as a function of depth for an initialization region. Values of dew point pressure are then assigned to each grid block by use of the *PTYPE or *ITYPE keyword.

Examples:

```
*PDEWT 1  
** depth      Pdew  
 3000.0    1200.0  
 3200.0    1300.0
```

The acceptable range of values for dew point pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

The acceptable range of values for depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Note that the reservoir depth and the depth for the water gas contact must fall within the depths specified in the table.

Initial Oil Content (Rv) vs. Depth (Conditional)

*RVT

PURPOSE:

*RVT indicates the input of oil content in tabular format as a function of depth.

TABLE:

*RVT	<i>set_number</i>
<i>depth</i>	<i>Rv</i>
:	:

DEFINITIONS:

set_number

Set number for this set of oil content vs. depth table. This is the initialization region number used with *PTYPE or *ITYPE to assign initial dew point pressures to grid blocks. The *set_number* must correspond to the *set_number* specified with *PVT or *NREGIONS and must increase monotonically.

depth

Depth (m | ft | cm)

Rv

Oil Content (m^3/m^3 | STB/scf | cm^3/cm^3)

DEFAULTS:

Required keyword when using *MODEL *GASWATER_WITH_CONDENSATE or *MODEL *VOLATILE_OIL unless the *PDEW/*PDEWT option is used instead. No default values.

This keyword to be used only when the oil content is a function of depth. If the oil content (dew point pressure) is constant, use *PDEW instead.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is required only when *MODEL *GASWATER_WITH_CONDENSATE or *MODEL *VOLATILE_OIL option is used. *RVT cannot be used when the *PDEW/*PDEWT option is used. The number of *RVT keywords must be equal to the number of PVT regions in the Component Properties section, if *NREGIONS is not defined, or *NREGIONS, if *NREGIONS is defined. Each initialization region can have at most one Rv vs. Depth table.

EXPLANATION:

The *RVT keyword defines the initial oil content as a function of depth for an initialization region. Values of dew point pressure (calculated from Rv) are then assigned to each grid block by use of the *PTYPE or *ITYPE keyword.

Examples:

```
*RVT 1  
** depth     Rv  
 3000.0    0.000014  
 3200.0    0.000013
```

The acceptable range of values for dew point pressure is (calculated from Rv):

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

The acceptable range of values for depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Note that the reservoir depth and the depth for the water gas contact must fall within the depths specified in the table.

Initial Solvent Bubble Point Pressure (Conditional)

*PBS

PURPOSE:

*PBS indicates input of solvent bubble point pressure (kPa | psi | kPa | kg/cm²).

ARRAY:

*PBS

DEFAULTS:

Conditional keyword. Default is 0.0 for all grid blocks.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. Used with *MODEL *MISNCG or *MODEL *MISCG.

EXPLANATION:

The *PBS keyword defines the initial solvent saturation pressure. Any of the array reading options can be used.

If rss (entered in *PVTS) is approaching zero, that is, there is no solvent in the water phase, then the values for the solvent bubble point pressures are reset to the block pressures.

The acceptable range of values for solvent bubble point pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm ²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

Initial Oil Saturation (Conditional)

***SO**

PURPOSE:

*SO indicates input of initial oil saturation (fraction).

ARRAY:

*SO

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is required with the *USER_INPUT initialization option except when the *MODEL *GASWATER fluid model is used.

EXPLANATION:

The *SO keyword defines the initial oil saturation at each block. Any of the array reading options can be used.

Initial Water Saturation (Conditional)

***SW**

PURPOSE:

*SW indicates input of initial water saturation (fraction).

ARRAY:

*SW

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. Required with

*USER_INPUT initialization option except when the *MODEL *OILWATER fluid model is used.

EXPLANATION:

The *SW keyword defines the initial water saturation at each block. Any of the array reading options can be used.

Initial Polymer Concentration (Conditional)

***POLYCONC**

PURPOSE:

*POLYCONC indicates input of initial polymer concentration (kg/m³ | lb/bbl | g/cm³).

ARRAY:

*POLYCONC

DEFAULTS:

Conditional keyword. If the *POLYCONC keyword is missing then the initial polymer concentration in the reservoir is set to zero.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. Required when the polymer option is being used and nonzero initial polymer concentrations are required.

EXPLANATION:

The *POLYCONC keyword defines the initial polymer concentration at each block. Any of the array reading options can be used.

The acceptable range of values for polymer concentration is:

	SI Kg/m³	Field Lb/stb	Lab g/cm³	Mod. SI Kg/m³
min	0.0	0.0	0.0	0.0
max	29.0	10.0	0.029	29.0

Initial Seawater Volume Fraction (Conditional) *SEAWATFRC

PURPOSE:

*SEAWATFRC indicates input of initial seawater volume fraction (fraction).

ARRAY:

*SEAWATFRC

DEFAULTS:

Conditional keyword. If the *SEAWATFRC keyword is missing then the initial seawater fraction in the reservoir is set to zero.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. Used when the seawater injection option is being used and nonzero initial seawater volume fractions are required.

EXPLANATION:

The *SEAWATFRC keyword defines the initial seawater volume fraction at each block. Any of the array reading options can be used.

The acceptable range of values for seawater volume fraction is:

	SI Fraction	Field Fraction	Lab Fraction	Mod. SI Fraction
min	0.0	0.0	0.0	0.0
max	1.0	1.0	1.0	1.0

Initial Light Oil Volume Fraction (Conditional)

*API

PURPOSE:

*API indicates input of initial light oil volume fraction at surface conditions (fraction).

ARRAY:

*API

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is required if *MODEL *API-INT or *API-INTOW is used and *APIT is not used.

EXPLANATION:

The *API keyword defines the initial light oil volume fraction at surface conditions at each block. Any of the array reading options can be used.

Initial Light Oil API Volume Fraction vs. Depth (Conditional)

*APIT

PURPOSE:

*APIT indicates the input of Initial light oil volume fraction (Surface Conditions) in tabular format as a function of depth (fraction).

TABLE:

*APIT	<i>set_number</i>
<i>depth</i>	<i>API</i>
:	:

DEFINITIONS:

set_number

Set number for this set of light oil volume fractions vs. depth table. This is the initialization region number used with *PTYPE or *ITYPE to assign initialization regions to grid blocks. The *set_number* must correspond to the *set_number* specified with *PVTAPI or *NREGIONS and must increase monotonically.

depth

Depth (m | ft | cm).

API

Initial Light Oil Volume Fraction (fraction).

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group and is Required if *MODEL *API-INT or *API-INTOW is used and *API is not used.

This keyword must be in the Initial Conditions keyword group and must consist of two columns of numerical entries. The number of *APIT keywords must be equal to the number of PVT regions in the Component Properties section, if *NREGIONS is not defined, or *NREGIONS, if *NREGIONS is defined. Each initialization region can have at most one API vs. Depth table.

Light oil volume fraction is limited to a range between 0.0 and 1.0.

EXPLANATION:

The *APIT keyword defines initial light oil volume fraction as a function of depth for an initialization region. Values of light oil volume fraction are then assigned to each grid block by use of the *PTYPE or *ITYPE keyword.

Examples:

```
*APIT 1
** depth    Light oil volume fraction
  3000.0          0.8
  3200.0          0.6
```

Note that the reservoir depth and the depth for the woc and goc must fall within the depths specified in the table which is used in that initialization region.

Reference Depth and Reference Pressure (Conditional)

*REFDEPTH, *REFPRES

PURPOSE:

*REFDEPTH indicates input of reference depth.

*REFPRES indicates input of reference pressure.

FORMAT:

*REFDEPTH *depth*

*REFPRES *pressure*

DEFINITIONS:

depth

Depth of reference pressure (*REFPRES) (m | ft | cm). 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.

pressure

The reference pressure (kPa | psi | kPa | kg/cm²) at the reference depth (*REFDEPTH) is used to set reservoir block pressures.

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. Required with all *VERTICAL subkeywords.

EXPLANATION:

The reference pressure at the reference depth is used in the vertical equilibrium option, along with the capillary pressures, fluid densities, water-oil contact and the gas-oil contact depths, to calculate the block pressures and saturations.

If multiple initialization regions have been specified, then a corresponding number of reference depths and pressures must also be specified. If the number of reference depths and pressures is less than the number of initialization regions then the last entered values will be assigned to the remaining initialization regions.

Each initialization region can have at most one reference depth and pressure. Multiple reference depths and pressures can be entered by specifying a corresponding number of PVT regions with identical PVT data for each region or by using the *NREGIONS and *ITYPE keywords.

For multiple initialization regions *REFDEPTH and *REFPRES can be entered in either of two ways

1. Separate keywords for each region:

Example: *REFPRES "Pressure for Region 1"
 *REFPRES "Pressure for Region 2"

2. Multiple regions on one keyword:

Example: *REFPRES "Pressure for Region 1" "Pressure for Region 2"

The acceptable range of values for reference depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for reference pressure is:

	SI kPa	Field psi	Lab kPa	Mod. SI kg/cm²
min	0.0	0.0	0.0	0.0
max	1.0E+6	145,038.0	1.0E+6	1.0E+4

If using the *PBT keyword, the reference depth must be within the range of depth values entered in the bubble point pressure vs. depth table which used in that initialization region.

Depth to Water-Oil Contact (Conditional) *DWOC, *DGOC, *DWGC

PURPOSE:

- *DWOC indicates input of water-oil contact(s).
- *DGOC indicates input of gas-oil contact depth.
- *DWGC indicates input of water-gas contact depth.

FORMAT:

- *DWOC *woc_depth*
- *DGOC *goc_depth*
- *DWGC *wgc_depth*

DEFINITIONS:

woc_depth

Depth (m | ft | cm) to water-oil contact plane. If the value is negative, then *DWOC is interpreted as an altitude above some datum.

goc_depth

Depth (m | ft | cm) to gas-oil contact plane. If the value is negative, then *DGOC is interpreted as an altitude above some datum.

wgc_depth

Depth (m | ft | cm) to water-gas contact plane. If the value is negative, then *DWGC is interpreted as an altitude above some datum.

DEFAULTS:

Conditional keyword. No default value.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group. The use of the *DWGC keyword is not allowed in a *WATER_OIL_GAS initialization. Use equal *DWOC and *DGOC values instead.

```
*DWOC is required with:  
  *VERTICAL *BLOCK_CENTER *WATER_OIL  
  *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS  
  *VERTICAL *DEPTH_AVE    *WATER_OIL  
  *VERTICAL *DEPTH_AVE    *WATER_OIL_GAS  
*DGOC is required with:  
  *VERTICAL *BLOCK_CENTER *WATER_OIL_GAS  
  *VERTICAL *DEPTH_AVE    *WATER_OIL_GAS  
*DWGC is required with:  
  *VERTICAL *BLOCK_CENTER *WATER_GAS  
  *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.

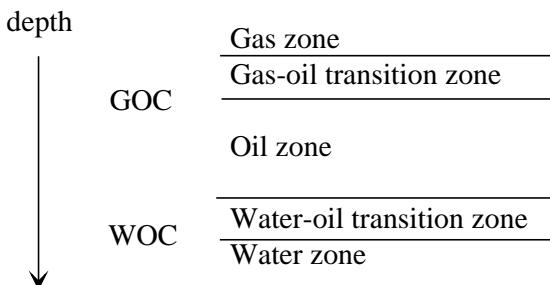
For the *BLOCK_CENTER options, the water-oil contact depth is defined as the smallest depth at which the water saturation assumes its maximum value (unity or nearly unity). This definition is consistent with the one from Slider, H.C., "Practical Petroleum Reservoir Engineering Methods", Chapter 5, 1976, PennWell Books, Tulsa, pp. 266.

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 nonzero value for the water-oil capillary pressure at the water-oil contact in the *DEPTH_AVE options by using the *WOC_PC keyword.

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. This definition is consistent with the one from Slider, H.C., "Practical Petroleum Reservoir Engineering Methods", Chapter 5, 1976, PennWell Books, Tulsa, pp. 266.

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 nonzero value for the oil-gas capillary pressure at the gas-oil contact in the *DEPTH_AVE options by using the *GOC_PC keyword.

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 (see the explanations with the *WOC_PC and *GOC_PC keywords.).



If the number of water-oil, gas-oil or gas-water contacts is less than the number of initialization regions then the last entered value will be assigned to the remaining initialization regions.

Each initialization region can have at most one water-oil, gas-oil and water-gas contact only. Multiple contacts can be entered by specifying a corresponding number of PVT regions with identical PVT data for each region or by using the *NREGIONS and *ITYPE keywords.

For multiple initialization regions *DWOC *DGOC and *DWGC can be entered in either of two ways

1. Separate keywords for each region:

Example:

```
*DWOC "contact for Region 1"  
*DWOC "contact for Region 2"
```

2. Multiple regions on one keyword:

Example:

```
*DWOC "contact for Region 1" "contact for Region 2"
```

The value for depths entered must fall within the range of depths entered with *PBT for that initialization region (if bubble point pressure is a function of depth).

The acceptable range of values for depth of contacts is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Datum Depth Specification (Optional)

***DATUMDEPTH**

PURPOSE:

*DATUMDEPTH specifies the datum depth for pressure printout corrected to datum for each initialization region. A *DATUMDEPTH keyword should be entered for each initialization region. The first *DATUMDEPTH applies to initialization region 1, the second to initialization region 2 ...and so on.

FORMAT:

```
*DATUMDEPTH  depth  (*INITIAL |  
                      *REFDENSITY (*GRIDBLOCK / density))
```

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/m³ | lb/ft³). 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.

If one *DATUMDEPTH keyword is found in a model with multiple initialization regions, the single datum depth specification will be applied to all initialization regions. If fewer *DATUMDEPTH keywords than PVT regions are found (but more than a single *DATUMDEPTH is specified) all remaining initialization regions will default to the last specified *DATUMDEPTH and a warning will be issued.

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.

$$\text{datum_pressure(ib)} = \text{pressure(ib)} - \text{pressure_0(ib)} + \text{datum_pressure_0}$$

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.

$$\text{datum_pressure(ib)} = \text{pressure(ib)} + \text{density} * g * (\text{datumdepth} - \text{depth(ib)})$$

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.

$$\text{datum_pressure(ib)} = \text{pressure(ib)} + \text{density(ib)} * g * (\text{datumdepth} - \text{depth(ib)})$$

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.

When *DATUMDEPTH is specified for a multiple initialization region model, each initialization region can use a different *DATUMDEPTH calculation method and/or use different method parameters.

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.

The above 5 examples could also define the datum depths for a single run with 5 initialization regions.

Specification of Capillary Pressure Values at Contacts (Optional)

*GOC_PC, *WOC_PC

PURPOSE:

*GOC_PC AND *WOC_PC introduce the value of the oil-gas capillary pressure at the gas-oil contact and the value of the water-oil capillary pressure at the water-oil contact, respectively. These values are used in the *VERTICAL *DEPTH_AVE initialization options. Values of *GOC_PC AND *WOC_PC can be entered for each initialization region.

FORMAT:

*GOC_PC *pcggoc*
*WOC_PC *pcwwoc*

DEFINITIONS:

pcggoc

A real number specifying the value of the oil-gas capillary pressure at the gas-oil contact (kPa | psi | kPa | kg/cm²) for an initialization region.

pcwwoc

A real number specifying the value of the water-oil capillary pressure at the water-oil contact (kPa | psi | kPa | kg/cm²) for an initialization region.

DEFAULTS:

Optional keywords. If this keyword is 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 and *pcwwoc* = 0.

OILWET Option:

If the OILWET option is used, pcwwoc is defaulted to the value of the largest capillary pressure in all of the *SWT tables in the initialization region. This is normally the value at the minimum wetting phase (oil) saturation in each table. The user can always explicitly input a desired value to override this default. Please see the Tutorial section for details.

CONDITIONS:

These keywords, if present, must be in the Initial Conditions keyword group. No error message is printed if a *VERTICAL *DEPTH_AVE option is chosen but *GOC_PC or *WOC_PC does not appear in the data; the values of pcggoc or pcwwoc are defaulted 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 IMEX, 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. Similarly, *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.

Examples:

```
*GOC_PC -0.5  
*WOC_PC 10.0
```

The acceptable range of values for *GOC_PC is between the minimum and maximum values of gas- oil capillary pressure from gas-liquid relative- permeability tables in the initialization region it applies to.

The acceptable range of values for *WOC_PC is between the minimum and maximum values of water- oil capillary pressure from water-oil relative- permeability tables in the initialization region it applies to.

If the *WOC_PC for a PVT region is less than the minimum value of water oil capillary pressures in the initialization region, a warning message will be issued. The run will continue to initialize using the following logic:

Below the WOC: Water saturation is set to 1.0

From the WOC to the height above the water oil contact where the head difference between reservoir condition water and oil equals the minimum water oil capillary pressure in the relative permeability table which applies to a particular grid block minus the *WOC_PC, the water saturation is set to 1.0 – Soirw (oil saturation = Soirw).

Moving up from the height above the water oil contact where the head difference between oil and water equals the minimum water oil capillary pressure in the applicable relative permeability table minus the *WOC_PC, the water and oil saturation in the transition zone is determined from the water oil capillary pressure curve, in the standard fashion.

A similar logic is applied if the *GOC_PC is less than the minimum value of gas oil capillary pressure in an initialization region.

If one *WOC_PC or *GOC_PC keyword is found then this value will be assigned to all initialization regions. If there are fewer values entered than PVT regions (but more than one) then an error will be raised.

Each initialization region can have at most one *WOC_PC or *GOC_PC. For multiple initialization regions *WOC_PC or *GOC_PC can be entered in either of two ways

1. Separate keywords for each region:

Example:
*WOC_PC "pcwwoc for Region 1"
*WOC_PC "pcwwoc for Region 2"

2. Multiple regions on one keyword:

Example:

```
*WOC_PC "pcwwoc for Region 1" "pcwwoc for Region 2"
```

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 is exactly equivalent to the previous *GASCAP, which is still accepted by IMEX.

FORMAT:

*GASZONE (*NOOIL | *OIL)

DEFINITIONS:

***OIL**

Indicates that the oil/gas residual saturation S_{org} is to be set initially 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.

***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.

DEFAULTS:

Optional keywords. If *GASZONE is not present then *NOOIL is assumed and oil saturations of zero are assigned initially in the gas zone.

CONDITIONS:

This keyword, if present, must be in the Initial Conditions keyword group. It applies for all of the *VERTICAL initialization options. If *GASZONE is entered when the *USER_INPUT initialization option is specified, then *GASZONE is ignored and no message is issued.

EXPLANATION:

Normally in black-oil simulation, reservoirs are initialized such that the primary gas-cap does not have any residual oil. However if a nonzero oil saturation is desired in the gas-cap at initial time then the *GASZONE *OIL option should be used. The oil saturation in the gas cap is set to the residual oil saturation derived from the liquid-gas relative-permeability table. This is normally calculated as the residual liquid saturation in the gas-liquid table minus the connate water saturation. When the *NOSWC option is used, the residual oil saturation is equal to the residual liquid saturation.

Example:

*GASZONE *OIL

Specification of Water Saturation Above GOC or Below WOC (Optional)

*GOC_SW, *WOC_SW

PURPOSE:

*GOC_SW and *WOC_SW allow the user to define the water saturation above the gas oil contact and below the water-oil contact for each initialization region.

FORMAT:

*GOC_SW	<i>sw_gascap</i>
*WOC_SW	<i>sw_waterzone</i>

DEFINITIONS:

sw_gascap

A real number specifying the value of the water saturation above the gas oil contact for an initialization region (Fraction).

sw_waterzone

A real number specifying the value of the water saturation below the water-oil contact for an initialization region (Fraction).

DEFAULTS:

Optional keywords. If this keyword is present and any of the *BLOCK_CENTER initializations is being used, the values will be accepted. WOC_SW is also accepted for DEPTH_AVE initialization. If *USER_UNPUT initialization is being used a warning will be issued and the input values will be ignored.

CONDITIONS:

These keywords, if present, must be in the Initial Conditions keyword group. One GOC_SW (WOC_SW) card must appear for each initialization region. The first appearance assigns a value to the first initialization region. The next appearance of GOC_SW (WOC_SW) assigns a value to the next initialization region (and so on). One card must appear for each initialization region defined.

*WOC_SW can be used with *BLOCK_CENTER and *DEPTH_AVE initialization options.

*GOC_SW can only be used with the *BLOCK_CENTER initialization options.

*GOC_SW is not to be used with the *GASWATER or *GASWATER_WITH_CONDENSATE options.

Care must be exercised when using this option as it is possible to create mobile oil in the water zone or mobile water in the gas cap by poor choices of input. In general, *GOC_SW should be less than the initialization region's critical water saturation and *WOC_SW should be greater than one minus the initialization region's residual oil saturation (Sorw).

EXPLANATION:

By default IMEX defines the initial water saturation of each block in a Gas Cap to the block's connate water saturation. This value is either defined by the relative permeability table assigned to each block in the Gas Cap or by the value explicitly defined on the *SWCON card. Also, by default, IMEX assigns the initial water saturation of each block in the water zone to 1.0.

The *GOC_SW and WOC_SW keywords allow the user to easily override these defaults for each initialization region. *GOC_SW and *WOC_SW will only alter water saturation in the gas cap and water zone respectively. They allow the user to set initial water saturation to a single value above (or below) the oil zone.

If defined, one *GOC_SW (or *WOC_SW) must appear for each initialization region. The order of input defines which initialization region water saturation is assigned to. The first card assigns *GOC_SW or *WOC_SW to the first initialization region, the last card assigns *GOC_SW or *WOC_SW to the last initialization region.

*GOC_SW and *WOC_SW override all values calculated in the gas cap or water zone, even those defined on the *SWINIT card.

For multiple initialization regions *GOC_SW and *WOC_SW can be entered in either of two ways

1. Separate keywords for each region:

Example:
*GOC_SW "Saturation for Region 1"
*GOC_SW "Saturation for Region 2"

2. Multiple regions on one keyword:

Example:
*GOC_SW "Saturation for Region 1" "Saturation for Region 2"

Examples:

```
*GOC_SW 0.20  
*WOC_SW 0.25
```

Specification of Minimum Initial Oil Saturation for API Tracking (Optional)

*APIMINSO

PURPOSE:

*APIMINSO allows the user to alter the default minimum oil saturation (0.0001) in each block for the API tracking option when using DEPTH_AVE initialization.

FORMAT:

*APIMINSO *somin*

DEFINITIONS:

somin

A real number specifying the minimum oil saturation a block can be initialized to.

DEFAULTS:

Optional keyword. If this keyword is not present and API tracking is active with DEPTH_AVE initialization, the default somin of 0.0001 is applied. If this keyword is present, somin will override the default value.

CONDITIONS:

This keyword, if present, must be in the Initial Conditions keyword group. Only one *APIMINSO is used for all initialization regions. *APIMINSO is used with DEPTH_AVE initialization when running an API tracking model.

Somin must be less than Sorw and Slcon (Swcon + Soirg).

EXPLANATION:

By default IMEX, when using the API tracking option and DEPTH_AVE initialization, defines a minimum initial oil saturation to avoid formulation issues when So = 0.0. The default value of 0.0001 can be modified (usually increased) in order to improve model performance.

Example:

*APIMINSO 0.0003

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 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 0.0 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. The *GOC_SW overrides the *SWINIT and can be used to modify gas zone Sw further.

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, 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 0.0 at all saturations). This eliminates the two restrictions above.

In the water zone, *SWINIT values are ignored. The value of water saturation in the water zone is normally 1.0. The *WOC_SW keyword overrides the *SWINIT keyword and can be used to modify water zone water saturation.

In summary

Gas Cap:

$0 < \text{Swinit} < \text{Sw(critical)}$
 $\text{Sw reset to } \text{Sw(critical)} \text{ if } > \text{Sw(critical)}$

Oil Zone:

$\text{Sw reset to } \text{Sw(connate)} \text{ if } \text{Sw} < \text{Sw(connate)}$
 $\text{Pcow scaled only if both original Pcow and scaled Pcow} > 0.1 \text{ kPa}$
(*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:

Swinit entry is ignored

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 sat.

*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,**

***SONEQ**

PURPOSE:

*SWNEQ and/or *SONEQ indicates input of non equilibrium initial saturation overrides to be used with gravity-capillary equilibrium initialization.

ARRAY:

*SWNEQ
*SONEQ

DEFAULTS:

Conditional keyword. No default values.

CONDITIONS:

This keyword must be in the Initial Conditions keyword group.

*SWNEQ and *SONEQ 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. In models where both oil and gas phases could initially exist, both *SWNEQ and *SONEQ need to be defined in each block the user overrides.

If oil saturation is never present (model *GASWATER) or not input initially (model *GASWATER_WITH_CONDENSATE), then *SWNEQ alone is entered. *SONEQ is internally set to zero.

If the gas phase is not modeled, in a model like model *OILWATER, then *SONEQ alone is entered. *SWNEQ is internally set to $(1.0 - *SONEQ)$.

EXPLANATION:

The *SWNEQ/*SONEQ keywords allow the user to override the equilibrium 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 SONEQ/SWNEQ *IJK range.

The “*ALL” array input option maybe used with *SONEQ and *SWNEQ. Values of *SONEQ and *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

*SONEQ *IJK
  4:28 14:18 4:6    0.2
  5:27 15:17 4:6    0.1
  6:26 16:16 5:5    0.01
```

Specification of Initialization Behavior when Contacts Lie Outside PBT Table Depths

***CONTACT_PBT_BEHAVIOR**

PURPOSE:

*CONTACT_PBT_BEHAVIOR controls how the simulator handles the situation when an initialization region's WOC or GOC is outside of the depth range specified by the upper and lower depths of an initialization region's PBT table.

FORMAT:

*CONTACT_PBT_BEHAVIOR (*ERROR | *LEGACY | *WARNING)

DEFINITIONS:

***LEGACY**

Warn the user that WOC or GOC lies outside of the depth range defined by the PBT table and then reset WOC or GOC to lie within that range. This has the effect of altering the user input value WOC or GOC.

***ERROR**

Current default. Issue an error message telling the user that WOC or GOC lies outside of the depth range defined by the PBT table and stop the run after initialization is complete. This allows the user to correct the contact or PBT table.

***WARNING**

Warn the user that WOC or GOC lies outside of the depth range defined by the PBT table; this is purely a warning and does not reset user specified contacts.

DEFAULTS:

Optional keyword. If *CONTACT_PBT_BEHAVIOR is not present then *ERROR is assumed. The run will stop.

CONDITIONS:

This keyword, if present, must be in the Initial Conditions keyword group. It applies to all initialization regions. It is ignored when the *USER_INPUT initialization option is used.

EXPLANATION:

Normally in black-oil simulation, reservoirs are initialized such that the contacts lie within the range of depths specified by the PBT table depths. When the contacts DWOC, DGOC (or DWGC) lie outside this range; three possible behaviors can be specified.

The *LEGACY option duplicates what IMEX did by default (until version 2014.10) This option resets the contacts to lie within the PBT table range and issues a warning to the user about the action taken.

The *ERROR option makes this an error and forces the user to extend the PBT table or possibly alter the contact depth.

The *WARNING option issues a warning when contacts lie outside of the depth range of the PBT table but does not alter the contacts to lie within the contact depths.

When using the *LEGACY option to alter contacts outside of the range defined by the PBT table (for each initialization region), the following is done:

For contacts above the first (shallowest) depth in the table, the contact is reset to the depth at the first tabulated depth-pressure pair. For contacts below the last (deepest) depth in the table, the contact is reset to the depth at the last tabulated depth-pressure pair.

Numerical Methods Control

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.

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. If it is not present in the input-data-file, the default is that the number of timesteps for the simulation is not explicitly limited.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group. If the command-line options "-onestep" or "-maxsteps *n*" are used they will override this keyword. If both "-onestep" and "-maxsteps *n*" are specified then "-onestep" will be used.

EXPLANATION:

Simulation will be terminated when the number of timesteps is equal to the number specified by *MAXSTEPS.

Examples:

```
** Limit the number of timesteps to 100
*MAXSTEPS 100
```

Maximum CPU Seconds (Optional)

*MAXCPU

PURPOSE:

*MAXCPU is used to specify the maximum CPU seconds for the simulation run.

FORMAT:

*MAXCPU *maxcpu*

DEFINITIONS:

maxcpu

A number to specify the maximum CPU time in seconds allowed for this simulation run.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is that CPU time for the simulation is not explicitly limited.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

Simulation will be terminated when the CPU time exceeds the number specified by *MAXCPU.

This allows the program to terminate gracefully, having written output and restart records before running out of CPU time.

This option requires the user to supply a subroutine called GETCPU which makes a system call and returns with the current CPU time from the system. Please see the installation instructions and notes section of the User manual.

Examples:

```
** Limit the CPU time to one hour.  
*MAXCPU 3600.0
```

Maximum Timestep Size (Optional)

*DTMAX

PURPOSE:

*DTMAX identifies the maximum time-step size.

FORMAT:

*DTMAX *max_time_size*

DEFINITIONS:

max_time_size

A real number to specify the maximum time-step size allowed (day | day | min).

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the maximum timestep size is 365 days.

CONDITIONS:

This keyword may be located in the Numerical Methods Control keyword group, and may also be part of recurrent data. When restarting a run, the *DTMAX keyword must be in recurrent data, after the restart time, to change from the value in the restarted run.

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*.

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. The maximum time-step size has no real preset minimum or maximum. But to avoid computer round-off problems a min. of 1.0E-10 days and a max. of 1.0E+20 days has been imposed.

Examples:

```
** Limit the maximum time-step size to half a day
*DTMAX 0.5
```

Minimum Timestep Size (Optional)

*DTMIN

PURPOSE:

*DTMIN identifies the minimum time-step size.

FORMAT:

*DTMIN *min_time_size*

DEFINITIONS:

min_time_size

A real number to specify the minimum time-step size allowed (day | day | min).

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the minimum time-step size is 0.001 days.

CONDITIONS:

This keyword may be located in the Numerical Methods Control keyword group, and may also be part of recurrent data. The minimum timestep size should be set to a reasonable value. Do not simply reduce the minimum time-step to unrealistic sizes to try to converge.

*DTMIN must be smaller than *DTWELL.

EXPLANATION:

If the automatic time-step selector cuts the timestep size to smaller than *min_time_size* specified by *DTMIN, the time-step size is set to *min_time_size*. If the timestep does not converge, the simulation is stopped. The maximum time-step size has no real preset minimum or maximum. But to avoid computer round-off problems a min. of 1.0E-10 days and a max. of 1.0E+20 days has been imposed.

Examples:

```
** Limit the minimum time-step size to 6 hours
*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.

*MAXCHANGE identifies whether a timestep is cut if the maximum change in pressure or saturation is greater than that specified by *MAXCHANGE.

FORMAT:

*NORM / *MAXCHANGE vars *value*

where

vars = *PRESS | *SATUR | *PBUB | *PDW

DEFINITIONS:

***PRESS**

This subkeyword identifies pressure (kPa | psi | kPa | kg/cm²).

***SATUR**

This subkeyword identifies saturations (fraction, dimensionless).

***PBUB**

This subkeyword identifies bubble point pressure (kPa | psi | kPa | kg/cm²).

***PDW**

This subkeyword identifies dew point pressure (kPa | psi | kPa | kg/cm²) (*GASWATER_WITH_CONDENSATE and *VOLATILE_OIL model only).

value

A real number to specify the controlled value.

DEFAULTS:

Optional keyword. If *NORM is not present in the input-data-file the defaults are:

Pressure - 3000.0 kPa, 435.0 psia or 30.0 kg/cm² for pressure or bubble point pressure.

Saturation - 0.1 for oil, water, and gas saturation.

Bubble Point Pressure - defaults to the same value as pressure

Dew Point Pressure - 10,000 kpa, 1450 psi or 100 kg/cm²

For all Models, except *MISCG and *MISNCG, the defaults for *MAXCHANGE are two times the *NORM values (i.e. if the *MAXCHANGE keyword is missing).

For the *MISCG and the *MISNCG options defaults for *MAXCHANGE are:

Pressure -	1.0E+30 kPa, 1.45E+29 psia or 1.0E+28 kg/cm ² for pressure or bubble point pressure.
Saturation -	1.0 for oil, water, gas or solvent saturation.
Bubble Point Pressure -	defaults to the same value as Pressure

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

The *NORM keyword specifies the typical changes in the basic variables during a timestep. These are used for automatic time-step selection.

The *MAXCHANGE keyword specifies the typical changes in the basic variables during a timestep. If the change is greater than that specified by *MAXCHANGE then the timestep is cut in order to maintain the changes specified by *NORM.

The values for pressure and saturation changes MUST always be greater for *MAXCHANGE than *NORM.

For field-scale water injection simulation, the pressure variation may be set to 7000 kPa (1000 psi or 70 kg/cm²) and the saturation variation may be set to 0.20.

For single well coning studies a pressure variation of 1000 kPa (145 psi or 10 kg/cm²) and a saturation variation between 0.05 and 0.10 may be used.

Examples:

```
*NORM *PRESS 145.0  
*NORM *SATUR 0.08
```

Relaxation Options (Optional)

***RELAX**

PURPOSE:

*RELAX selectively enables various relaxation options.

FORMAT:

*RELAX (*ACC (*ON | *OFF)) (*GAS (*ON | *OFF))

DEFINITIONS:

***ACC**

This subkeyword enables over-relaxation in the accumulation term. This may be helpful in gas-appearance, gas-disappearance problems.

***GAS**

This subkeyword enables the under-relaxation for the gas saturation disappearance.

***OFF**

Disables the corresponding relaxation option.

DEFAULTS:

Optional keyword. If *RELAX is not present in the data set then the default is *OFF for *GAS relaxation. *ACC relaxation is set by default to on, except when the pseudo-miscible options are used. When the *MISCG or *MISNCG models are used, *ACC relaxation is turned off by default.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

The *RELAX *GAS enables the gas-saturation under-relaxation option. This option can lead to faster convergence for simulations where the overall pressure in the reservoir is declining and gas is being liberated due to the pressure falling below the bubble point.

This option prevents the gas from disappearing in a grid block for three Newtonian iterations. If IMEX predicts that gas in a grid block will disappear for more than three iterations, only then will the gas be allowed to disappear.

Conversely, this option can slow the convergence for those simulations where there is free gas present, and the reservoir pressure is rising. This is because for such situations when IMEX predicts gas disappearance in a grid block the gas indeed should be allowed to disappear.

The *RELAX *ACC option performs over-relaxation of the accumulation term. This normally improves model performance.

Examples:

```
*RELAX *GAS *OFF    ** Do not use gas saturation  
                      ** under-relaxation  
*RELAX *GAS          ** Use gas saturation  
                      ** under-relaxation
```

Adaptive Implicit Switching (Optional)

*AIM

PURPOSE:

*AIM controls the adaptive implicit switching option.

FORMAT:

```
*AIM    *STAB      (*AND-THRESH (frac1 frac2) |
          *BACK freq | *ALL-BLOCKS)) |
*THRESH   (frac1 frac2) |
*OFF
```

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.

*BACK

This subkeyword specifies backward switching. It enables switching from implicit to explicit based on the stability switching criterion.

*ALL-BLOCKS

By default, the stability based switching algorithm only checks explicit neighbours of implicit blocks for stability. This feature checks all blocks for explicit to implicit switching. This feature is more robust than the *STAB option on its own but can add 10-15% to CPU times when many explicit blocks are present.

*AND-THRESH

A hybrid technique which checks explicit neighbours of implicit blocks for stability switching and checks all blocks for threshold switching using the *NORM fractions $frac_1$ and $frac_2$. This technique is nearly as robust as the *ALL-BLOCKS method and requires no extra CPU time.

freq

An integer number to specify the frequency of checking backward switching. (timesteps)

*THRESH

This subkeyword identifies threshold values for adaptive implicit switching.

frac₁

A real number specifying the fraction of values set by *NORM for checking the bubble point pressure (the primary variable).

*frac*₂

A real number specifying the fraction of values set by *NORM for checking the oil, water and gas saturations and to the polymer concentration when applicable (primary variables). The value *frac*₂ is required with *THRESH.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is:

*AIM *STAB *AND-THRESH

Default values *frac*₁ = 0.25 and *frac*₂ = 0.25.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

This keyword controls the switching of a grid block between an IMPES and a fully implicit formulation.

Switching is based on either the numerical stability of a local amplification matrix or on threshold changes in primary variables or on both. It is recommended that the stability switching criterion *STAB or the hybrid criterion *STAB *AND-THRESH be used.

The variable *frac*₁ applies to bubble point pressures, and *frac*₂ applies to saturations. The variable *frac*₂ also applies to the polymer reference concentration (*PREFCONC) when *POLYMER option is used.

One of two cases may exist when using *frac*₁ and *frac*₂: either both are entered or none are entered and the defaults apply.

Examples:

```
** Threshold values for adaptive implicit switching
*AIM *THRESH 0.25 0.25
** stability adaptive implicit switching
*AIM *STAB
** Hybrid stability-threshold adaptive implicit
** switching
*AIM *STAB *AND-THRESH
```

Convergence of Newton's Method (Optional)

*CONVERGE

PURPOSE:

*CONVERGE is used to specify the convergence criteria for Newton's Method when solving the non-linear conservation equations in the simulator.

FORMAT:

*CONVERGE type *value*

where

type = *PRESS | *SATUR | *MAXRES (eqtn) |

*TOTRES | *WELLRES

eqtn = (*OIL) (*WATER) (*GAS) (*SOLVENT) (*POLYMER)
(*LIGHTOIL) (*SEAWATER) (*OTHER)

DEFINITIONS:

*PRESS

Indicates changes in convergence tolerance for pressure (kPa | psi | kPa | kg/cm²).

*SATUR

Indicates changes in convergence tolerance for saturation (fraction).

*MAXRES

Indicates the maximum scaled residual allowed for any single equation at convergence.

<u>Eqtn</u>	<u>Applies value to All Equations</u>
*OIL	Oil Equation
*WATER	Water Equation
*GAS	Gas Equation
*SOLVENT	Solvent Equation
*POLYMER	Polymer Equation
*LIGHTOIL	API Tracking Light Oil Component
*SEAWATER	Seawater Equation
*OTHER	Solvent, Polymer, Seawater or Light Oil Component

*TOTRES

Indicates the maximum average scaled residual for all equations for a particular phase or component. (i.e. the average of all oil equations).

*WELLRES

Indicates the maximum scaled residual allowed for any single well equation at convergence.

value

A real number indicating the tolerance or residual value.

DEFAULTS:

Optional keyword. The default is to use a residual convergence criteria:

Maximum average residual 0.001

Maximum well residual 0.001

Maximum Residual 0.1 (Single Porosity)

Maximum Residual 0.5 (Dual Porosity)

Polymer Maximum Residual is reduced to 0.005 times the default value if not explicitly specified using the *MAXRES *POLYMER keyword

Seawater Maximum Residual is reduced to 0.005 times the default value if not explicitly specified using the *MAXRES *SEAWATER keyword

Oil Maximum Residual (when using the *GASWATER_WITH_CONDENSATE model only) is reduced to 0.1 times the default value if not explicitly specified using the *MAXRES *OIL keyword

If a changes convergence criteria is specified, the default for the other changes tolerance is:

Pressure 10 kPa, 2.0 psia or 0.10 kg/cm²

Saturation 0.005

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group. You cannot specify both changes convergence criteria and residual convergence criteria.

EXPLANATION:

*CONVERGE specifies the convergence criteria and tolerances for conservation equations during Newton iterations.

The default is to use a residual criteria to specify convergence. Equations are considered converged when the maximum residual is less than a specified value, and the L1 norm (average absolute value) of the residuals for each phase is less than a specified value.

The material conservation equation is:

$$(\text{Flow Terms}) + (\text{Source Terms}) - (\text{Accumulation Term}) = R$$

Using Newton's Method, the equation is not solved exactly to zero, but to some residual value R, less than the specified residual for convergence. R is first scaled to give a "saturation equivalent" error. Well equation residuals are expressed as a fraction of the specified rate.

If the format *CONVERGE *MAXRES *value* is used, or the default values above are used, then:

1. The Polymer or Seawater Maximum residual is modified to be $0.005 \times \text{value}$.
2. When using the GASWATER_WITH_CONDENSATE model, the Oil residual is modified to be $0.10 \times \text{value}$.

The format *CONVERGE *MAXRES eqtn *value* has been introduced to allow the user complete control over Maximum Residual convergence tolerances. The ‘eqtn’ keyword can be one of *OIL, *WATER, *GAS, *SOLVENT, *POLYMER, *SEAWATER, *LIGHTOIL, or *OTHER. When an equation’s maximum residual tolerance is specified explicitly in this fashion it is used directly. This format may be used to override the default tolerance or a tolerance specified on a preceding *MAXRES *value* keyword. See examples below.

Alternately, the changes in primary variables can be used as a convergence criteria. THIS WAS THE METHOD USED IN IMEX 4.0 AND PRIOR VERSIONS.

Equations are considered converged when changes during Newtonian iterations are less than tol.

Examples:

```
*CONVERGE *PRESS 2.50
*CONVERGE *SATUR 0.005
*CONVERGE *MAXRES 0.01
*CONVERGE *MAXRES *GAS 0.01          **Set tolerance to 0.01 globally
                                         **Override Default for gas
                                         **component, use defaults for
                                         **other components
*CONVERGE *MAXRES 0.05
    *MAXRES *OIL 0.002
    *MAXRES *GAS 0.05          **Set tolerances to 0.05 globally
                                **Set Oil Component tolerance
                                **to 0.002
                                **Set Gas Component tolerance
                                **to 0.05
*CONVERGE *MAXRES 0.05
*CONVERGE *MAXRES *OIL 0.002          **Same action as preceding
*CONVERGE *MAXRES *GAS 0.05          **example
```

For a comparison of residual criteria and changes criteria, see CMG report 90.02.R.

Maximum Newtonian Cycles (Optional)

*NEWTONCYC

PURPOSE:

*NEWTONCYC is used to specify the maximum number of Newtonian cycles.

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 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, the simulator will reduce the timestep size and repeat the timestep.

Examples:

*NEWTONCYC 8

Minimum Number of Newtonian Cycles (Optional) *MINCYC

PURPOSE:

*MINCYC is used to specify the minimum number of Newtonian cycles when using the *DRSDT, *DPBDT, *DRVDT, *DPDWDT options.

FORMAT:

*MINCYC *minn*

DEFINITIONS:

minn

An integer indicating the minimum value.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default maximum number of Newtonian cycles is 1.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group and is only active when the *DRSDT or *DPBDT options are used (*DRVDT or *DPDWDT with the VOLATILE_OIL or GASWATER_WITH_CONDENSATE models).

EXPLANATION:

This keyword specifies the minimum number of Newton iterations allowed in a timestep for solution of the basic reservoir equations.

Examples:

```
*MINCYC 2
```

Formulation Modification for Blocks with no Hydrocarbon

*WATER_FIX

PURPOSE:

*WATER_FIX is used to override the default behavior in the simulator of handling bubble point pressure changes in water filled blocks.

FORMAT:

*WATER_FIX *num*

num

An integer indicating WATER_FIX type.

0	The default, standard formulation
1	Not used
2	Improve handling of water filled blocks with very low oil saturation as discussed below in Explanation
3	Alter criteria employed for <i>num</i> =2, recommended for API tracking models, takes into account both the water saturation of the last iterate update (like <i>num</i> =2) as well as the latest value of water saturation
4	As option 3, but also attempts to prevent water saturation greater than 1.0 from occurring

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default handling of bubble point pressure changes in water filled blocks is used.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

Normally the IMEX formulation and variable switching algorithm is robust enough to handle even the most difficult problems.

However under certain circumstances, when oil attempts to enter a block which is completely empty of any hydrocarbon, numerical issues may occur. These can be identified as periods of a simulation where timestep sizes fails to build up significantly and where timestep repeats are common.

Using *WPRN *ITER *MATRIX to turn on more timestep convergence information, these periods are fairly obvious as times when oil and gas residuals fail to reduce in blocks at the border/interface of hydrocarbon and completely water filled blocks. The maximum residuals of oil and gas fail to converge at these border blocks and appear to be stuck at values above the convergence tolerances. Occasionally the message “Newton Method is Stuck” will appear.

An optional algorithm to improve our formulation under these circumstances has been developed. The algorithm may be started by specifying *WATER_FIX num in the numerical data section or by adding the –water_fix *num* command-line option.

Examples:

```
*WATER_FIX 2
```

Force Maximum Change Checks (Optional)

*FORCE_MAXCHK

PURPOSE:

*FORCE_MAXCHK is used to override the default maximum change timestep repeating strategy.

FORMAT:

*FORCE_MAXCHK (*ON | *OFF)

DEFINITIONS:

*ON

Override maximum change convergence testing behavior.

*OFF

Do not override maximum change convergence testing behavior.

DEFAULTS:

Optional keyword. If it is not present in a data set the default maximum change convergence testing behavior will be used. By default it is *OFF for all models except API tracking models. By default it is *ON for API tracking models. If found in the data set without a sub-keyword *ON is assumed.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group

EXPLANATION:

When *FORCE_MAXCHK is off, maximum changes are used as a test for repeating timesteps only after the 3rd Newton iteration. If convergence is achieved within the first three Newton iterations and the maximum changes of primary variables exceed the specified maximum change allowed during a timestep, the timestep is accepted nonetheless.

This can occasionally lead to the acceptance of a bad solution, which can cause problems in subsequent timesteps.

When *FORCE_MAXCHK is on, convergence will be allowed in the first three Newton iterations only if the maximum change test is not violated. However the test itself is not applied. If it is violated, the simulator will continue iterating until either the maximum change test is not violated within the first three Newton iterations or the 4th Newton iteration is reached and the maximum change test is applied.

Examples:

```
*FORCE_MAXCHK *ON
```

Number of Cuts (Optional)

***NCUTS**

PURPOSE:

*NCUTS controls the number of time-step size cuts allowed in a single timestep.

FORMAT:

*NCUTS *value*

DEFINITONS:

value

A real number to specify the maximum number of cuts allowed.

DEFAULTS:

Optional keyword. If it does not appear in the data set, then the default is 4 cuts.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

Convergence Tolerance for Linear Solver (Optional) *RELtol

PURPOSE:

*RELtol is used to specify the convergence tolerance for the linear equation solver (AIMSOL), when using *CONVERGE *PRESS or *CONVERGE *SATUR to specify Newton Iteration convergence.

FORMAT:

*RELtol *tol*

DEFINITIONS:

tol

A real number indicating the tolerance value.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is: 0.1

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

Convergence of the linear equation solver is when, for all primary variables, the change in the update value for the primary variable during a linear solver iteration is less than the Newton Iteration convergence value specified by *CONVERGE times the value specified by *RELtol.

Examples:

```
*CONVERGE *PRESS 5.0** kPa  
*CONVERGE *SATUR 0.1  
*RELtol 0.05
```

The linear solver stop its iterations when all the changes in pressure primary variable updates are less than 0.25 kPa (5.0×0.05) and all the changes in saturation primary variable updates are less than 0.005 (0.1×0.05).

The linear solver will also stop iterating if the criteria specified by *PRECC is satisfied.

Convergence Tolerance for Linear Solver (Optional)

*PRECC

PURPOSE:

*PRECC is used to specify the convergence tolerance for the linear equation solver (AIMSOL).

FORMAT:

*PRECC *precc*

DEFINITIONS:

precc

A real number indicating the tolerance value.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is: 1.0E-04

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

Convergence of the linear equation solver is when the r.m.s. value of the residuals of the equations has been reduced to *precc* times its original value.

If the set of linear equations is

$$\mathbf{Av} = \mathbf{b}$$

the r.m.s. residual is

$$\mathbf{r} = \| \mathbf{b} - \mathbf{Av} \|$$

Convergence is when, for the *i*th iteration,

$$\mathbf{r}(i) / \mathbf{r}(0) < \textit{precc}$$

Orthogonalization (Optional)

*NORTH

PURPOSE:

*NORTH controls the maximum number of orthogonalizations to be performed before resetting for the iterative solution method.

FORMAT:

*NORTH *num*

DEFINITIONS:

num

An integer defining the maximum number of orthogonalizations allowed.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default value is: 30

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

NOTE: The maximum number of orthogonalizations The user is not advised to alter solver dimensioning without first contacting IMEX support at CMG for guidance.

Examples:

*NORTH 60

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**

Red-black reduced system preconditioning is used, similar to a D4 reordering scheme. Elimination is performed on the reduced system of grid blocks labeled "black", saving storage and computer time.

***RCM**

Reverse Cuthill-McKee ordering is used. The ordering algorithm attempts to minimize the bandwidth in the L & U factors. Compared to *NATURAL, use of this scheme can result in significant CPU savings for higher values of degree, especially *GAUSS. Savings are more modest when a low value of IDEG is used, since matrix fill is small. For a regular grid with no wells and no null blocks this method is equivalent to D2 ordering.

***RCMRB**

Use reverse Cuthill-McKee ordering, then red/black ordering. For a regular grid with no wells and no null blocks this method is equivalent to D4 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 keyword group.

EXPLANATION:

Examples:

*SORDER *NATURAL

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.

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is:

*SDEGREE 1 (first degree with non-natural ordering)

*SDEGREE 2 (second degree with natural ordering)

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

This keyword controls the maximum degree of fill terms used in the calculation 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 general, larger values of *max_deg* may be required for more difficult problems (extreme permeability contrasts). Larger values of *max_deg* result in more calculations and a longer simulation run time.

Starting with IMEX 97.00, the solver efficiency for higher order (>1) has been improved significantly. Thus it may be feasible to use the *SDEGREE 2 factorization without a large increase in CPU time. However, memory usage has not changed and may limit the use of higher order factorizations when running large models.

Examples:

```
** 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 keyword group.

EXPLANATION:

This keyword selects the pivot stabilization of diagonal submatrix inversion.

Examples:

```
** Use pivot stabilization.  
*PIVOT *ON
```

Maximum Iterations (Optional)

*ITERMAX

PURPOSE:

*ITERMAX is used to specify the maximum number of 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 40.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

This keyword specifies the maximum number of iterations allowed in the iterative solution of the linearized set of equations.

The minimum allowable value for *maxn* is 1. This is recommended with *GAUSS only.

The maximum allowable value for *maxn* is 200.

AIMSOL/PARASOL Switch (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:

Optional keyword. If it is not present in the input-data-file, ***SOLVER *AIMSOL** is assumed.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

This keyword specifies which solver to use. ***SOLVER *PARASOL** is required in order to solve the linear system of equations in parallel. ***JACPAR**, ***JACDOMS**, ***DPLANES** or ***DTYPE** is required to solve the Jacobian problem in parallel.

When installed in the Launcher, the Parallel version of IMEX is controlled via the command-line arguments:

`-jacpar`

which is equivalent to

***JACPAR *ON**

and overrides the **DTYPE/DPLANES** keywords in the data set, and

`-parasol n`

which is equivalent to the

***SOLVER *PARASOL**

***PPATTERN *AUTOPSLAB n**

***PNTHRDS m**

keywords

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 CPU's. Note, for previous versions of IMEX, `-parasol n` would only work for $n = 2, 4, 8, 16$ or 32 and would use **PPATTERN l**, where $l = 2, 3, 4, 8$ or 9 respectively.

In addition the command-line option “`-aimsol`” can be used to override ***SORDER** ***PARASOL** and any PARASOL keywords in the data.

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 | *OFF

DEFINITIONS:

*OFF

Always use Red-Black ordering for a class.

*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 60% of the number of black blocks).

DEFAULTS:

Optional keyword. If it is not present in the input-data-file, the default is *OFF, thus Red-Black ordering is always performed when specified as the ordering method.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group and used 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.

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. Default is the value of *SDEGREE, which by default is 1 for Red-Black and RCMRB ordering, and 2 for natural ordering (see *SDEGREE in the Numerical Methods Section).

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group and used 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. Defaults to *idegaa* + 1 (See *PDEGAA).

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group and used 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.

Increase PDEGAA and PDEGAB for Blocks on Class Boundaries (Optional)

*PDEGRBB

PURPOSE:

Locally increase degree of factorization on the boundary of classes for REDBLACK to mimic what is done for NATURAL ordering.

FORMAT:

*PDEGRBB *itmprbb*

DEFINITIONS:

itmprbb

Locally increase degree of factorization on class boundaries by this amount

DEFAULTS:

Optional keyword. Normally the degree of factorization is locally not increased. If PDEGRBB is entered and itmprbb is not specified, itmprbb is set to 1.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group and used with *SOLVER *PARASOL. If PDEGRBB is specified with *SORDER *NATURAL a warning message is issued and the keyword is ignored.

EXPLANATION:

Increase PDEGAA for blocks on the boundary of classes to PDEGAA+itmprbb and increase PDEGAB to PDEGAB+itmprbb.

On the boundary of classes, all blocks are black. Thus there is no induced extra terms due to red block pre-elimination. Typically we use SDEGREE 1 for REDBLACK and SDEGREE 2 for NATURAL.

This implies PDEGAA = 1 and PDEGAB = 2 for SORDER REDBLACK or RCMRB when not using PDERGRBB.

The default for PDEGAA and PDEGAB are PDEGAA = 2 and PDEGAB = 3 for SORDER NATURAL or RCM.

Thus, if PDEGRBB (with default itmprbb=1) is used for a default IMEX SDEGREE 1 case for SORDER REDBLACK or RCMRB, the blocks on class boundaries use PDEGAA = 2 and PDEGAB=3 while other level one class blocks use PDEGAA=1.

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 | ipatrn | *PARTITION |
              *PPARTITION | *GPARTITION | *APARTITION |
              *AUTOP2D *I n / *J m / *K l
```

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 *I *n* / *J *m* / *K *l*

Automatic parallel partitioning in 2 dimensions (any two directions in 3D reservoir model). See IO control section for command-line options for AUTOP2D. 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. There is a command-line option –partition_info, which will print the domains/classes with boundaries, numbers of active blocks into the .out file.

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 mXn 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 1 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 mXn. 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. Further development is planned to fix this.

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.

Please note that 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.

*PARTITION

'class partitioned' '1st major new class' '2nd major new class' 'separator class' (*I|*J|*K) *ind*.

Each line directs the partitioning of the first class into two major and one separator class, with the original class no longer existing after the partition. The partitioning is planar, with the separator parallel to the I, J, or K axis as specified, with index value *ind*. Initially there is the one class 'FIELD'; each line creates three new classes and destroys one, for a net gain of two classes per line. The names serve only to identify the classes while the pattern is being created; they are not referred to thereafter. In principle any number of lines may be entered after *PPATTERN *PARTITION;

Current limit is 64 lines per *PPATTERN keyword.

*PPARTITION

'class partitioned' '1st major new class' '2nd major new class' 'separator class'

Like *PARTITION but simulator decides automatically what direction plane to use and where to put it. The decisions are made to equalize class sizes as much as possible and to minimize the size of the separator class.

*GPARTITION

'class partitioned' '1st major new class' '2nd major new class' 'separator class'

Uses Alan George's rooted-level-structure method to partition 'class partitioned' into 3 parts, 2 large classes and a separator. Like *PPARTITION but doesn't use planes.

*APARTITION

'class partitioned' '1st major new class' '2nd major new class' 'separator class'

"agglomeration partition" -- like *GPARTITION but provides classes somewhat more nearly equal in size, but somewhat less regular in shape.

Example:

The 3-class, 2 level partitioning given in the initial description of the PPATTERN keyword can be realized either with

*PPATTERN *AUTOPSLAB 2

or

*PPATTERN 2

or

*PPATTERN *PARTITION

FIELD' 'Class 1' 'Class 2' 'Class 3' *I 51

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.

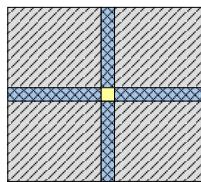
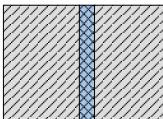
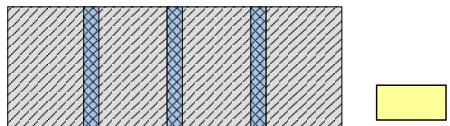
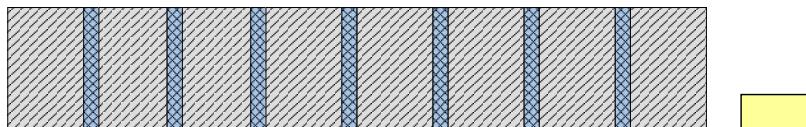
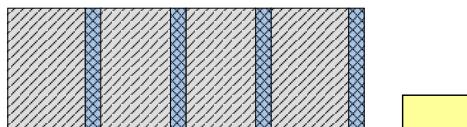
There must be no flow between blocks which are in different classes at the same level.

Example:

Consider a $101 \times 50 \times 10$ 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.

<i>ipatrn</i>	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 <i>ipatrn</i> 2, but separating plane is horizontal	
7	Like <i>ipatrn</i> 2, but with 2 vertical separating planes	

Legend

- Level - 1 
- Level - 2 
- Level - 3 

Figure 1: Geometrical representation of class distribution under different ipatrn

Table 1: Summary of number of levels and classes under different *ipatrns*.

ipatrн	Description	Total levels	Total classes	Class Distribution			Remarks
				Level 1	Level 2	Level 3	
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 <i>ipatrн</i> 2, but separating plane is horizontal	2	3	2	1	0	
7	Like <i>ipatrн</i> 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

Target number of Planes per Jacobian Domain (Optional)

***DPLANES**

PURPOSE:

Choose the target number of planes per Jacobian domain. (In order to use *DPLANES, the parallel computing licensing feature must be active).

FORMAT:

DPLANES *imxdom

DEFINITIONS:

imxdom

Target number of planes per domain.

DEFAULTS:

Optional keyword. Planes are chosen in the dimension direction with the largest number of non-trivial planes. *imxdom* is the number of corresponding non-trivial planes in this direction per domain. If *DPLANES is specified and *imxdom* is not, then the default is 4.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

This keyword allows the user to control the number of planes per Jacobian domain.

*SOLVER *PARASOL is required in order to solve the linear system of equations in parallel. *JACPAR, *JACDOMS, *DPLANES or *DTYPE is required to solve the Jacobian problem in parallel.

When installed in the Launcher, the Parallel version of IMEX is controlled via the command-line arguments:

`-doms`

which is equivalent to

***DPLANES**

and overrides the JACPAR/JACDOMS/DTYPE/DPLANES keywords in the data set, and

`-parasol n`

which is equivalent to the keywords

***SOLVER *PARASOL**

PPATTERN *AUTOPSLAB *n

Note: Changing between parallel and serial options or among parallel options such as

*DPLANES/JACDOMS/JACPAR or different values of *PPATTERN *AUTOPSLAB *n* on a restart run is currently not supported.

If the original run was parallel (for example `-parasol n -jacpar`) and you wish to run a restart on a machine with fewer CPUs “*m*”, you may use `-parasol n -pnthrds m -jacpar`.

Number of Threads to be Used (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 it is defaulted to one. If parallel Jacobian building or PARASOL is specified, then it is defaulted to the minimum of 2 and the number of processors in the current machine.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

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.

Number of Threads for Parallel Grid-Processing (Optional)

***GRIDPAR**

PURPOSE:

Choose the number of threads to be used for parallel processing of grid-tasks.

FORMAT:

GRIDPAR *ngrthrs

DEFINITIONS:

ngrthrs

Number of threads used for parallel grid-processing.

DEFAULTS:

Optional keyword. If *GRIDPAR 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 *GRIDPAR 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 *ngrthrs* larger than the number of available CPUs will result in setting *ngrthrs* equal to number of CPUs. Setting *ngrthrs* = 1 (via *GRIDPAR 1) will ensure that grid-processing tasks are done in serial mode.

Related command-line option is: `-gridpar ngrthrs`

Complete Storage Grid Array of Jacobian Domain Numbers (Optional)

***DTYPE**

PURPOSE:

*DTYPE explicitly defines the domain numbers of individual blocks.

ARRAY:

*DTYPE

DEFINITIONS:

values

A number indicating a given blocks domain.

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword must be located in the Numerical Methods Data keyword group.

EXPLANATION:

This keyword explicitly sets the Jacobian domains of individual blocks.

Target Number of Jacobian Domains (Optional)

*JACDOMS

PURPOSE:

Choose the target number of Jacobian domains. (In order to use *JACDOMS, the parallel computing licensing feature must be active).

FORMAT:

*JACDOMS *ndoms*

DEFINITIONS:

ndoms

Target number of domains.

DEFAULTS:

Optional keyword. If *JACDOMS is specified and *ndoms* is not, then the default is 2, unless the command-line argument –jacdoms *n* or –parasol *n* is specified (*ndoms* = *n*).

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

This keyword allows the user to control the number of Jacobian domains. For *ndoms*, *ndoms* domains, plus, if needed, an additional “garbage can” domain may be set up. *SOLVER *PARASOL is required in order to solve the linear system of equations in parallel. *JACPAR, *JACDOMS, *DPLANES or *DTYPE 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 min (*ndoms*, number of logical CPU's).

The parallel version of IMEX may also be controlled via the command-line arguments:

–jacdoms *n*

which is equivalent to

*JACDOMS *n*

and overrides the *JACPAR/*JACDOMS/*DTYPE/*DPLANES keywords in the data set, and

–parasol *n*

which is equivalent to the keywords:

*SOLVER *PARASOL

*PPATTERN *AUTOPSLAB *n*

To use *n* logical CPU's where *n* ≥ 2, on a machine with two or more logical CPU's, the command-line options:

“–jacdoms *n* –parasol *n*”

may be used.

Using command-line arguments override the keywords in the data set. If “`–jacdoms m`” and “`–parasol n`” are both used, then the number of threads to be used will be determined by *m*. If “`–pnthrds l`” is used, then *l* threads will be used, rather than depending on *m* or *n*.

Note: Changing between parallel and serial options or among parallel options such as `*DPLANES/JACDOMS/JACPARD` or different values of `*PPATTERN *AUTOPSLAB n` on a restart run is currently not supported.

If the original run was parallel (for example `–parasol n –jacpar`) and you wish to run a restart on a machine with fewer CPUs “*m*”, you may use `–parasol n –pnthrds m –jacpar`.

Target Number of Jacobian Domains based on Parasol Classes (Optional)

***JACPAR**

PURPOSE:

Choose the target number of Jacobian domains. (In order to use *JACPAR, the parallel computing licensing feature must be active).

FORMAT:

***JACPAR (*ON | *OFF)**

DEFAULTS:

Optional keyword.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

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, *JACDOMS, *DPLANES or *DTYPE 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 min (*ndoms*, number of logical CPU's).

The parallel version of IMEX may also be controlled via the command-line arguments:

`-jacpar`

which is equivalent to

`*JACPAR *ON`

and overrides the JACPAR/JACDOMS/DTYPE/DPLANES keywords in the data set,
and

`-parasol n`

which is equivalent to the keywords

`*SOLVER *PARASOL`

`*PPATTERN *AUTOPSLAB n`

To use *n* logical CPU's where *n* \geq 2, on a machine with two or more logical CPU's, the command-line options:

“`-jacpar -parasol n`”

may be used.

Using command-line arguments override the keywords in the data set. If “`-pnthrds l`” is used, then *l* threads will be used, rather than depending on *n*.

Note: Changing between parallel and serial options or among parallel options such as *DPLANES/JACDOMS/JACPAR or different values of *PPATTERN *AUTOPSLAB n on a restart run is currently not supported.

If the original run was parallel (for example –parasol n –jacpar) and you wish to run a restart on a machine with fewer CPUs “ m ”, you may use –parasol n –pnthrds m –jacpar.

Specify Behavior When Solver Stalls (Optional)

*SOLSTALL

PURPOSE:

*SOLSTALL is used to override the default behavior when the solver stalls.

FORMAT:

*SOLSTALL *CONTINUE | *REPEAT

DEFINITIONS:

*CONTINUE

When the Solver stalls, that is: the calculated multiplier for the current linear solver iteration according to GMRES is very small, the iteration stops and the run continues.

*REPEAT

When the Solver stalls, that is: the calculated multiplier for the current linear solver iteration according to GMRES is very small, the iteration stops and the timestep repeats (with a smaller timestep size).

DEFAULTS:

Optional keyword. If it is not present in a data set the default *CONTINUE is used.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

When *SOLSTALL *CONTINUE is used, if the calculated multiplier for the GMRES iteration is very small, the solver iteration stops and the run continues, normally the simulation continues to the next Newton's iteration or to the next timestep if the Newton's iteration has converged. When *SOLSTALL *REPEAT is used, the iteration stops as before, but the timestep is repeated with a smaller timestep size.

Examples:

```
*SOLSTALL *CONTINUE
```

Sub-matrix Inversion Algorithms (Optional)

***MATINV**

PURPOSE:

Specify algorithms for block diagonal sub-matrix inversion.

FORMAT:

***MATINV *COFACTOR | *NOCOFACTOR | *OLD**

DEFINITIONS:

***COFACTOR**

Cofactor-based algorithms are to be used for inverting all the block diagonal matrices.

***NOCOFACTOR**

No cofactor-based algorithms are to be used for inverting the block diagonal matrices. Gaussian elimination is used instead.

***OLD**

Cofactor-based algorithms are to be used only for inverting the 3×3 block diagonal matrices. Prior to version 2014.10 this was the only method available.

DEFAULTS:

Optional keyword. If it is not present in a data set the default ***COFACTOR** is used.

CONDITIONS:

This keyword must be located in the Numerical Methods Control keyword group.

EXPLANATION:

Keyword ***MATINV** specifies the algorithm for block diagonal sub-matrix inversion. With ***COFACTOR**, cofactor-based algorithms will be used for inverting the block diagonal 2×2 , 3×3 and 4×4 sub-matrices. Cofactor techniques tend to be slightly more accurate with less round-off error, require no pivoting, and are faster than Gaussian elimination. The 4×4 matrix inversion uses the expansion by 2×2 sub-matrices that involves fewer arithmetic operations. For details, see David Eberly, “The Laplace Expansion Theorem: Computing the Determinants and Inverses of Matrices”, Geometric Tools, LLC, 2008. Cofactor techniques were already used for 3×3 block diagonal matrices (***OLD**) in previous versions of IMEX. The fluid models affected by this change are OILWATER and GASWATER (2×2) and MISCG, MISNCG, POLY, API-INT and BLACKOIL_SEAWATER (4×4).

Examples:

```
*MATINV *COFACTOR
```


Well and Recurrent Data

Notes on Well and Recurrent Data

The *DATE or *TIME keyword indicates a new well change time. Any well keywords that appears 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 indicates the grid blocks where the well is perforated. Refer to Appendix B for the calculation of well index from the well geometries and other information for the well model used.

The type of well is described by the keywords *PRODUCER or *INJECTOR. They must be followed by the operating conditions and constraints of the well indicated by the *OPERATE and *MONITOR keyword. For injectors, the injection fluid composition must also be specified by the *INCOMP keyword. Wells can be connect 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 IMEX. Any operating condition and its value can be altered at a later time using a single keyword *TARGET

A well can be shut in at any time by the keyword *SHUTIN and reopened at a later time using the keyword *OPEN. It can also 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. It allows the user to meet prespecified targets for a group of wells by allowing the program to open (drill) new wells to meet a specified target.

The wellbore hydraulics model relates the bottom-hole pressure to the well head pressure of a well. This model must be activated for wells with well head pressure constraints specified. This could be achieved by entering the wellbore data using the *PWELLBORE or *IWELLBORE keywords. These keywords must be entered after the well type specification keywords (*PRODUCER or *INJECTOR).

The simulation is started by *RUN and terminated by *STOP. The timestep sizes 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.

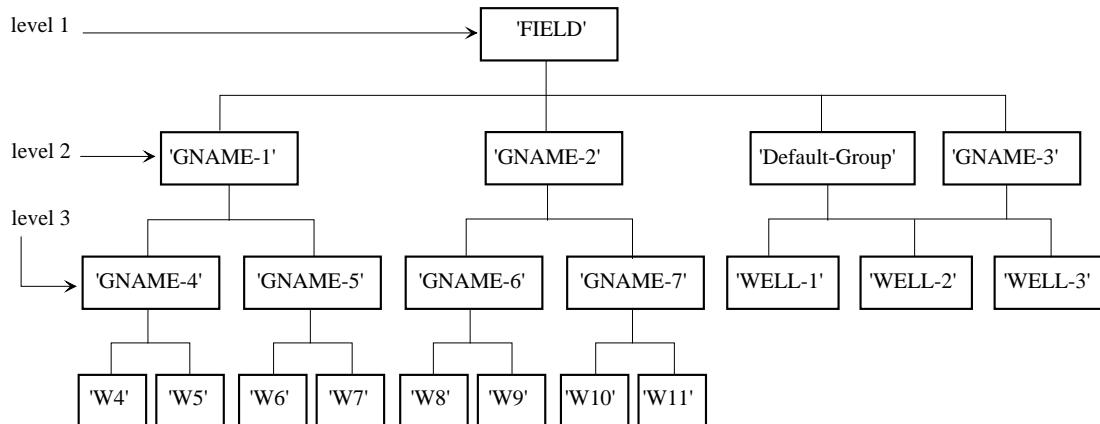
In this section of the User manual all *MODSI units are the same as *SI units except for pressures which are in kg/cm² for *MODSI.

The use of quotes in well names is supported in a consistent manner. To use a double quote in a well name, use a single quote as a well string delimiter. To use a single quote in a well name, use a double quote as a well string delimiter.

Well Management and Group Control

There is no limitation on the maximum of group levels in the group hierarchy; 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. Wells or groups may be connected to a second-level or lower-level group, but not a mixture of the two.

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:

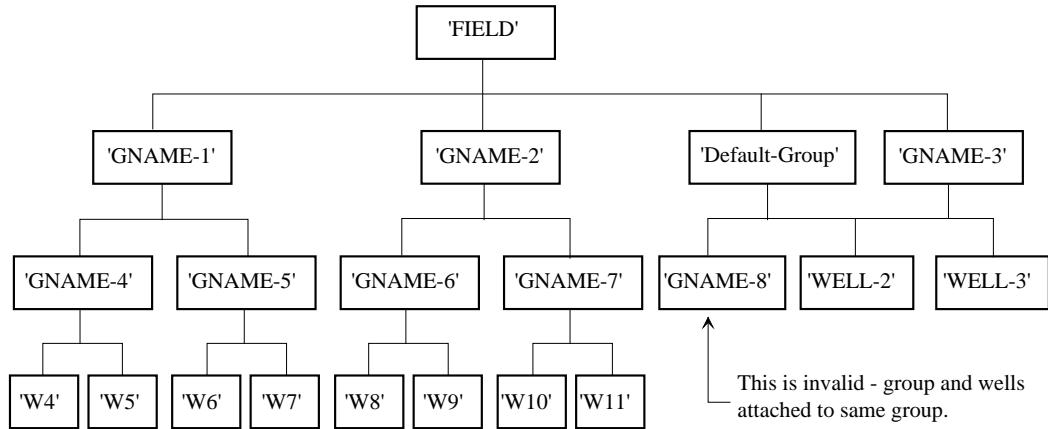
```
*GROUP 'G1' *ATTACHTO 'G3'  
*GROUP 'G2' *ATTACHTO 'G4'
```

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 IMEX data sets from previous versions that do not have gathering centers can be run unchanged with this version of IMEX. 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 with the name 'Default-Group'.

An example of an invalid well-management hierarchy is given below:



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 specified by using the *GUIDEP and *GUIDEI keywords. 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. For describing more complex well trajectories such as multilateral wells, new connection keywords, FLOW-TO (producers) and *FLOW-FROM (injectors) are now supported under the *PERF keyword. In addition a layer may be assigned a status of "*CLOSED" to indicate that the layer is introduced purely to define the well trajectory. Such a layer is not perforated and cannot produce or inject any fluids. Greater detail and examples are provided in this manual under the *PERF keyword.

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 reopening (*AUTOLAYER), and monitoring the well for possible reopening when the GOR or water-cut improves (*AUTOWELL).

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. *ONTIME), 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.

Please note that it is possible to accidentally overwrite group properties, constraints and actions. When all the groups have been defined using *GROUP cards and all group actions, properties and constraints have been specified using card such as *GCONI, *GCONP and *GCONN. A further specification of any previous or new group using a *GROUP keyword requires the user to completely respecify all group actions, properties and constraints.

If the group actions, properties and controls are not respecified, they are lost.

The same is not true for moving wells between groups by using the *ATTACHTO keyword. This can be done without group action respecification.

Examples:

The first example below has a new group defined after group 1 actions are specified. Group 1 controls would be lost.

The second example shows the proper way of specifying multiple groups.

The third example below has a new group defined after group 1 actions are specified, but the lost group controls are respecified.

Example 1: Group 1 Actions lost

```
*GROUP 'Group1' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GROUP 'Group2' *ATTACHTO 'Field'  
*GCONP 'Group2'  
  *MAX      *STG 110000.0  
  *GTARGET *STO  900.0
```

However in the cases below all group controls would be maintained

Example 2: Group definitions grouped together followed by all group control specifications.

```
*GROUP 'Group1' *ATTACHTO 'Field'  
*GROUP 'Group2' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GCONP 'Group2'  
  *MAX      *STG 110000.0  
  *GTARGET *STO  900.0
```

Example 3: Group 1 group control definitions lost and then respecified.

```
*GROUP 'Group1' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GROUP 'Group2' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GCONP 'Group2'  
  *MAX      *STG 110000.0  
  *GTARGET *STO  900.0
```

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 16 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.

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, water and solvent only. Oil 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 / close layers. Thus if the well is perforated horizontally, the behavior of this option may be unpredictable.
6. Please see the *GROUP Keyword for further limitations

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:

*WRST, *WPRN, *OUTPRN, *WSRF and *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:

*RANGE, *REFINE, *TRANSI, *TRANSJ, *TRANSK, *TRANLI,
*TRANLJ, *TRANLK, *PERMI, *PERMJ, *PERMK, *CTYPE

Recurrent Data Use Limitations:

*PERMI: (1) cannot be used with *DYNAGRID, (2) refined grid inheritance not done, thus it is always necessary to explicitly refer to the children blocks when altering *PERMI in LGRs.

*PERMJ: (1) cannot be used with *DYNAGRID, (2) refined grid inheritance not done, thus it is always necessary to explicitly refer to the children blocks when altering *PERMJ in LGRs, (3) no *EQUALSI functionality.

*PERMK: (1) cannot be used with *DYNAGRID, (2) refined grid inheritance not done, thus it is always necessary to explicitly refer to the children blocks when altering *PERMK in LGRs, (3) no *EQUALSI functionality.

*CTYPE: 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:

*RTYPE, *SWCON, *SWCRIT, *SORW, *SORMAX, *SORG, *SOIRW
*SLCON, *SGCON, *SGCRIT, *KRWIRO, *KROCW, *KRGCL, *KROGCG
*PCWMAX, *PCWMIN, *PCWIMIN, *PCGMAX, *PCGMIN, *PCGIMIN
*JFWMAX, *JFWMIN, *JFGMAX, *JFGMIN, *NDARCYCOR

From the Numerical Methods Control section, these keywords may be changed:

*DTMAX, *DTMIN

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.

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: 1.0

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 according to the fraction of the day given.

For example, a well change date of August 19, 1988, at noon, is entered as:

*DATE 1988 08 19.5

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 length of time after the simulation starting date that the well will be changed (days | days | mins | days).

DEFAULTS:

Conditional keyword. Default for starting time only is: 0.0

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 this keyword and the *DATE keyword do not appear immediately following the *RUN keyword, the date 0000 01 01 and the 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.

Example:

```
*PRODUCER 2
*OPERATE *MAX *STO 750.0
** At the following time, change the
** primary operating constraint.
*TIME 20.0
*ALTER 2
 500.0
** The following *TIME forces a simulation
** timestep at 365.0 days to force
** printout and plot information at this
** time.
*TIME 365.0
** Simulation stops at 900.0 days.
*TIME 900.0
*STOP
```

Well Change First Time-Step Size (Optional)

*DTWELL,

*DTWAUTO

PURPOSE:

*DTWELL identifies the first time-step size to be used immediately after the current well change. *DTWAUTO specifies how this time-step size is to be reinstated at any simulation time step in response to a significant well action.

FORMAT:

```
*DTWELL      time_size  
*DTWAUTO    (*ALL | *AUTODRL | *OFF)
```

DEFINITIONS:

time_size

A real number to specify the time-step size (days | days | min | days).

*ALL

The starting time-step size will be reinstated automatically at any simulation time step if a well action is deemed significant by the well management, which may include auto-drilling to meet group targets, auto-well / auto-layer due to violation of monitoring constraints, or changes of operating constraint types.

*AUTODRL

The starting time-step size will be reinstated automatically only when the auto-drilling occurs to meet group targets (see keyword *AUTODRILL).

*OFF

The starting time-step size will not be reinstated automatically at any simulation time step unless the user specifies *DTWELL again at a subsequent well change. This is the default when *DTWAUTO is absent.

DEFAULTS:

Optional. Default *DTWELL is *time_size* = 1.0 days for Cartesian or variable depth/variable thickness grid, and *time_size* = 0.025 days for radial grids. If *DTWAUTO is specified without any valid sub-keyword, *ALL is assumed.

CONDITIONS:

*DTWELL and *DTWAUTO must be located in the Well and Recurrent Data keyword group. *DTWELL must be larger than *DTMIN. *DTWAUTO takes the effect only when *time_size* is smaller than the current time step size.

EXPLANATION:

Keyword *DTWELL is required to specify the size of the first timestep in the simulation, and may optionally appear for subsequent well changes.

These keywords are part of recurrent data, and may be changed during the simulation run.

Maximum and Minimum Time-Step 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 (days | days | min | days).

min_time_size

A real number to specify the minimum time-step size allowed (days | days | mins | days).

DEFAULTS:

Optional keywords. Default: *max_time_size* = 365 days and *min_time_size* = 0.001 days.

CONDITIONS:

*DTMAX and *DTMIN may be located in the Well and Recurrent Data keyword group, and may also be located as part of Numerical Methods Control.

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 time-step size is set to *min_time_size*. If the timestep does not converge, 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.

If the maximum or minimum time-step sizes is once defined by a *DTMAX or *DTMIN keyword, the maximum or minimum time-step size remains the same until changed by a subsequent *DTMAX or *DTMIN keyword.

The acceptable range of values for maximum or minimum time-step size is:

	SI Days	Field Days	Lab Min
min	1.0E-10	1.0E-10	1.44E-07
max	1.0E+20	1.0E+20	1.44E+23

Setting Grid Blocks to Implicit or Explicit (Optional)

***AIMSET**

PURPOSE:

*AIMSET initializes grid blocks to implicit or explicit formulations.

ARRAY:

*AIMSET

DEFINITIONS:

values

A number indicating a given block is explicit (0) or implicit (1).

DEFAULTS:

Optional keyword. Default: 0 (explicit)

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group.

EXPLANATION:

This keyword sets blocks to an initial explicit or implicit pattern. After the blocks are set, the active well blocks are set to implicit. Optionally the neighbours or the neighbours plus their neighbours (see diagram in *AIMWELL) are set to implicit by using *AIMWELL *WELLN (or *WELLNN).

If the keyword is left out at subsequent *DATE or *TIME keywords, then the implicitness of the blocks does not change except for those well blocks which are beginning operation. These blocks are set to implicit.

For example,

*AIMSET *CON 0

sets all blocks to explicit (except for the active well blocks). This is the default.

During a simulation run, the default would result in blocks being switched to implicit as required.

*AIMSET *CON 1

sets all blocks to implicit.

The *AIMSET keyword is applied first. If either keyword strings, *AIMWELL *WELLN or *AIMWELL *WELLNN, appear then these are used next.

In general *AIMSET and *AIMWELL should be omitted. If you are doing a field study and wish to adjust only the rates as part of a history match, then omit the keyword. But on the first well opening time use a small time-step size (*DTWELL) such as 1.0 days. For subsequent well changes, time-step sizes, typically as large or larger than thirty days, may be used.

If you are modelling a process where you inject into the well and then produce from it, then a small time-step size should be used at the start of each cycle. Then the combination *AIMSET *CON (with, optionally, *AIMWELL) may be used to set most of the blocks to explicit at the beginning of each cycle, in order to reduce execution time.

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 well blocks and neighbours of active well blocks and well blocks to implicit.

*WELLNN

This subkeyword sets well blocks, the neighbours of active well blocks and the neighbours of the neighbours to implicit.

DEFAULTS:

Optional keyword. Default: Only well blocks are set to implicit.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group.

EXPLANATION:

Observing the diagram, *AIMWELL on its own sets the block marked W to implicit. W denotes the well block.

Observing the diagram, *AIMWELL *WELLN sets the blocks marked W and N to implicit. W denotes the well block. N denotes neighbour blocks.

Observing the diagram, *AIMWELL *WELLNN sets the blocks marked W,N and NN to implicit. W denotes the well block. N denotes neighbour blocks and NN denotes neighbours of neighbours.

J = 5		NN		
J = 4	NN	N	NN	
J = 3	NN	N	W	N
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 *AIMSET keyword is applied first. If either keyword strings, *AIMWELL *WELLN or *AIMWELL *WELLNN, appear then these are used next.

In general *AIMSET should be omitted (i.e. defaults to *CON 0). For all but the largest field studies, where the number of total well perforations rival the number of grid blocks, *AIMWELL *WELLNN should be used.

Set Frequency of Initialization of Bottom-Hole Pressure (Optional)

*WELLINIT

PURPOSE:

*WELLINIT allows the user to specify that bottom hole 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:

*WELLINIT (well_list) *ITER | *CHANGE | *TIMESTEP

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which the initialization frequency applies. See ‘[Using Wildcards in Well Lists](#)’ in the Tutorial Section for more information about using wildcarding. *WELLINIT will apply to ALL wells with the global initialization frequency if well_list is absent.

*CHANGE

Indicates that bottom hole 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 bottom hole pressures of rate- or implicit well head pressure-constrained wells are to be reinitialized at the beginning of each timestep.

*ITER

Indicates that bottom hole pressures of rate- or implicit well head pressure-constrained wells are to be reinitialized before each Newtonian iteration.

DEFAULTS:

Optional keyword. If *WELLINIT does not appear in the data set, *WELLINIT *ITER is the default global well initialization frequency in IMEX.

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 *CHANGE may give a sufficiently accurate initial bottom hole pressure that the Newtonian iterations for the well constraint equation converge rapidly. In difficult cases, however, the well equations may converge slowly; in such cases, invoking *WELLINIT *Timestep or *WELLINIT *ITER may facilitate convergence of the Newtonian iterations. The *WELLINIT *ITER option requires more CPU time per Newtonian iteration, but may result in an overall CPU time saving if the number of Newtonian iterations is reduced significantly, as is frequently the case in typical IMEX runs. *WELLINIT *Timestep is intermediate between the other two frequencies.

Sometimes only a few wells in a large field require the *WELLINIT *ITER or *WELLINIT *Timestep treatment; in such cases the global treatment may be set to *CHANGE using the first format above and the problem wells may be flagged for special treatment by using the second format above, including a well list.

In IMEX the global initialization frequency applies to all wells.

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.

PDRAIN Method (Optional)

***PDRAIN-METHOD**

PURPOSE:

*PDRAIN-METHOD controls the method used to calculate an average spatial pressure near a well to approximate its drainage pressure when calculating well productivity index (PI). When using “*OUTSRF *WELL *PDRAIN”, the average pressure (\bar{P}_d) is made up of weighted averages of well layer block pressures and neighbouring block pressures (areal not vertical neighbours). \bar{P}_d is a proxy for the pressure at the drainage radius in the calculation of the steady-state productivity index of each well. The user has the option of choosing either the well layer transmissibility (see *PERF, *GEOMETRY keywords) or the grid block pore volume for weighting the spatial average.

FORMAT:

*PDRAIN-METHOD *fraction1* | *VOL *fraction2*

DEFINITIONS:

fraction1

A real number specifying the weighting factor in averaging the pressure of each well layer block pressure P_i and neighbouring block pressures P_j to determine an average layer pressure $\bar{P}_{ave,i}$. *fraction1* can be assigned any real number between 0.0 and 1.0 (See explanation).

fraction2

A real number specifying the weighting factor in determining well \bar{P}_d . This controls how \bar{P}_d is weighted between a well layer transmissibility weighted pressure and a block volume weighted pressure. *fraction2* can be assigned any real number between 0.0 and 1.0 (see Explanation).

*VOL

Using this keyword indicates that the average pressure of each well layer block pressure P_i and neighbouring block pressures P_j to create $\bar{P}_{ave,i}$ is weighted according to pore volume (see Explanation).

DEFAULTS:

For IMEX, the default *PDRAIN method is *PDRAIN-METHOD (*fraction1 fraction2*) with following values *fraction1* = 0.5 and *fraction2* = 1.0

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

*OUTSRF *WELL *PRDRAIN must be in the IO CONTROL Keyword group to use this more complex calculation of \bar{P}_d or it will have no effect on well PI output. *PDRAIN-METHOD controls how \bar{P}_d is calculated.

EXPLANATION:

The “productivity index” is an important parameter which defines the production potential of a well. A well’s productivity index is defined as the ratio of a production rate divided by the pressure drop. The pressure drop here is the difference between the well’s bottom hole pressure and the pressure at the “drainage radius”, r_d . Thus the productivity index (PI) is:

$$PI = \frac{Q_p}{P_d - P_{wf}}$$

Where Q_p is the production rate of the chosen phase, P_d is the pressure at the drainage radius, and P_{wf} is the bottom hole flowing pressure. Practically, it is very hard to define the drainage radius accurately. While both Q_p and P_{wf} can be measured directly, the drainage radius cannot. It is therefore more common to express the pressure drawdown in terms of a measurable quantity. An alternative approach is to replace P_d with the a spatial average pressure of well layer grid blocks and optionally their adjacent neighbours (areal neighbours). The weighting factor in the averaging may be defined according to either the well layer transmissibilities (WI , see *PERF, *GEOMETRY keywords) or grid block pore volumes. WI_i is the geometric part of each layers transmissibility and does not include phase mobility or total mobility contributions.

IMEX uses the following procedure to calculate the above mentioned special average pressure (\bar{P}_d). $fraction1$ is used as weighting factor in calculating the local average pressure at each well layer block (i) and its adjacent neighbours (j):

$$\bar{P}_{ave, i} = fraction1 \times P_i + (1 - fraction1) \times \frac{\sum_j P_j}{N_j}$$

Where P_i is the pressure of the well layer grid block itself, $\sum_j P_j$ is the summation over the pressure of its adjacent areal neighbours and N_j is the total number of neighbours.

Then the average of pressure over all of the connections of the well, weighted according to their well layer transmissibility, WI_i will be calculated as follows:

$$\bar{P}_T = \frac{\sum_i WI_i \bar{P}_{ave, i}}{\sum_i WI_i}$$

The next step is calculating the pore volume weighted average (\bar{P}_{pv}). This is simply the average over all of the layers and areal neighbours of the well, weighted according to their pore volume

$$\bar{P}_{pv} = \frac{\sum_k V_k P_k}{\sum_k V_k}$$

Where P_k and V_k are the pressure and the pore volume of each block accordingly. The set of grid blocks k is selected from the blocks containing connections of the well and their areally adjacent neighbours. The choice of $fraction1$ defines the averaging scheme as follows:

- If ($fraction1 = 1$) just the well layer grid block (i) will contribute in summation
- If ($fraction1 = 0$) just the adjacent neighbours (j) of well layers will contribute in the summation
- For any other value of ($fraction1$) both (i) and (j) will contribute in summation

Finally, the summation of \bar{P}_T and \bar{P}_{pv} weighted by *fraction2* is used to estimate the drainage pressure in the *PI* calculation pressure drop as:

$$\bar{P}_d = \textit{fraction2} \times \bar{P}_T + (1 - \textit{fraction2}) \times \bar{P}_{pv}$$

And the modified productivity index will be calculated as:

$$PI = \frac{Q_p}{\bar{P}_d - P_{wf}}$$

Special case: The user has the option of replacing *fraction1* with sub-keyword *VOL. In this case the $\bar{P}_{ave,i}$ will be replaced by the average of the pressures weighted by pore volume:

$$\bar{P}_{ave,i} = \frac{V_i P_i + \sum_j V_j P_j}{V_i + \sum_j V_j}$$

Where P_i and V_i are the pressure and pore volume of the well layer grid block itself; $\sum_j V_j P_j$ is the summation over the product of pore volume and pressure of its areally adjacent neighbours.

Example:

*PDRAIN-METHOD 0.5 0.75

or

*PDRAIN-METHOD *VOL 0.75

Group Identification (Optional)

*GROUP

PURPOSE:

*GROUP is used to identify gathering centers, groups and platforms. This keyword is useful for building a tree structure of groups.

FORMAT:

*GROUP '*child_1*' ... '*child_n*' *ATTACHTO '*parent*'

DEFINITIONS:

'*child_1*', ... , '*child_n*'

Names of child groups that are attached to the '*parent*' group. Each group is identified by a unique name up to 16 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:

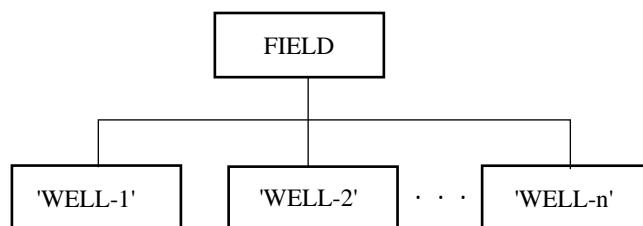
Optional keyword. If no *GROUP line appears in the data, no group structure exists in the simulation and well rates and cumulatives are summed directly into a field cumulative. This is reported as the FIELD cumulative in output, but no group called 'FIELD' actually exists in this case and no group control 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:

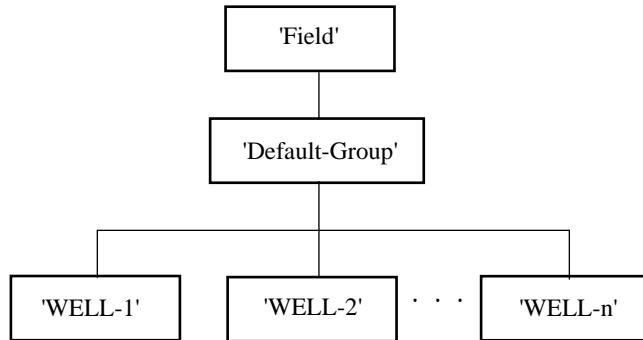


Quotes were purposely omitted around FIELD above to emphasize that no group with the name 'FIELD' actually exists in this case. No group control is possible in this case.

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.

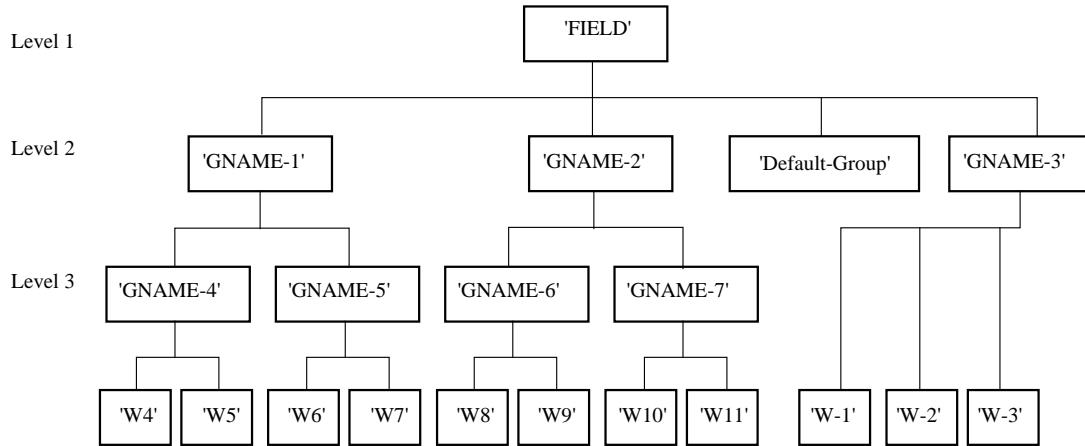
EXPLANATION:

This keyword identifies the group, by name, and assigns it to a parent group. There is no limitation on the levels of groups 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 16 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 which do not explicitly have a parent group assigned. Level 2 groups can have either wells or groups attached to them, but not a mixture of the two; i.e. 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-1 groups as their parents. Level n groups can have either wells or groups attached to them, but not a mixture of the two.

Examples of valid and invalid well control trees are given below.

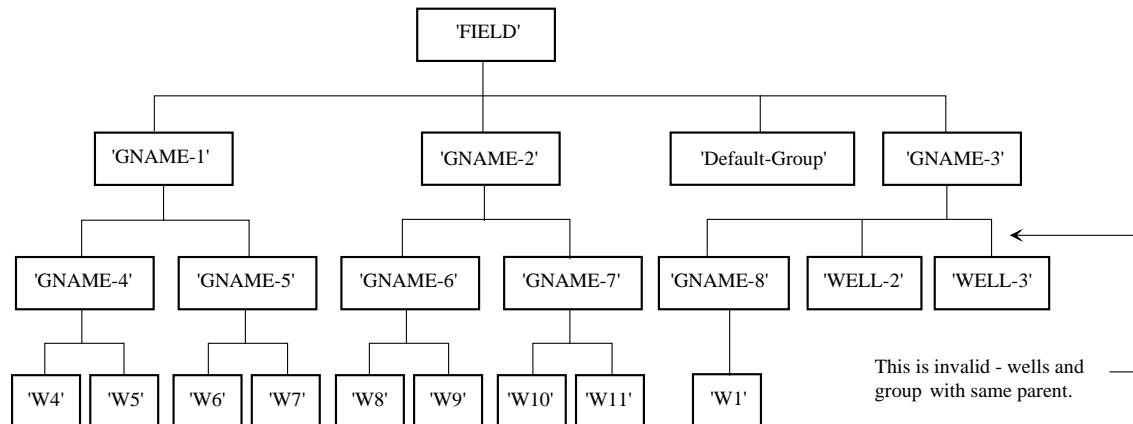
Valid example of a well control hierarchy



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:



Please note that it is possible to accidentally overwrite group properties, constraints and actions. When all the groups have been defined using *GROUP cards and all group actions, properties and constraints have been specified using card such as *GCONI, *GCONP and *GCONN. A further specification of any previous or new group using a *GROUP keyword requires the user to completely respecify all group actions, properties and constraints.

If the group actions, properties and controls are not respecified, they are lost.

The same is not true for moving wells between groups by using the *ATTACHTO keyword. This can be done without group action respecification.

Examples:

The first example below has a new group defined after group 1 actions are specified. Group 1 controls would be lost.

The second example shows the proper way of specifying multiple groups.

The third example below has a new group defined after group 1 actions are specified, but the lost group controls are respecified.

Example 1: Group 1 Actions lost

```
*GROUP 'Group1' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GROUP 'Group2' *ATTACHTO 'Field'  
*GCONP 'Group2'  
  *MAX      *STG 110000.0  
  *GTARGET *STO  900.0
```

However in the cases below all group controls would be maintained

Example 2: Group definitions grouped together followed by all group control specifications.

```
*GROUP 'Group1' *ATTACHTO 'Field'  
*GROUP 'Group2' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GCONP 'Group2'  
  *MAX      *STG 110000.0  
  *GTARGET *STO  900.0
```

Example 3: Group 1 group control definitions lost and then respecified.

```
*GROUP 'Group1' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GROUP 'Group2' *ATTACHTO 'Field'  
*GCONP 'Group1'  
  *MAX      *STG 100000.0  
  *GTARGET *STO  1000.0  
*GCONP 'Group2'  
  *MAX      *STG 110000.0  
  *GTARGET *STO  900.0
```

Well Identification (Required)

*WELL

PURPOSE:

*WELL is used to identify wells.

FORMAT:

```
*WELL    wnum 'well_name'    (*VERT ib jb) (*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. The use of quotes in well names is supported in a consistent manner. To use a double quote in a well name, use a single quote as a well string delimiter. To use a single quote in a well name, use a double quote as a well string delimiter.

*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)

ib jb

I and J direction grid block indices for the vertical well. (Required with *VERT.) There is no default value.

*FRAC *frac*

This optional subkeyword indicates that the well 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)

DEFUALTS:

Required keyword. No defaults. Minimum required is:

*WELL 'well_name'

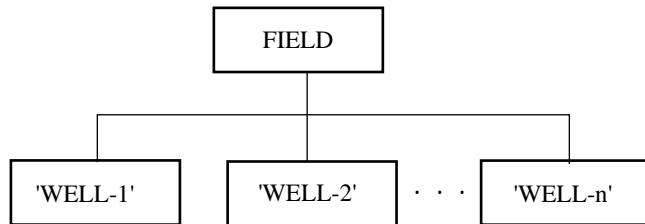
-or-

*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. It also identifies vertical wells.

'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:

```
** Both well sequence number and name  
** are specified.  
*WELL 1 'Producer1'  
  
** Only well sequence number is specified.  
** In this example the program will assign  
** the name 'WELL-2' to the well.
```

```
*WELL 2 *VERT 12 14
```

```
** Only the well name has been specified.  
** In this example the program will assign  
** 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° corner, 1/8 in a 45° 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 and *MONITOR.
2. Well indices entered directly via *PERF or *PERFV (default). The index used will be *wi·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 Backflow Model (Optional)

*XFLOW-MODEL

PURPOSE:

*XFLOW-MODEL is used to identify the well backflow, crossflow model to be used.

FORMAT:

*XFLOW-MODEL (*well_list*)
(*FULLY-MIXED | *ZERO-FLOW | *SOURCE-SINK)

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. 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, then it is modelled using a fully mixed assumption. This option is based on one from Phillips Petroleum (cf. Coats et al, SPE 29111, 1995).

*ZERO-FLOW

This option will zero the flow of a backflowing layer at the Newton iteration level as soon as it is detected. It will reopen the layer at the Newton iteration level, as soon as it will no longer backflow.

*SOURCE-SINK

This option uses the block phase mobilities to calculate rates for backflowing wells. This approach may result in unphysical results and is not recommended. It is provided only to allow comparison with IMEX 96.11 and earlier runs.

DEFAULTS:

The default cross flow option is *FULLY-MIXED for all wells except injectors which do not use mobility weighting (i.e. use the *UNWEIGHT) option. “Unweighted” injectors are converted to use the *ZERO-FLOW cross flow option.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group. For the *FULLY-MIXED and the *ZERO-FLOW options, *WELLINIT *ITER is recommended. *UNWEIGHT injectors are automatically converted to use the *ZERO-FLOW option regardless of the option specified.

EXPLANATION:

This keyword identifies the well crossflow, backflow model to be used as a function of well number and/or name.

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'  
*XFLOW-MODEL 1:2 *FULLY-MIXED  
*XFLOW-MODEL 'Prod. 54-233a' *ZERO-FLOW
```

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 model param
  where
  model = *MODEL-AG | *MODEL-HOMOGENEOUS |
           *MODEL-MOMENTUM | *MODEL-DRIFTFLUX
  param = (*HEADRROUGH rrough)
          (*HEADSURFT surft)
          (*HYRADIUS hrad)
```

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. 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.

*MODEL-HOMOGENEOUS

Homogeneous well friction model (*GRAV-FRIC sub-keyword).

*MODEL-AG

The Aziz-Govier well friction model (*GRAV-FRIC sub-keyword)

*MODEL-MOMENTUM

The momentum equation well friction model (*GRAV-FRIC sub-keyword).

*MODEL-DRIFTFLUX

The drift flux well friction model (*GRAV-FRIC sub-keyword).

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.

HEADSURFT *surft

Sub-keyword of *GRAV-FRIC introducing input of the value of wellbore gas-liquid surface tension (dynes/cm) used for the wellbore frictional head calculation. The default value of *surft* is 30 dynes/cm.

HYRADIUS *hedrad

Sub-keyword of *GRAV-FRIC introducing input of the value of hydraulic radius (m | ft | cm) 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.

DEFUALTS:

For IMEX, the default head method is *GRAVITY.

The default *GRAV-FRIC method is *MODEL-AG.

The default for *HEADRROUGH is 1.0e-4.

The default for *HEADSURFT is 30.0 dynes/cm.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

EXPLANATION:

This keyword identifies the well head calculation 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 3 *ZERO-HEAD  
*HEAD-METHOD 2 *GRAVITY
```

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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See '**Using Wildcards in Well Lists**' in the Tutorial Section for more information about using wildcarding. 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/pressure oscillations from timestep to timestep. It is recommended that the default *XFLOW-MODEL *FULLY-MIXED is used when the ITERATIVE option is specified.

DEFAULTS:

For IMEX, 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 *ITERATIVE suffices to eliminate oscillations in most models without the use of BHPHEADINIT.

If the well bore pressure in completion k is denoted as p_{wk} and the surrounding grid block pressure as p_{bk} , then the flow rate in the well bore 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 h_k 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 p_{wk} (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 h_k values for which $p_{wk} - (BHP + h_k) = 0$ 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. It is recommended that the default *WELLINIT *ITER option is used when specifying *BHPHEADINIT *COMBINED

FORMAT:

*BHPHEADINIT (*well_list*) *SEPARATE | *COMBINED

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See '**Using Wildcards in Well Lists**' in the Tutorial Section for more information about using wildcarding. 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 accounts for the effects of varying reservoir properties during the timestep.

DEFAULTS:

For IMEX, the default setting is *SEPARATE.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

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 or of layer rates.

When *SEPARATE is in effect, a well's BHP is adjusted iteratively until $T(\text{calculated}) - T(\text{specified}) = 0$, 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 $T(\text{calculated}) - T(\text{specified}) = 0$ 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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

*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 bottom hole 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 Type Definition (Required)

*PRODUCER, *INJECTOR

PURPOSE:

*PRODUCER indicates that the well identified by wellnum is a producer.

*INJECTOR indicates that the well identified by wellnum is an injector.

FORMAT:

*PRODUCER *well_list*
-or-
*INJECTOR (*MOBWEIGHT | *UNWEIGHT) *well_list*

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

*MOBWEIGHT

This subkeyword defines a total mobility weighted injector.

*UNWEIGHT

This subkeyword defines an unweighted injector.

DEFAULTS:

Required keywords. For *INJECTOR, *MOBWEIGHT is the default.

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 *PWB or *IWELLBORE lines if present. Perforation data may precede the well type specification. *GEOMETRY cannot be used with *INJECTOR *UNWEIGHT.

The *FULLY-MIXED cross flow option cannot be used with *INJECTOR *UNWEIGHT option, the *ZERO-FLOW option will be used instead.

A given well may be declared *SHUTIN any time after it has been defined with a *WELL statement. Wells are initialized to the *SHUTIN status when *WELL is issued for the well; *SHUTIN may be entered for the well but need not be.

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.

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 'Injector'  
*INJECTOR *MOBWEIGHT 1 ** Define the type of well 1.  
*INCOMP *GAS          ** 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'  
*PRODUCER 'Producer' ** 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  
** At this point you may shut in any of these wells.  
** Shut in the injector  
*TIME 900.0  
*SHUTIN 1
```

Shut in Wells above Formation (Optional)

*MODELHUT,

*EQUILIBRATE

PURPOSE:

*MODELHUT 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:

*MODELHUT (*well_list*) (*ON | *OFF)
*EQUILIBRATE *epsmds*

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. 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 | cm³/min) must be a non-negative, real number.

DEFAULTS:

Optional keyword. Default is *MODELHUT *OFF for all wells. Use of *MODELHUT with no occurrence of *ON or *OFF has the effect of setting *MODELHUT *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.

The *MODELHUT option cannot be applied to a well whose type does not support the *FULLY-MIXED back/cross flow model, such as the unweighted injectors.

EXPLANATION:

Various actions can cause a well to become shut in: *SHUTIN, effective *WLISHUT, *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 *MODELHUT allows fluid to equilibrate for some time in the wellbore before all the layers are closed (“model-shut”).

Keyword *MODELHUT 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 *WLISOPEN, *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. *SHUTIN may be issued for a well any time after it has been defined with a *WELL statement. A producer or an injector must be fully defined, including the constraints before *OPEN or *AUTODRILL may be issued for the well. A well with a monitor of *AUTOWELL or *AUTOLAYER can reopen a shut in well unless this monitor is removed by refining the well operating conditions.

*OPEN indicates that a previously shut-in well is reopened.

*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.

FORMAT:

*SHUTIN | *OPEN | *AUTODRILL *well_list*

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

DEFAULTS:

Optional keywords. No defaults.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group, and must be located after the well type (*PRODUCER or *INJECTOR) as well as the operating constraints have been defined.

EXPLANATION:

These required keywords are used for well status. If a status of *AUTODRILL is specified, the well must be defined completely, i.e. all its perforations and operating constraints must be specified. When *WELL is first issued for a well, its status is initialized as *SHUTIN, and *SHUTIN may be explicitly issued for a well any time after that, though it need not be.

Example:

```
*WELL 1 'Injector' *ATTACHTO 'Group-1'  
*INJECTOR *MOBWEIGHT 1 ** Define the type of well 1.  
*INCOMP *GAS ** 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 'Producer'
** At this point you may open any of these wells.
** Open the injector
*TIME 900.0
*OPEN 1
```

Shut and Reopen a List of Wells (Optional)

*WLISTSHUT,

*WLISTOPEN

PURPOSE:

*WLISTSHUT and *WLISTOPEN provide a means to temporarily shut 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_numbers or well_names are 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 the 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	<i>well_list</i>
*WLISTOPEN	(<i>well_list</i>)

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this definition applies. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding.

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.

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, *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 *WLISSTOPEN are taken out of the remaining well list. The entire well list entered with *WLISHTSHUT will be processed if there is no *well_list* following the keyword *WLISSTOPEN.

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.  
*WLISHTSHUT 1:5  
**Shutting a list of wells. Since Well 1 has already been shut in  
before entry of the keyword, it will be automatically excluded from the  
list. Well 3 is temporally disallowed to be auto-drilled.  
  
*TIME 200.  
*ALTER 2  
    400.  
**Well 2 is explicitly altered and is thus removed from the *WLISHTSHUT  
list.  
  
*TIME 300.  
*WLISSTOPEN  
**Reopening wells on the remaining list issued by *WLISHTSHUT, 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 the wellbore pressure drops (reservoir to surface) will be computed for an injector. Tubing data is required if a wellbore model is used. 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 well depth of an injector well (m | ft | cm | m). 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 well length of an injector well (m | ft | cm | m).

rel_rough

A real number specifying the relative well roughness. Dimensionless.

whtemp

A real number specifying the well head temperature (°C | °F | °C | °C).

bhtemp

A real number specifying the reservoir temperature (°C | °F | °C | °C).

wradius

A real number specifying the tubing radius (m | ft | cm | m).

*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 and must precede the well completion keywords.

*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.

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 center 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 center of the grid block in which the well's reference layer is completed.

Relative well roughness is defined as:

$$\frac{\text{maximum roughness size}}{\text{well diameter}}$$

The acceptable range of values for well depth is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for well length is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for the relative roughness is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for wellbore temperature is:

	SI °C	Field °F	Lab °C
min	-1.0E+4	-18032.0	-1.0E+6
max	1.0E+4	18032.0	1.0E+6

The acceptable range of values for reservoir temperature is:

	SI °C	Field °F	Lab °C
min	-1.0E+4	-18032.0	-1.0E+6
max	1.0E+4	18032.0	1.0E+6

The acceptable range of values for well radius is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

Tubing Data for Producers (Conditional)

*PWELLBORE

PURPOSE:

*PWELLBORE specifies that the wellbore model will be used for this producer.

FORMAT:

```
*PWELLBORE (*TABLE)
  wdepth table_number
-or-
*PWELLBORE *MODEL gas_comp
  wdepth wlength rel_rough whtemp bhtemp wradius
```

DEFINITIONS:

*TABLE

This subkeyword specifies that the well-bore hydraulic pressure loss table (input using the *PTUBE1) will be used for pressure loss calculations.

wdepth

A real number specifying the well depth of an producer well (m | ft | cm).

table_number

*PTUBE1 table number.

*MODEL

This subkeyword specifies that the well-bore hydraulic pressure loss FOR SINGLE PHASE GAS PRODUCERS will be calculated by using the friction factor correlation and the equation of state.

wlength

A real number specifying the well length of an producer well (m | ft | cm).

rel_rough

A real number specifying the relative well roughness. Dimensionless.

whtemp

A real number specifying the well head temperature (°C | °F | °C).

bhtemp

A real number specifying the reservoir temperature (°C | °F | °C).

wradius

A real number specifying the tubing radius (m | ft | cm).

gas_comp

Mole fractions of produced gas components in the following order. C1 C2 C3 IC4 NC4 IC5 NC5 C6 CO2. Nine values are required, one for each of the above components.

DEFAULTS:

Conditional keyword. No defaults. If both *TABLE and *MODEL are missing then the default is *TABLE.

CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group immediately following *PRODUCER if the wellbore model is used and before the well completion keywords.

*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.

The *MODEL option for calculating the pressure loss in the wellbore is applicable for single phase GAS PRODUCERS ONLY. Use the pressure loss tables (*TABLE) for multi phase flow in producers.

The composition of a gas producer is NOT used elsewhere in the simulator.

In IMEX the global composition of the gas must be entered in terms of the following components in the order:

C1 C2 C3 IC4 NC4 IC5 NC5 C6 CO2

This is because the properties of the above components are built into the program.

Well depth is the vertical distance from the well head to the center 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 center of the top grid block that is perforated by the well.

Relative well roughness is defined as:

$$\frac{\text{maximum roughness size}}{\text{well diameter}}$$

Example # 1

```
** In the following example *MODEL is used to calculate
** Wellbore hydraulic pressure loss for a gas producer.
** The units being used are *MODSI.
*DATE 1988 01 01
*DTWELL 0.003
*DTMIN 1.0E-06
*WELL 1 'Water Injector'
*INJECTOR 1
*INCOMP *WATER
*OPERATE *MAX *BHP 71.380
```

```

** Well geometry for the producer.
**           rw      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
** Gas producer; uses analytical model.
** Composition C1  C2  C3  IC4  NC4  IC5  NC5  C6  CO2
*PWEELBORE *MODEL 0.77 0.0 0.20 0.0  0.0 0.0 0.0 0.03 0.0
** wdepth wlenth rel_rough whtemp bhtemp wradius
  8400. 8400. 0.0001 60.0 160.0 0.25
*OPERATE *MIN *WHP 50.986
** Well geometry for the producer.
**           rw      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.
**           rw      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.
*PWEELBORE *TABLE 8400.0 1
*OPERATE *MIN *BHP 2098.0
** Well geometry for the producer.
**           rw      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
*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    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 m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for well length is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for the relative roughness is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

The acceptable range of values for wellbore temperature is:

	SI °C	Field °F	Lab °C
min	-1.0E+4	-18032.0	-1.0E+6
max	1.0E+4	18032.0	1.0E+6

The acceptable range of values for reservoir temperature is:

	SI °C	Field °F	Lab °C
min	-1.0E+4	-18032.0	-1.0E+6
max	1.0E+4	18032.0	1.0E+6

The acceptable range of values for well radius is:

	SI m	Field ft	Lab cm
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

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 | cm) 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
```

Composition of Injected Water, Gas, and Oil Phases (Conditional)

*INCOMP

PURPOSE:

*INCOMP specifies the compositions of injected oil, water and gas phases.

ARRAY:

```
*INCOMP (*OIL)
        (*WATER      (cpvalue))
        (*GAS        (*GLOBAL values))
        (*SOLVENT    (*GLOBAL values))
        (*SEAWATER   (fsvalue))
```

DEFINITIONS:

*OIL

This subkeyword identifies an oil injector.

*WATER

This subkeyword identifies a water injector.

*GAS

This subkeyword identifies a gas injector.

*SOLVENT

This subkeyword identifies a solvent injector. Use only with *MODEL
*MISCG or *MODEL *MISNCG.

*SEAWATER

This subkeyword identifies a seawater injector. Use only with *MODEL
*BLACKOIL_SEAWATER or *MODEL *OILWATER_SEAWATER.

*GLOBAL

This subkeyword identifies the global composition in mole fractions. Use
with *GAS or *SOLVENT.

cpvalue

Polymer concentration to be injected in the water phase. Use only with
*MODEL *POLYOW or *MODEL *POLY (kg/m³ | lb/stb | g/cm³).

fsvalue

Volume fraction of seawater injected. Default value is 1.0. Use only with
*MODEL *BLACKOIL_SEAWATER or *MODEL
*OILWATER_SEAWATER (fraction).

values

Mole fractions of components in the order as defined in the Explanation section.

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 *INJECTOR and *IWELLBORE and before the well completion keywords.

Note: The use of the *GLOBAL subkeyword to model wellbore effects depends upon the prior definition of *IWELLBORE for this well. Failure to define *IWELLBORE before *INCOMP *GAS (or *SOLVENT) *GLOBAL is read will cause a read error.

Required for all injection wells.

EXPLANATION:

The keyword *INCOMP identifies the composition of injector fluid.

In IMEX, *INCOMP *GAS (or *SOLVENT) *GLOBAL is required for wellbore calculations (*WHP constraint or when the wellbore model is desired) for gas (or solvent) injectors ONLY.

When *IWELLBORE *TABLE is specified, *GLOBAL and the following mole fraction values need not (but may) be entered.

The composition of a gas or solvent injector is NOT used elsewhere in the simulator.

In IMEX the global composition of the gas (or solvent) must be entered in terms of the following components in the order:

C1 C2 C3 IC4 NC4 IC5 NC5 C6 CO2

This is because the properties of the above components are built into the program.

Usually most of the injected gas should be C1 (i.e. greater than 80% of the total composition) with trace amounts of C2, C3, C4, C5+ and CO2 unless the pseudo-miscible option is used. Typically, solvent is entirely CO2, or predominantly C1, C2, C3 and trace amounts of C4+.

Let us assume the composition is as follows:

77% C1, 20% C3, 3% C6

```
*MODEL *MISCG
:
*RUN
:
*INJECTOR
*IWELLBORE
** wdepth wlengt rough whtemp bhtemp rw
  2305.0  2617.0  0.002   215.0   215.0  0.25
*INCOMP *SOLVENT *GLOBAL
** C1    C2    C3    IC4   NC4   IC5   NC5   NC6   CO2
  0.77   0.0   0.2   0.0   0.0   0.0   0.0   0.03  0.0
```

Well Operating Constraints (Required)

*OPERATE

PURPOSE:

*OPERATE defines the well operating constraints, and the remedial action to be taken when a constraint is violated.

FORMAT:

*OPERATE (*MAX | *MIN) type *value* (action)
-or-
*OPERATE (*PENALTY) type (mode) (*value*) (action)
-or-
*OPERATE (*WCUTBACK) type mode *v1* (*v2* (*v3*)) (action)
where type for *MAX or *MIN is one of:

- *STO
- *STG, *BHG
- *STW, *BHW
- *STS, *BHS
- *STL
- *BHP
- *WHP (*INITIALIZE | *IMPLICIT)
- *BHF
- *DWN, *DWA, *DWB

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
- *STS
- *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 (v_1) of GOR, GLR, WOR, WGR or PRS constraint. The control phase (mode) has to be chosen from one of *STO, *STG, *STW, *STS, *STL 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 ($0 < v_2 < 1$). The cutback reversing process (rate increasing) is activated whenever the well violates the reversal limit (v_3) 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 | cm^3/min) 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 | cm^3/min) 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 cm^3/min) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.
*STS	This subkeyword identifies a surface solvent rate (m^3/day scf/day cm^3/min) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.
*STL	This subkeyword identifies a total surface liquid rate (oil + water) (m^3/day stb/day cm^3/min) 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 + solvent) (m^3/day bbl/day cm^3/min) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. This keyword is applicable to producers only.
*BHW	This subkeyword identifies a reservoir water rate (m^3/day bbl/day cm^3/min) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword.
*BHG	This subkeyword identifies a reservoir gas rate (m^3/day ft^3/day cm^3/min) constraint. Specified rate values are allowed and have the same effect as the *SHUTIN keyword. This keyword is applicable to injectors only.
*BHS	This subkeyword identifies a reservoir solvent rate (m^3/day ft^3/day cm^3/min) constraint. Specified rate values of zero are allowed and have the same effect as the *SHUTIN keyword. This keyword is applicable to injectors only.
*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 cm^3/cm^3). 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 cm^3/cm^3). 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 | cm^3/cm^3). 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 | kPa | kg/cm^2) from all the open layers of the well.

***BHP**

This subkeyword identifies a bottom hole pressure (kPa | psi | kPa | kg/cm^2) constraint. All producers must have a minimum bottom hole pressure constraint. If *MIN *BHP is missing for a producer then a default minimum bottom hole pressure of 101.3 kPa or 14.7 psia 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 | kPa | kg/cm^2) 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 | kPa | kg/cm²) constraint – the difference between the wellbore pressure and the grid block pressure.

Only *MAX 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,

$$\Delta P_d = \pm(P_{block_ref} - P_{well_ref})$$

where the sign \pm 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,

$$\Delta P_d = \max_{l, \text{open}} \pm (P_{block_l} - P_{well_l})$$

which corresponds to the notion of imposing the maximum drawdown constraint to avoid formation damage. *DWB specifies the average drawdown for all open layers:

$$\Delta P_d = \pm \sum_{l, \text{open}} PI_l (P_{block_l} - P_{well_l}) / \sum_{l, \text{open}} PI_l$$

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.

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 violated 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.

v1

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 m³/m³ (561 scf/stb) for *GOR or *GLR;

100 m³/m³ (100 stb/stb) for *WOR;

100 m³/m³ (17.8 stb/scf) for *WGR;

No default for *PRS.

v2

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 1.0. If v2 is missing or the value entered is larger than or equal to 1.0, internally calculated cutback factor will be applied. v1 must be entered before entering v2.

v3

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 v1. The default is to switch off the reversing process. v2 must be entered before entering v3.

*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 with in one timestep. However, the maximum number of total repeats for this timestep is limited by user input (see keyword *MXCNRPT).

DEFUALTS:

Required keywords. No defaults. At least one operating constraint must appear in the data set.

All producers must have a minimum bottom hole pressure constraint. If *MIN *BHP is missing for a producer then a default minimum bottom hole pressure of 101.3 kPa or 14.7 psia will be assigned.

All injectors must have a maximum bottom hole pressure constraint. If *MAX *BHP is missing for an injector then a default maximum bottom hole pressure of 1,000,000 kPa or 147,000 psia will be assigned.

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.

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
---	STG, BHG	STG
---	STS, BHS	STS
STW, BHW	---	STW, BHW
---	---	STL
---	BHF	BHF
DWN, DWA, DWB	DWN, DWA, DWB	DWN, DWA, DWB
--	--	GOR, GLR
--	--	WOR, WGR
PRS	PRS	PRS

For example, to produce a well at a maximum rate of 500 m³/day, with a minimum bottom hole pressure at 2500 kPa and with monitoring constraints of GOR at 2000 m³/m³, 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
*MONTIOR     *WCUT          0.98     *STOP
*MONITOR      *BACKFLOW
```

If, while the primary constraint is active, the bottom hole pressure falls below 2500 kPa then the bottom hole 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

$$\text{Allowed rate} = (\text{Max rate}) \cdot (\text{Base GOR}) / (\text{Measured GOR})$$

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 :

$$PF(gor) = \text{Base_GOR} / \text{Produced GOR}$$

$$PF(glr) = \text{Base_GLR} / \text{Produced GLR}$$

$$PF(wor) = \text{Base_WOR} / \text{Produced WOR}$$

$$PF(wgr) = \text{Base_WGR} / \text{Produced WGR}$$

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:

$$Qo_{\text{new}} = Qo_{\text{max}} \times PF$$

for GOR and WOR penalties,

$$Qg_{\text{new}} = Qg_{\text{max}} \times PF$$

for a WGR penalty, and

$$Ql_{\text{new}} = Ql_{\text{max}} \times PF$$

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,

$$Qo_{\text{new}} = (Qo_{\text{base}} \times PF \times Qo_{\text{old}})^{1/2}$$

for GOR and WOR penalties,

$$Qg_{\text{new}} = (Qg_{\text{base}} \times PF \times Qg_{\text{old}})^{1/2}$$

for a WGR penalty, and

$$Ql_{\text{new}} = (Ql_{\text{base}} \times PF \times Ql_{\text{old}})^{1/2}$$

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

$$Qg_{\text{target}} = Qo_{\text{base}} \times \text{Base_GOR}$$

for the GOR penalty,

$$Qw_{\text{target}} = Qo_{\text{base}} \times \text{Base_WOR}$$

for the WOR penalty,

$$Qw_{\text{target}} = Qg_{\text{base}} \times \text{Base_WGR}$$

for the WGR penalty or

$$Qg_{\text{target}} = Ql_{\text{base}} \times \text{Base_GLR}$$

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
```

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 type value (action)

where:

type for producers is one of:

*GOR	
*SOR	
*GSOR	
*WCUT	
*WGR	
*BACKFLOW	
*WHYSTAB	
*RATSTAB	
*MIN	*STO
*MIN	*STG
*MAX	*STG
*MAX	*STW
*MAX	*STS
*MAX	*STL
*MAX	*WHP
*MAX	*BHP

type for injectors is one of:

*BACKFLOW	
*WHYSTAB	
*RATSTAB	
*MIN	*STG
*MIN	*STW
*MIN	*STS
*MIN	*WHP
*MIN	*BHP

action for producers is one of:

*STOP	
*SHUTIN (*REPEAT)	
*AUTOWELL <i>frequency</i>	
*RECOMPLETE (*UP *DOWN *UP_OR_DOWN)	

action for producers is one of:

- *SHUTLAYER
- *SHUTLAYER1
- *AUTOLAYER *frequency*
- *AUTOLAYER1 *frequency*
- *WORKOVER (*pi_factor ntimes wo_action*)

action for injectors 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 | cm^3/cm^3) monitor for a producer.

*SOR

This subkeyword identifies a maximum solvent-oil ratio (m^3/m^3 | scf/stb | cm^3/cm^3) monitor for a producer.

*GSOR

This subkeyword identifies a maximum (gas+solvent)/oil ratio (m^3/m^3 | scf/stb | cm^3/cm^3) monitor for a producer.

*WCUT

This subkeyword identifies a maximum water-cut (fraction) monitor for a producer.

*WGR

This subkeyword identifies a maximum water-gas ratio (m^3/m^3 | stb/scf | cm^3/cm^3) 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 versus 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, or WHP constraint is to be monitored as a minimum.

***MAX**

This subkeyword indicates that a rate, BHP, or WHP constraint is to be monitored as a maximum.

***STO**

This subkeyword identifies a surface oil rate (m^3/day | stb/day | cm^3/min) constraint for a producer.

***STG**

This subkeyword identifies a surface gas rate (m^3/day | scf/day | cm^3/min) constraint.

*STW	This subkeyword identifies a surface water rate (m^3/day stb/day cm^3/min) constraint.
*STS	This subkeyword identifies a surface solvent rate (m^3/day scf/day cm^3/min) constraint.
*STL	This subkeyword identifies a total surface liquid (oil + water) rate (m^3/day stb/day cm^3/min) constraint for a producer.
*WHP	This subkeyword identifies a maximum (producers) or minimum (injectors) well head pressure (psi kPa kPa kg/cm^2) monitor. It can only be used for wells for which a method for computing WHP has been introduced with the *PWB or *IWELLBORE keywords.
*BHP	This subkeyword identifies a maximum (producers) or minimum (injectors) bottom-hole pressure (psi kPa kPa kg/cm^2) monitor.
<i>value</i>	A real number specifying the monitored value.
*STOP	This subkeyword specifies that the simulation is to 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 constraint ceases to be violated. The *AUTOWELL action is valid for the *MIN *STO, *MIN *STG, *GOR, *SOR, *WCUT, *GSOR, *WGR, *MAX *WHP, *MAX *BHP, *WHYSTAB and *RATSTAB monitors for producers only. *AUTOWELL can reopen explicitly shut in wells. To avoid this, redefine the wells monitors to exclude the *AUTOWELL condition, the well can be then shut using the *SHUTIN keyword without fear of reopening.

frequency

This is the frequency in days at which the checking which is part of the *AUTOWELL or *AUTOLAYER 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 or *PERFV 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, MAX STW, MAX STL, and WGR violations, and downward from the most offending layer for GOR, SOR, GSOR, MAX STG, MAX STS, 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, 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 only to producers -- a backflowing injector cannot have the *AUTOLAYER action specified.

*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 and *MIN *STG monitored constraints.

pi_factor

A real number specifying the improvement in all the layer productivities for the well:

$$\text{New_PI} = \text{Old_PI} \times \text{pi_factor}.$$

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.

DEFUALTS:

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 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, *SOR, *GSOR, and *WCUT are used only for producing wells, while the *MIN *BHP monitor can be used only for injectors.

*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 m³/day, with a minimum bottom-hole pressure at 500 kPa and with monitoring constraints of GOR at 25000 m³/m³, 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 m³/m³	Field scf/stb	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+20	5.615E+20	1.0E+20

The acceptable range of values for solvent-oil ratios is:

	SI m³/m³	Field scf/stb	Lab cm³/cm³
min	0.0	0.0	0.0
max	1.0E+10	56,165	1.0E+10

The acceptable range of values for water-gas ratios is:

	SI m³/m³	Field stb/scf
min	0.0	0.0
max	1.0E+20	1.781E+19

Alter Primary Well Operating Constraint Value (Optional)

*ALTER

PURPOSE:

*ALTER allows modification of only the primary operating constraint value 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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this alteration of the primary operating constraint applies. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. These names or numbers must be on the same line as the *ALTER keyword.

values

One number for each well identified by *well_list* (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 *ALTER 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 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.

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
*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
```

Reset Injection Composition (Optional)

*INCOMPWL

PURPOSE:

*INCOMPWL sets the global compositions of injected lift-gas for producers or injected streams (gas or solvent) for injectors used in the wellbore calculations.

FORMAT:

*INCOMPWL *well_list*
 values

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword. See '**Using Wildcards in Well Lists**' in the Tutorial Section for more information about using wildcarding. The wells must have the same well type.

values

Global compositions in mole fractions of the injected streams (gas or solvent) or lift-gas in the order defined in the Explanation section of keyword

*INCOMP. *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.

DEFAULTS:

Optional keyword. Default is to use the global mole fraction of injected streams specified by *INCOMP *GAS or *SOLVENT for injectors, and the global mole fraction of produced gas for producers on gas lift specified by *PWEILBORE *MODEL.

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 *GAS or *SOLVENT also applies to *INCOMPWL.

EXPLANATION:

This keyword allows the user to reset the mole fractions of injected streams for a gas or solvent injector or injected lift-gas for producers at any well change time without having to redefine the whole well via *INJECTOR *INCOMP or *PWEILBORE *MODEL. The keyword has no effect on listed wells that are not a producer nor a gas or solvent injector. *INCOMPWL does not define/change the injectors' type as opposed to *INCOMP.

Example:

```
*DATE 2000 01 01
    *INJECTOR 'Injector'
    *INCOMP *SOLVENT *GLOBAL
        0.77  0.0  0.2  0.0  0.0  0.0  0.0  0.03  0.0
    .....
*PRODUCER 'Producer'
.....
*GLIFT *OPT 'Producer'
*INCOMPWL 'Producer'    **set lift-gas global composition
    0.85  0.0  0.1  0.0  0.0  0.0  0.0  0.05  0.0
*DATE 2000 02 31
    *INCOMPWL 'Injector'    **reset solvent global composition
    0.85  0.0  0.1  0.0  0.0  0.0  0.0  0.05  0.0
```

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. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. 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
```

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      type    well_list  
            value_list
```

DEFINITIONS:

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 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. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding.

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.

EXPLANATION:

This optional keyword is used to alter the target constraint for wells without having to redefine all of the additional operating constraints. It is an effective method of altering targets when performing a history match.

*TARGET also changes a target constraint with a new input value if the well is operating with a different target or if the well has been shut in by a previous action or if the well has been initially defined as a shut in well.

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 for 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 | *STS | *OFF
```

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this special mode applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. 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, *STS 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 or *STS, the wells will be put on the corresponding surface oil, gas, water or solvent 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 *con_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* is present 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, *STS) are specified through keyword *OPERATE as regular operating constraints. The fourth-phase flow rate can be entered by *STS after *OPERATE for solvent (m^3/day | scf/day | cm^3/min); or interpreted as light oil API tracking (m^3/day | stb/day | cm^3/min), or polymer (m^3/day | stb/day | cm^3/min) when the proper option is turned on. 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 1000000.0
*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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this alteration of flow rate values applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

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, *STS), 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
```

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 type well_list values  
type = *OIL | *WAT | *LIQ | *GAS | *TOT | *MULT | *MULTO
```

DEFINITIONS:

*OIL

This subkeyword identifies that the PI of oil will be set to the values specified. Units are ($\text{m}^3/\text{day}/\text{kPa}$ | $\text{stb}/\text{day}/\text{psi}$ | $\text{cm}^3/\text{min}/\text{kPa}$)

*WAT

This subkeyword identifies that the PI of water will be set to the values specified. Units are ($\text{m}^3/\text{day}/\text{kPa}$ | $\text{stb}/\text{day}/\text{psi}$ | $\text{cm}^3/\text{min}/\text{kPa}$)

*GAS

This subkeyword identifies that the PI of gas will be set to the values specified. Units are ($\text{m}^3/\text{day}/\text{kPa}$ | $\text{scf}/\text{day}/\text{psi}$ | $\text{cm}^3/\text{min}/\text{kPa}$)

*LIQ

This subkeyword identifies that the PI of the sum of oil and water will be set to the values specified. Units are ($\text{m}^3/\text{day}/\text{kPa}$ | $\text{stb}/\text{day}/\text{psi}$ | $\text{cm}^3/\text{min}/\text{kPa}$)

*TOT

This subkeyword identifies that the PI of the sum of oil, water and gas will be set to the values specified. Units are ($\text{m}^3/\text{day}/\text{kPa}$ | $\text{scf}/\text{day}/\text{psi}$ | $\text{cm}^3/\text{min}/\text{kPa}$). 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_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this alteration of well productivity applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. *well_list* 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_list* (or a single value for all wells) specifying the new value of the well productivity. 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 keywords. 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

A warning is issued when this occurs.

If 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, oil, 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, 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. The PI's which are printed in the detailed well summary using *OUTPRN *WELL *ALL are calculated in reservoir conditions and use each phases formation volume factor to bring the result to surface conditions. Gas which comes out of solution is ignored.

Example 1:

```
*INUNIT  *FIELD
*WELL 1 'PRODUCER'
*WELL 2 'INJECTOR'
**          rw      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'
**          rw      geofac    wfrac   skin
*GEOMETRY *K  0.375  0.2488   1.0     -1.0
  7.0
*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
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

```

User-Specified Reference Depth for Well BHP (Optional)

*BHPDEPTH

PURPOSE:

*BHPDEPTH allows the user to specify a depth to which a well's bottom hole 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* | *DEFAULT

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this specification of reference depth applies. See ‘[Using Wildcards in Well Lists](#)’ in the Tutorial Section for more information about using wildcarding.

depth_values

A list consisting of non-negative real numbers (m | ft). 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

Restores the well to the default state in which the bottom hole pressure is the wellbore pressure in the reference completion.

DEFAULTS:

Optional keyword. If BHPDEPTH does not appear in the data set, all wells have bottom hole 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 on 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

$$delp = g \cdot rho \cdot (ref_depth - completion_depth)$$

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 bottom hole 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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this specification of pressure gradient applies. See '**Using Wildcards in Well Lists**' in the Tutorial Section for more information about using wildcarding.

gradient_values

A list consisting of non-negative real numbers (kPa/m | psi/ft | kPa/cm | kg/cm²/m). 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 *depth_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

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

$$delp = p_grad \cdot (ref_depth - completion_depth)$$

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.

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} hgrad | *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*}

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 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).

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.

hgrad

Non-negative real value numbers (kPa/m | psi/ft | kPa/cm | kg/cm²/m) 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.

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

$$Delp = (P_{next} - P_{named}) = hgrad \cdot (depth_{next} - depth_{named})$$

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.

EXAMPLES:

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.

Alter Polymer Concentration (Optional)

*ALTERCP

PURPOSE:

*ALTERCP allows modification of the polymer concentration given originally with *INCOMP *WATER (*cpvalue*).

FORMAT:

```
*ALTERCP      well_list  
               cpvalues
```

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this alteration of polymer concentration applies. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. These well names or numbers must be on the same line as the *ALTERCP keyword. If more than one line is required to specify all the wells, then the *ALTERCP keyword must be repeated.

cpvalues

One number for each well entered in the list of well names or well numbers (or one value for all wells), specifying the new value of the polymer concentration (kg/m³ | lb/stb | g/cm³). Values must appear on one or more new lines immediately following the *ALTERCP line.

If only one value of *cpvalues* is input all wells defined in the well list will be set to that single value.

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

*ALTERCP must be located in the Well and Recurrent Data keyword group, and may appear anywhere in this keyword group following the well definitions.

EXPLANATION:

This optional keyword is used to alter the polymer concentration for an injection well or a group of injection wells. *ALTERCP requires the same format as *ALTER.

Examples:

```
*MODEL *POLY  
:  
*DATE  
*WELL  1  
:
```

```
*INJECTOR 1
*INCOMP *WATER 0.7
:
*DATE 1989 09 01
*ALTERCP 1
    0.0
```

Alter Seawater Injection Volume Fraction (Optional) *ALTERFS

PURPOSE:

*ALTERFS allows modification of the seawater injection volume fractions given originally with *INCOMP *SEAWATER (*fsvalue*).

FORMAT:

```
*ALTERFS      well_list  
           fsvalues
```

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this alteration of seawater fraction applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. These well names or numbers must be on the same line as the *ALTERFS keyword. If more than one line is required to specify all the wells, then the *ALTERFS keyword must be repeated.

fsvalues

One number for each well entered in the list of well names or well numbers (or one value for all wells), specifying the new value of the seawater volume fraction. Values must appear on one or more new lines immediately following the *ALTERFS line.

If only one value of *fsvalues* is input all wells defined in the well list will be set to that single value.

DEFAULTS:

Optional keyword. No defaults.

CONDITIONS:

*ALTERFS must be located in the Well and Recurrent Data keyword group, and may appear anywhere in this keyword group following the well definitions.

EXPLANATION:

This optional keyword is used to alter the seawater injection volume fraction for an injection well or a group of injection wells. *ALTERFS requires the same format as *ALTER.

Examples:

```
*MODEL *BLACKOIL_SEAWATER  
:  
*DATE 1979 01 01  
*WELL 1  
:
```

```
*INJECTOR 1
*INCOMP *SEAWATER 0.95
:
*DATE 1989 09 01
*ALTERFS 1
    1.0
```

Well Scale Table Assignment (Optional)

*SCLTBL-WELL

PURPOSE:

*SCLTBL-WELL assigns deposition and damage tables (defined in the Rock-Fluid data section to wells.

FORMAT:

*SCLTBL-WELL *well_list*
 Deposition Table Set Number *Damage Table Set Number*

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which the scale table definition applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. These names or numbers must be on the same line as the *SCLTBL-WELL keyword. If more than one line is required for the well list, then the *SCLTBL-WELL keyword must be repeated.

Deposition Table Set Number

The set number of the deposition table used for this well. New perforations defined after this keyword is set will also use the defined deposition table number.

Damage Table Set Number

The set number of the damage table used for this well. New perforations defined after this keyword is encountered will also use the defined damage table number for the new perforations.

DEFAULTS:

Conditional keyword. There are no defaults. If not defined the well will not experience scaling damage.

CONDITIONS:

This keyword must be in the Recurrent Data keyword group.

EXPLANATION:

The deposition and damage tables defined in the rock-fluid data section using *SCLDPS and *SCLDMG are applied to the wells listed on the *SCLTBL-WELL keyword. Each combination of deposition and damage table entry would require a separate *SCLTBL-WELL keyword. After the reading of *SCLTBL-WELL, if the well is reperforated, new perforations would also use the deposition and damage tables defined on *SCLTBL-WELL.

*SCLTBL-WELL deposition and damage table values can be overwritten by either a subsequent *SCLTBL-LAYER keyword (for specified layers) or by a subsequent *SCLTBL-WELL keyword.

Layer Scale Table Assignment (Optional)

*SCLTBL-LAYER

PURPOSE:

*SCLTBL-LAYER assigns deposition and damage tables (defined in the Rock-Fluid data section to multiple specified layers of a single well.

FORMAT:

```
*SCLTBL-LAYER  wn
  {location}    Deposition Table Set Number    Damage Table Set Number
    :           :           :
```

DEFINITIONS:

wn

A single well name (in quotes) or a single well number.

{*location*}

i_f *j_f* *k_f* (/ ...(/ *i_m* *j_m* *k_m*)...)

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 the well to be assigned deposition and damage tables. Any layer named under *SCLTBL-LAYER must already have been defined for the well using a *PERF or *PERFV statement.

Optional (*i_m, j_m, k_m*) 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.

Deposition Table Set Number

The set number of the deposition table used for this layer.

Damage Table Set Number

The set number of the damage table used for this layer.

DEFAULTS:

Conditional keyword. There are no defaults. Any number of layers can be entered per instance of this keyword.

CONDITIONS:

This keyword must be in the Recurrent Data keyword group. It is not necessary to enter all layers.

EXPLANATION:

The deposition and damage tables defined in the rock-fluid data section using *SCLDPS and *SCLDMG are applied to the layers of the well listed on the *SCLTBL-LAYER keyword.

*SCLtbl-LAYER need not be specified for each layer. Unspecified layers will either not experience scale damage (if *SCLtbl-WELL was not specified) or will use the deposition and damage tables specified on a previous *SCLtbl-WELL keyword.

*SCLtbl-LAYER deposition and damage table values can be overwritten by either a subsequent *SCLtbl-LAYER keyword (for specified layers) or by a subsequent *SCLtbl-WELL keyword.

Well Scale Removal (Optional)

***SCLRMV-WELL**

PURPOSE:

*SCLRMV-WELL removes a fraction of the accumulated deposited scale from a well.

FORMAT:

*SCLRMV-WELL *well_list*
 value

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which the scale removal applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. These names or numbers must be on the same line as the *SCLRMV-WELL keyword. If more than one line is required for the well list, then the *SCLRMV-WELL keyword must be repeated.

value

The fraction of cumulative scale deposited which is removed A single value is entered for all listed wells. A value of 1.0 implies complete removal of accumulated scale.

DEFAULTS:

Conditional keyword. If value is omitted a value of 1.0 (i.e. complete scale removal is assumed).

CONDITIONS:

This keyword must be in the Recurrent Data keyword group. Value must lie between 0.00 and 1.00. There is only one value of the fraction removed entered per instance of the SCLRMV-WELL keyword

EXPLANATION:

This keyword allows the user to completely or partially remove the scale built up on a well over time. This will completely or partially reverse the damage to the wells productivity.

Layer Scale Removal (Optional)

*SCLRMV-LAYER

PURPOSE:

*SCLRMV-LAYER removes a fraction of the accumulated deposited scale from multiple specified perforations of a single well.

FORMAT:

```
*SCLRMV-LAYER  wn
    {location}      value
    :           :
```

DEFINITIONS:

wn

A single well name (in quotes) or a single well number.

{*location*}

i_f *j_f* *k_f* (/ ...(/ *i_m* *j_m* *k_m* ...))

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 the well to remove a fraction of the accumulated deposited scale. Any layer named under * SCLRMV-LAYER must already have been defined for the well using a *PERF or *PERFV statement.

Optional (*i_m, j_m, k_m*) 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.

value

The fraction of cumulative scale deposited which is removed. A value of 1.0 implies complete removal of accumulated scale.

DEFAULTS:

Conditional keyword. If value is omitted, a value of 1.0 (i.e. complete scale removal) is assumed.

CONDITIONS:

This keyword must be in the Recurrent Data keyword group. Values must lie between 0.00 and 1.00. Any number of layers can be entered per instance of this keyword.

EXPLANATION:

This keyword allows the user to completely or partially remove the scale built up on perforations over time. This will completely or partially reverse the damage to the perforation's productivity.

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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this specification of *WORKOVER parameters applies.

See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. 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.

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 *STG 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 STG 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 *STG 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 beset 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

One or more well numbers or quoted well names assigned with the *WELL keyword to which the this alteration of initialization frequency applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

*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 the 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. The wells in the list included on the *MRC-RESET line must have already been defined using the *WELL keyword.

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 bottom hole 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 'SPECIAL-PRODUCER'  
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 well numbers or quoted well names assigned with the *WELL keyword to which this specification applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. These names or numbers must be on the same line as the *GLIFT keyword. If more than one line is required for the well list, then the *GLIFT keyword must be repeated.

*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 (m³/day | scf/day | cm³/min | m³/day).

*GLR

When this subkeyword is specified, gas lift injection rates are calculated using produced gas-liquid ratios from the previous timestep (explicitly). When all of *RATE, *GLR and *OPT are missing *RATE is the default. For this option the units of *values* are (m³/m³ | scf/stb | cm³/cm³ | m³/m³).

*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. When all of *RATE, *GLR and *OPT are missing, *RATE is the default. 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_list*. The units of the values are (m³/day | SCF/day | cm³/min | m³/day) when used with *OPT. When no values are entered, very large maxima (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* (or one value for all wells on the 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.

If only one value of *values* is input all wells defined in *well_list* will be set to that single value.

DEFUALTS:

Optional keyword. If the keyword is missing then gas lift rates are zero. The default suboption 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 *PWEELBORE must also be specified for that well. A well must have its type defined as *PRODUCER before it appears in a well list following *GLIFT; otherwise an error message is printed and the simulation is terminated.

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 *PWEELBORE 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 time- step's production data. However, the lift gas rates will not be used for the calculation of production rates until the next timestep.
$$\text{Lift_gas_inj} = \text{GLR_input} \times \text{Total_liquid_est} - \text{GOR_est} \times \text{Oil_rate_est}$$
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 results file (SRF) do not include lift gas.

This option has the following assumptions:

1. The lift gas 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 well numbers or quoted well names assigned with the *WELL keyword to which this specification applies. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. These names or numbers must be on the same line as the *GLCONTROL keyword. If more than one line is required for the well list, then the *GLCONTROL keyword must be repeated.

*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 | kPa | kg/cm²).

*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 | kPa | kg/cm²).

*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* (m³/day | stb/day | cm³/min | m³/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 *PWEELBORE 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 *PWBBORE.

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 *PWBBORE must also be specified for that well. A well can appear in the list following *GLCONTROL only if it has already had its type defined with a *PRODUCER statement; otherwise an error message will be printed 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 with 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 versus 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 | cm^3/cm^3 | m^3/m^3)

consum_rate

The compressor gas consumption rate per unit gas injected. (m^3/m^3 | scf/scf | cm^3/cm^3 | m^3/m^3)

*MAXGAS

This keyword 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 | cm^3/min | m^3/day)

*RECYCLE

When this subkeyword is specified, the total produced gas will be recycled and used for gas lift injection. 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 will increase with time.

<i>'group_name'</i>	Group name from which the gas to be injected is produced.
<i>recyc_frac</i>	Recycle fraction of gas production from a group of wells.
<i>make_up_vol</i>	Make-up volume of gas available for injection from some source outside the field being modelled. (m ³ /day SCF/day cm ³ /min m ³ /day)
<i>max_rate</i>	Compressor maximum gas injection rate. (m ³ /day SCF/day cm ³ /min m ³ /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.
<i>time_interval</i>	The time interval between optimizations specified using the *GLOPT-TIME keyword. The default is 182.5 days (six months). A well is put on gas lift only at the times specified by the *GLOPT-TIME keyword. Thus if a well requires gas lift after 10 days of simulation and the *GLOPT-TIME time interval is 182.5 days, an additional 172.5 must pass before the well is put on gas lift. Use of smaller time intervals are recommended if the wells are to be put onto gas lift more rapidly.

DEFUALTS:

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 0.00028m³/m³. The default value for *consum_rate* is 0.0 SCF/day (m³/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:

$$\text{SUM_over_packets} \left(\frac{d(\text{oil rate})}{d(\text{gas inj rate})} \right)_{\max}$$

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" its 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 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 dir rw geofac wfrac skin  
dir = *I | *J | *K
```

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 well radius (m | ft | cm).

geofac

A real number specifying the geometric factor for the well element. This value is not used when the well effective radius is calculated using the Peaceman formula (see below), 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* = 7.62 cm (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.

*GEOMETRY may be used with *INJECTOR *MOBWEIGHT or with *PRODUCER.

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.

For example, to define two wells with identical well geometry, and a third with a larger radius, the input data would appear as:

```
** 3 wells are defined.  
*WELL 1 'Producer 1' *VERT 12 14  
*WELL 2 'Producer 2' *VERT 15 19  
*WELL 3 'Producer 3' *VERT 21 23  
...  
** The well geometries are input for wells 1,2  
**           rw      geofac wfrac skin  
*GEOMETRY *K 0.375 0.2488 1.0    0.0  
** The well completion is defined for wells 1,2  
*PERFV *GEO 1:2  
** kf      wi      setn  
     2:4    1.      1  
** Well geometries are input for well 3  
**           rw      geofac wfrac skin  
*GEOMETRY *K 0.5 0.2488 1.0    0.0  
** The well completion is defined for well 3  
*PERFV *GEO 3  
** kf      wi      setn  
     2:4    1.      1
```

The variable *rw* is required if it has not been entered elsewhere in the data set.

If *GEOMETRY is used, then *rw* must be equal to the well radius entered with *REFINE *HYBRID. Since the well is assumed to be in the center of the grid block *geofac* is assumed to be 0.50. Hybrid grid is used to model, especially in producers, more accurately, near-well effects, such as breakthrough.

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) (*DIR) (*GEO | *GEOA) (*KH | *KHA) (*RE) (*RW)
(*SKIN) (*TURB | *TURB_GRM | *TURB_FG1 | *TURB_FG2)
(*QUAD | *PSEUDOP)

DEFINITIONS:

{locat.}

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

{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.}

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

(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(_GRM / _FG1 / _FG2) 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 for the layer for each phase, and injection for MOBILITY WEIGHTED injectors 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 | md-cm). 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.

If the well is a MOBILITY UNWEIGHTED INJECTOR, then the equation is:

$$q = wi \cdot (P_{block} - P_{well})$$

with units for wi of (m³/kPa-day | stb/psi-day or scf/psi-day | cm³/kPa-min | m³/kg/cm²-day)

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 | md-cm). *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 | cm). 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 | cm). *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 | *TURB_GRM | *TURB_FG1 | *TURB_FG2 - *di

Optional sub-keyword indicating that turbulent skin effects are to be considered due to high gas velocity near the wellbore which results in a

deviation from Darcy's law. The mechanical skin factor will be modified based on D factors:

$$\text{skin}' = \text{skin} + di \cdot q_i$$

with q_i being the gas rate at the surface condition. Keyword *TURB indicates the D factor is given by user input (di). If *TURB_GRM, *TURB_FG1 or *TURB_FG2 is used, the simulator calculates D factors based on the correlation of Geertsma (1974) or on the two correlations of Fredrick and Graves (1994) respectively. In this case, di serves as a multiplier. See keyword *NONDARCY for a description of these correlations.

If none of these subkeywords are present, turbulent effects will be ignored ($di = 0$). These subkeywords are mutually exclusive and can only be used when the well indices are calculated internally. The units for di are (days/m³ | days/ft³ | min/cm³) for *TURB, but dimensionless for the other options.

*QUAD

Optional sub-keyword indicating that a quadratic inflow relationship is to be used for calculating the gas inflow/outflow from a well. *QUAD does not take any column of the {well index} table. The density of the gas near the wellbore is assumed to be proportional to its pressure yielding:

$$q = wi \cdot \lambda \cdot (P_{block}^2 - P_{well}^2) / 2P_{block}$$

as the inflow relationship. If *QUAD is not present, the standard formulation is assumed. It can only be used when the well indices are calculated internally.

*PSEUDOP

Optional sub-keyword indicating that the variation in both gas density and gas viscosity with pressure is to be taken into account in computing the gas flow rate into or out of the wellbore. *PSEUDOP does not take any column of the {well index} table. The expression used for the gas flow rate when *PSEUDOP is in force is

$$q = wi \cdot K_{rg} \cdot \rho_{gstd} \cdot (E_g/\mu)_{ave} \cdot (P_{block} - P_{well}),$$

with the average value for the quotient of gas expansion factor (E_g) and gas viscosity (μ) computed as

$$(E_g/\mu)_{ave} = 0.5 [(E_g/\mu)_{well} + (E_g/\mu)_{block}],$$

with the gas expansion factor and gas viscosity evaluated directly from input PVT table at both pressures. *PSEUDOP can be used only when the well indices are calculated internally. *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 K_{rg} 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 K_{row} 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 K_{rog} 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

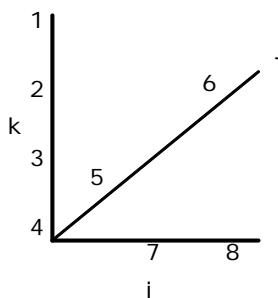
This subkeyword specifies that the well layer is geometrical node point for the purpose of defining the well trajectory. The layer will remain plugged unless overridden by a new *PERF / *PERFV card selecting the layer status at that location as *OPEN or *AUTO.

*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 *v1 v2*

 where

 type = *RANGE1 | *MEAN1 | *MEDIAN1 |
 *RANGE2 | *MEAN2 | *MEDIAN2 | *OFF

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. *WIRANGE will apply to ALL wells if *well_list* is not input.

*RANGE1, *RANGE2

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, MEAN2

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, MEDIAN2

Same as the above for *MEAN1 or MEAN2 except that wi_m is the median of the wi for all the perforations.

*OFF

Turn off the well indices range regulation. *v1* and *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. v1 and v2 are non-negative numbers and v2 cannot be smaller than v1. They must be present unless the subkeyword is *OFF.

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 $wi = 0$.

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. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding. *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 | cm). 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.

DEFUALTS:

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.*}

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

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_m, j_m, k_m) 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 | cm). See explanation.

$x_2 \ y_2 \ z_2$

Coordinates of the “exit point” of the wellbore from the perforated grid block. (m | ft | cm). See explanation.

h

Perforated length of the deviated wellbore within the named grid block (m | ft | cm). 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 reimposed 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 ($w_i > 0$) 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

$$wi = 2\pi \cdot wfrac \cdot K \cdot h \cdot ff / [\ln(re/rw) + skin]$$

The well angular fraction *wfrac*, the completion factor *ff*, the well radius *rw*, and the 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 line, 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

$$re(D) = geofac \cdot [V / (\pi \cdot xh(D) \cdot wfrac)]^{1/2}$$

where $V = xh(I) \cdot xh(J) \cdot xh(K)$ with *xh(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 \mathbf{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 \mathbf{i}, \mathbf{j} , and \mathbf{k} necessarily mutually orthogonal. The following two methods are supported.

Directional Weighting

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)*:

$$re(\mathbf{u}) = [u_i^2 re^2(I) + u_j^2 re^2(J) + u_k^2 re^2(K)]^{1/2}$$

Directional Weighting is the recommended and default method starting with version 2011.10.

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

$$re(\mathbf{u}) = [wt_i re^2(I) + wt_j re^2(J) + wt_k re^2(K)]^{1/2}$$

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.

Average Permeability

The completion planar averaged permeability K is computed similarly except that $re(I)$, $re(J)$, and $re(K)$ are replaced with

$$\begin{aligned} K(I) &= (K_j K_k)^{1/2} \\ K(J) &= (K_k K_i)^{1/2} \\ K(K) &= (K_i K_j)^{1/2} \end{aligned}$$

EXAMPLES:

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.*}

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

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_m, j_m, k_m) 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).

EXAMPLES:

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.

Data for Hydraulic Fracture Proxy (HFP) Model (Conditional)

***LAYERHFP**

PURPOSE:

*LAYERHFP allows the user to supply geometric and non Darcy information specifying productivity indices which act as proxies for vertical fractures (assuming pseudo radial flow) on a layer by layer basis.

FORMAT:

```
*LAYERHFP 'wname'  
{locat.}      {HFP Layer Information}  
  :           :
```

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.}

$i_f \ j_f \ k_f$ (/ ...(/ $i_m \ j_m \ k_m$)...)

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 fractured layers. It is valid to name some of the well's layers and not others in the *LAYERHFP lines. Layers not mentioned not treated as proxies for vertical fractures. Any layer named under *LAYERHFP must already have been defined for the well using a *PERF or *PERFV statement.

Optional (i_m, j_m, k_m) 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.

{HFP Layer Information}

$hf_hlen \ hf_wid \ hf_perm \ hf_bet_a \ hf_bet_b \ hf_ccf$ | *DEFAULT

hf_hlen

Fracture half length (m | ft).

hf_wid

Fracture width (m | ft).

hf_perm

Fracture Permeability (mD).

hf_bet_a

Constant term in the non Darcy β (Beta) factor correlation. See below.

hf_bet_b

Permeability term in the non Darcy β (Beta) factor correlation. See below.

hf_bet_a and *hf_bet_b* are used to determine the non Darcy β term for the fracture.

$$\beta_{\text{frac}}(p) = hf_bet_a \cdot (K_{\text{rp}} \cdot hf_perm)^{hf_bet_b} \quad p = \text{oil, water, gas, or solvent},$$

hf_perm is the fracture permeability in mD and *hf_bet_a* must be entered so as to yield $\beta_{\text{frac}}(p)$ in units 1/ft. K_{rp} is the relative permeability of the phase whose $\beta_{\text{frac}}(p)$ is being calculated. Regardless of the unit system employed in the simulator, β_{frac} has units of 1/ft.

hf_ccf

Fracture conductivity correction factor for non Darcy flow. See explanation, normally set to 0.50.

***DEFAULT**

Subkeyword indicating that the layer is henceforth to be treated as an unfractured layer.

DEFAULTS:

Layers named under *LAYERHFP with geometric information and non Darcy information (as opposed to the *DEFAULT subkeyword) are flagged as hydraulically fractured within the simulator; *DEFAULT is the default (unfractured) status assigned at the beginning of the run. Unfractured status can be reimposed through the *DEFAULT subkeyword. All 6 HFP entries must be defined if the proxy model is used (*hf_hlen*, *hf_wid*, *hf_perm*, *hf_bet_a*, *hf_bet_b*, *hf_ccf*).

CONDITIONS:

The named layers must all have been previously created for well ‘wname’ with *PERF or *PERFV lines using the *GEO *KH *GEOA or *KHA option for computation of well indices (cannot be used with the *WI option). If geometric and non Darcy information is supplied, all of the input (*hf_hlen*, *hf_wid*, *hf_perm*, *hf_bet_a*, *hf_bet_b*, and *hf_ccf*) must be specified or an error is generated. Not all of a well’s layers need to be named under *LAYERHFP; those omitted are treated as unfractured according to the original *PERF specification.

EXPLANATION:

This keyword specifies input of geometric and non Darcy parameters which allows the well productivity index (PI) to be used as a proxy for a vertical hydraulic fracture, assuming the flow pattern is pseudo radial. The work is based on the work of Pratt, Cinco-lay and Samaniego. Later Guppy and Gidley developed a similar non Darcy extension. This option applies these concepts to dynamically calculate well productivities within a reservoir simulator.

Early work in well test analysis of fractured wells has shown that for the pseudo radial flow period an infinite conductivity fractured well behaves like an unfractured well with an effective wellbore radius of $\frac{1}{2}$ the fracture half length.

Low and finite conductivity wells can be represented through the use of an effective wellbore radius as well. In this case, the effective radius is a function of dimensionless fracture conductivity ($K_f B_f$)D and fracture $\frac{1}{2}$ length. This analysis applies to Darcy flow in fractures only.

The extension which allows the application of these ideas to non Darcy flow is to calculate a resistance factor due to non Darcy flow in the fracture which lowers the fracture conductivity. The resistance factor, which is defined as $1/(1+Forch_{fr})$, is a function of dynamically changing reservoir properties and flow in the fracture, all of which vary from timestep to timestep and so must be reevaluated at every timestep.

For each phase (p), at each timestep, for each layer, we calculate the Fracture Forchheimer Number ($Forch_{fr}$) as:

$$Forch_{fr} = \frac{hf_ccf \times \beta_{frac}(p) \times \text{Density}(p) \times K_{rp} \times hf_perm \times Q(p)}{\text{Viscosity}(p) \times hf_wid \times (\text{layer vertical height})}$$

Where $\beta_{frac}(p) = hf_bet_a \times (K_{rp} \times hf_perm)^{hf_bet_b}$

If hf_ccf is 0.5, this assumes the non Darcy effects in each half of the fracture are based on $\frac{1}{2}$ of the layers total flow. $Q(p)$ is the production rate of phase p. K_{rp} is the relative permeability of phase p.

The Darcy dimensionless fracture conductivity is calculated as:

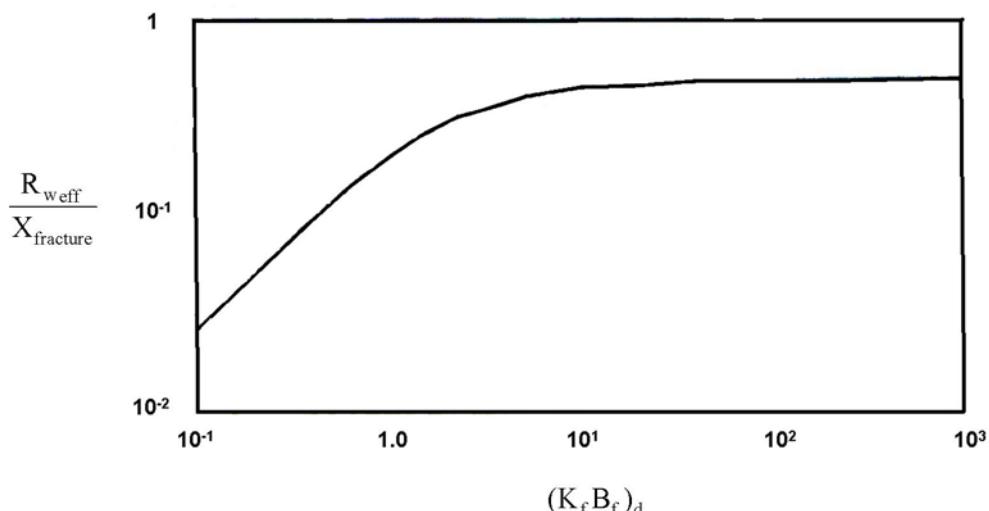
$$(K_f B_f)_D = \frac{hf_perm \times hf_wid}{(\text{Formation perm}) \times (\text{fracture } 1/2 \text{ length})}$$

The dimensionless conductivity is reduced by $1/(1+Forch_{fr})$ and this reduced dimensionless conductivity is used to calculate (using the function graphed below) an effective well radius from which a fracture proxy well PI is calculated.

hf_ccf is the fracture conductivity correction factor which is used to relate the flow of the layer to the flow in each wing of the fracture. As the Forchheimer number in each wing of the well is normally based on no more than $\frac{1}{2}$ of the well layer rate, we assume the hf_ccf should normally be no greater than 0.5. This implies that the maximum Forchheimer number is based on no more than $\frac{1}{2}$ the well layer rate.

Normally hf_ccf should be set to 0.5, hf_ccf values smaller or larger than 0.5 can be used in order to match observed well pressures. For example; if a significant amount of production enters the fracture right at the well, the Forchheimer number in the fracture would be less than if hf_ccf was assumed to be 0.5. In this case lower values of hf_ccf might be used to account for the production which effectively bypasses the fracture.

$hf_ccf=0.0$ can be used to model Darcy flow in the fracture (completely ignoring non Darcy effects), by making the Forchheimer number equal to 0.0. Darcy flow tends to estimate very large effective well radii; the model will limit this to be no more than 90% of the well block's effective radius (R_e). Using this model to model Darcy flow is only useful in illustrating the large influence non Darcy effects have on fracture productivity.



Effective Wellbore Radius versus Dimensionless Fracture Conductivity for a Vertical Fracture

Examples:

Example 1:

```
*LAYERHFP 'WELL-NNE17'
**      frac    frac    frac  Beta   Beta   conductivity
**      half     width   perm  A-term B-term correction
**      length
65 23 5 150.0  0.00833 50000 27.3e9 -1.1045 0.5
65 23 6 130.0  0.00800 45000 27.3e9 -1.1045 0.5
```

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:

```
*LAYERHFP 'WELL-SINGLE-REF'
**      frac    frac    frac  Beta   Beta   conductivity
**      half     width   perm  A-term B-term correction
**      length
16 48 11 / 2 2 1 80.0  0.00833 100000 27.3e9 -1.1045 0.5
```

In this example, the already defined completion of well ‘WELL-SINGLE-REF’ in the refined grid block 16 48 11 / 2 2 1 is flagged as being a proxy for a hydraulic fracture.

Example 3:

```
*LAYERHFP 'WELL-SINGLE-REF'
** The layer is unfractured (may have previously
** been a proxy for a fracture)
16 48 11 / 2 2 1 *DEFAULT
```

In this example, the already defined completion of well ‘WELL-SINGLE-REF’ in the refined grid block 16 48 11 / 2 2 1 is flagged as *not* being a proxy for a hydraulic fracture.

In the examples above the Beta A-term (`hf_bet_a`) = 27.3e9 and the B-term (`hf_bet_b`) = -1.1045 were taken from an example in the textbook by Dake (Fundamentals of Reservoir Engineering).

The correlation for β developed by Cooke, C.E. "Conductivity of Fracture Proppants in Multiple Layers" J. Pet. Tech, (Sept 1973) pp1101-1107 is relevant for fractured systems and is presented below (β_{frac} in 1/ft, K_f in mD).

Proppant Sand Mesh Size	<code>hf_bet_a</code>	<code>hf_bet_b</code>
8-12	538.108E9	-1.24
10-20	850.525E9	-1.34
20-40	3411.752E9	-1.54
40-60	2143.503E9	-1.60

Useful β conversion factors are:

$$1 \text{ atm}\cdot\text{sec}^2/\text{gm} = 1.01325\text{E}6 \text{ cm}^{-1}, 1 \text{ ft} = 30.48 \text{ cm},$$

$$1 \text{ mD} = 0.9869233\text{E}-11 \text{ cm}^2$$

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.}
:           :
```

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.*}

A set of one or more of the subkeywords *SETN, *SWCON, *SWCRIT, *SORW, *SOIRW, *SGCON, *SGCRIT, *SORG and *SLCON, 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.

*SWCON

Introduces the real number specifying the connate water saturation for the completion, which scales the *SWT relative permeability table. *SWCON is the smallest water saturation possible in the *SWT table. One minus *SWCON is the maximum oil saturation possible in the *SWT table. See DEFAULTS if the keyword is absent.

***SWCRIT**

Introduces the real number specifying the critical water saturation for the completion, which scales the *SWT relative permeability curve. *SWCRIT is the water saturation at which water first becomes mobile in the two phase oil-water table. *SWCRIT should be equal to or greater than *SWCON. See DEFAULTS if the keyword is absent.

***SOIRW**

Introduces the real number specifying the irreducible oil saturation for the completion, which scales the *SWT relative permeability curve. *SOIRW is the smallest oil saturation in the two phase oil-water table, one minus *SOIRW is the maximum water saturation in the table. See DEFAULTS if the keyword is absent.

***SORW**

Introduces the real number specifying the residual oil saturation for the completion, which scales the *SWT relative permeability curve. *SORW is the oil saturation in the two phase oil-water table at which oil first becomes mobile. *SORW should be equal to or greater than *SOIRW. See DEFAULTS if the keyword is absent.

***SORMAX**

Introduces the real number specifying the maximum residual oil saturation for the completion, which scales the Sormax value of the rock type of the completion. It is only effective when the Krow hysteresis or trapped oil hysteresis is modeled for the rock type of the completion. Sormax is the oil saturation in the two phase oil-water system at which oil first becomes immobile during the boundary/primary imbibition process (the imbibition starting from oil saturation equals to 1-Swcon). See DEFAULTS if the keyword is absent.

***SGCON**

Introduces the real number specifying the connate gas saturation for the completion, which scales the *SGT or *SLT relative permeability curve. *SGCON is the smallest gas saturation in the two phase gas-liquid relative permeability table. See DEFAULTS if the keyword is absent.

***SGCRIT**

Introduces the real number specifying the critical gas saturation for the completion, which scales the *SGT or *SLT relative permeability curve. *SGCRIT is the gas saturation in the two phase gas-liquid relative permeability table at which gas first becomes mobile. *SGCRIT should be equal to or greater than *SGCON. See DEFAULTS if the keyword is absent.

*SLCON	Introduces the real number specifying the connate liquid saturation for the completion, which scales the *SGT or *SLT relative permeability curve. *SLCON is the smallest liquid saturation entry in the two phase gas-liquid relative permeability table. See DEFAULTS if the keyword is absent.
	NOTE: *SLCON can only be input if the *SCALING-STARS option is not used
*SOIRG	Introduces the real number specifying the irreducible oil saturation for the completion, which scales the *SGT or *SLT relative permeability curve. See DEFAULTS if the keyword is absent.
	NOTE: *SOIRG can only be input if the *SCALING-STARS option is used.
*SORG	Introduces the real number specifying the residual oil saturation for the completion, which scales the *SGT or *SLT relative permeability curve. *SORG is the oil saturation in the two phase gas-liquid relative permeability table at which oil becomes mobile. The definition of *SORG changes depending on whether the *NOSWC option is active. If *NOSWC is not active, the residual liquid saturation equals *SORG plus *SWCON, if *NOSWC is active, the residual liquid saturation equals *SORG. See DEFAULTS if the keyword is absent.

DEFAULTS:

For the saturation endpoints which are not explicitly defined on the current *KRPERF keyword, the following algorithm is used to obtain their values.

If SETN is read in on the *KRPERF keyword (or defined on the *PERF line) for the layer in question, unassigned saturation endpoints are obtained from the original table value of the rock type assigned in *SETN. Current grid block saturation endpoints are ignored.

If SETN has not been read in on the current KRPERF keyword (or defined on the *PERF line) for the layer in question, unassigned saturation endpoints are obtained from the current grid block saturation endpoints.

*SETN, *SWCON, *SGCRIT, *SORW, and *SORG 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 *SWCRIT , *SOIRW, *SORMAX, *SGCON and *SLCON end points can only be set with the *KRPERF keyword.

When values of *SETN, *SWCON, *SGCRIT, *SORG, and *SORG 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 *SWCON, *SWCRIT, *SORW, *SOIRW, *SGCON, *SGCRIT, *SLCON, or *SORG.

Please note the *SCALING-OLD option is no longer supported.

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. Residual 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 residual oil saturation. Irreducible oil is calculated from *SLCON and *SWCON. If *NOSWC is not active, Soirg equals *SLCON minus *SWCON. If *NOSWC is active, Soirg equals *SLCON.
4. Critical gas saturation is less than connate gas saturation. Critical gas saturation is reset to connate gas saturation.
5. If when *NOSWC is not active, connate liquid saturation in the gas-liquid table is less than connate water saturation, connate liquid saturation is reset to connate water 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.

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, *SWCON, *SORW, *SORG, and *SGCRIT may be entered either in the *PERF.. line or using the *KRPERF keyword; *SWCRIT, *SOIRW, *SGCON and *SLCON 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 come up to the perforation (e.g. $S_w = 0.67$) 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.

Examples of Usage

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 *SWCRIT 0.22 *SOIRW 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 *SWCRIT 0.15 *SGCRIT 0.10  
*SORW 0.12 *SORG 0.10 *SWCON 0.12  
*SOIRW 0.10 *SLCON 0.17 *SGCON 0.00  
4 *SETN 3 *SWCRIT 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 *SWCRIT default. In layer 4, *SWCRIT is being set high to reduce water production.

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)          (*SHUTMOW)
            (*STS)          (*SHUTMOL)
            (*STL)          (*SHUTMOLDOWN)
            (*BHF)          (*SHUTMOLUP)
                           (*SHUTALL)
                           (*SHUTMOWS)

(*MIN)      (*MNP)      value          (*STOP)
            (*CPP)          (*)CONT

(*GTARGET)   (*STO)      value
            (*STG)
            (*STW)
            (*STS)
            (*STL)
            (*BHF)
            (*MNP)
            (*CPP)
            (*NONE)

(*VREP)           vrep_frac
(*RECYCLE)    (*GAS)      recyc_frac
            (*WATER)
            (*SOLVENT)

(*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 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, *WATER or *SOLVENT 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 | cm^3/min) 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 | cm^3/min) 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 | cm^3/min) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

***STS**

This subkeyword identifies a surface solvent rate (m^3/day | scf/day | cm^3/min) 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 total surface liquid rate (oil + water) (m^3/day | stb/day | cm^3/min) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

***MNP, *CPP**

This subkeyword introduces a manifold or compressor pressure (kPa | psi | kPa | kg/cm^2) 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 ‘Entire Field’.

***PMTARG**

Introduces the targeted average hydrocarbon pore-volume pressure (kPa | psi | kPa | kg/cm^2) 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 (c_1, c_2, c_3) must be non-negative real numbers. Defaulted to the internally estimated values. 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

***BHF**

This subkeyword identifies a bottom-hole fluid rate (rm^3/day | rbl/day | rcm^3/min) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

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.

value

Constraint value.

*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. This is the default action if no action subkeyword is found.

*SHUTMOW

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending well (MOW – the well 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 well with 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 well 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.

***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. Not available for MNP and CPP constraints.

***SHUTALL**

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then all currently open wells in the group should be shut. Not available for MNP and CPP constraints.

DEFUALTS:

Optional keyword. Default is no production constraints on groups. 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 mutually 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.

Example: Use IPP to distribute production

```
*GCONP 'Group1'  
      *MAX      *STG 100000.0  
      *GTARGET  *STO   1000.0  
      *APPOR-METHOD *PROD 'Group1' *IP
```

Example: Use oil phase guide rates to distribute production.

```
*GCONP 'Group1'  
      *MAX      *STG 100000.0  
      *GTARGET *STO    1000.0  
*APPOR-METHOD *PROD 'Group1' *GUIDE  
  
*GUIDEP *STO 'well-1' 'well-2' 'well-3'  
          200.0    150.0    650.0
```

Group Injection Constraints (Optional)

*GCONI

PURPOSE:

*GCONI is used to specify group injection controls.

FORMAT:

```
*GCONI group_list
(*MAX)          (*STG)      value           (*STOP)
                (*STW)                  (*CONT)
                (*STS)
                (*BHG)
                (*BHW)
                (*GMP)
                (*WMP)
                (*SMP)
                (*GCP)
                (*WCP)
                (*SCP)

(*GTARGET)       (*STG)      value / *NONE
                (*STW)
                (*STS)
                (*BHG)
                (*BHW)
                (*GMP)
                (*WMP)
                (*SMP)
                (*GCP)
                (*WCP)
                (*SCP)

(*VREP)          (*GAS)      vrep_frac
                (*WATER)
                (*SOLVENT)
                (*GMKUP)
                (*WMKUP)
                (*SMKUP)

(*VREFP)         (*GAS)      vrefp_frac
                (*WATER)
                (*SOLVENT)

(*RECYCLE)        (*GAS)      recyc_frac     (make_up volume)
                (*WATER)
                (*SOLVENT)
```

(*PMAINT)	(*GAS)	(*PMSECT	'sector_name')
	(*WATER)	(*PMTARG	<i>p_targ</i>)
	(*SOLVENT)	(*PMCOEF	<i>c1 c2 c3</i>)
		(*PMMAXR	<i>d1 d2</i>)

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, *BHG

This subkeyword identifies a surface or reservoir gas rate (m^3/day | ft^3/day | cm^3/min) 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, *BHW

This subkeyword identifies a surface or reservoir water rate (m^3/day | bbl/day | cm^3/min) maximum or target. Zero rates are allowed and have the same effect as shutting in all the water injection wells connected to that group.

*STS

This subkeyword identifies a surface solvent rate (m^3/day | scf/day | cm^3/min) maximum or target. Zero rates are allowed and have the same effect as shutting in all the solvent injection wells connected to that group.

*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, *WATER and *SOLVENT specify 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, *WMKUP or *SMKUP to meet a total voidage replacement fraction. One of *GAS, *WATER, *SOLVENT, *GMKUP, *WMKUP or *SMKUP 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. In this case *GAS, *WATER or *SOLVENT 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, *WATER or *SOLVENT is recycled (reinjected) into the reservoir at the same surface conditions specified for the injectors.

*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).

*GMP, *WMP, *SMP

This subkeyword introduces a gas, water or solvent manifold pressure (kPa | psi | kPa | kg/cm²) injection constraint. This subkeyword can only be entered if all of the listed groups have had gas, water or solvent injection identified as going through a manifold with the *MANIFOLD keyword.

*GCP, *WCP, *SCP

This subkeyword introduces a gas, water or solvent compressor (surface) pressure (kPa | psi | kPa | kg/cm²) injection constraint. This subkeyword can only be entered if all of the listed groups have had gas, water or solvent injection identified as going through a manifold with the *MANIFOLD keyword. Also, a hydraulics table for calculation of the gas, water or solvent 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, water or solvent). Existing maximum 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 ‘Entire Field’.
*PMTARG	Introduces the targeted average hydrocarbon pore-volume pressure (kPa psi kPa kg/cm ²) for the sector. The value <i>p targ</i> 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 (<i>c1</i> , <i>c2</i> , <i>c3</i>) 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 (<i>d1</i>) of the total produced (from the same group) and a make-up rate (<i>d2</i>). The values must be non-negative real numbers. The fraction <i>d1</i> is nondimensional and defaulted to 0. The make-up rate <i>d2</i> is (m ³ /day bbl/day cm ³ /min) for water and (m ³ /day scf/day cm ³ /min) for gas or solvent, and is defaulted to 1.0e+20.
*GAS, *WATER, *SOLVENT	Specifies that the gas, water or solvent phase to that is to be injected for voidage replacement, recycle pressure maintenance.
*GMKUP, *WMKUP, *SMKUP	Specifies that gas, water or solvent phase is the make-up stream supplemented to meet the total voidage replacement fraction.
<i>value</i>	Constraint value.
<i>vrep_frac</i>	When the voidage replacement subkeyword is used (*VREP) the <i>vrep_frac</i> 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 <i>vrep_frac</i> entered for a non make-up phase is applied as a target for that phase independently of the other specified values.
<i>vrefp_frac</i>	When the reference volume replacement subkeyword is used (*VREFP) the <i>vrefp_frac</i> 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.

make_up volume

The amount of make-up gas or solvent (m^3/day | scf/day | cm^3/min) to be injected with the recycled gas or solvent. For recycled water the make-up rates are (m^3/day | bbl/day | cm^3/min).

*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 constraints on groups, *make_up volume* defaults to 0.0.

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, *VREFP and *PMAINT are mutually exclusive in one timecard for the same injector stream (gas, water or solvent), and only the latest entry counts.

If a group is assigned by *GCONI with the following dependent constraints: *VREP, *RECYCLE, *VREFP 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 / *SMKUP) 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 and 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.

```
*GCONI 'FIELD'  
*RECYCLE *GAS 0.8  
*VREP *WMKUP 1.0
```

This sets the model to recycle 80% of the gas produced and make-up the voidage replaced to 100% by the injection of water.

Recycling: The following keywords supplement *GCONI and add flexibility to recycling target specification, making recycling data entry more straightforward: *GPRODGROUP, *WPRODGROUP, *SPRODGROUP, *GSALESFRAC, *GSALESMAXR, *GFUELFRAC, *GFUELMAXR, *WMKMAXR, *SMKMAXR, *WMKUPTO, *SMKUPTO, *GMKMAXR, *GMKUPTO, *WRECYMAXR, *SRECYMAXR, and *GRECYMAXR. Please see the pages in this manual describing these keywords.

A water recycling target volumetric rate is now calculated as follows in IMEX:

$$\begin{aligned}\text{water_injection_target} &= \text{wrecyc} + \text{wmakeup} \\ \text{wmakeup} &= \min(\text{max_water_makeup}, \text{total_water_rate} - \text{wrecyc}) \\ \text{wrecyc} &= \min(\text{fwrecyc} \times \text{wprod}, \text{max_water_recycling_rate})\end{aligned}$$

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. total_water_rate is a target injection rate which will be met unless a maximum make-up rate is exceeded. total_water_rate 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 total_water_rate has been entered using the *WMKUPTO keyword, wmakeup is set equal to the specified maximum make-up rate and the min operation in the second equation is not performed.

Solvent re-injection is handled similarly to water re-injection:

$$\begin{aligned}\text{solvent_injection_target} &= \text{srecyc} + \text{smakeup} \\ \text{smakeup} &= \min(\text{max_solvent_makeup}, \text{total_solvent_rate} - \text{srecyc}) \\ \text{srecyc} &= \min(\text{fsrecyc} \times \text{sprod}, \text{max_solvent_recycling_rate})\end{aligned}$$

In the above, max_solvent_makeup is the quantity entered after the solvent recycling fraction fsrecyc on the *RECYCLE *SOLVENT line, or directly using the *SMKMAXR keyword. total_solvent_rate is a target injection rate which will be met unless a maximum solvent make-up rate is exceeded. total_solvent_rate can only be entered with the *SMKUPTO keyword. sprod is the solvent production rate of, by default, the group upon which the solvent recycling injection target is imposed. However sprod can be taken as the solvent production rate of another group named using the *SPRODGROUP keyword. If no total_solvent_rate has been entered using the *SMKUPTO keyword, smakeup is set equal to the specified maximum make-up rate and the min operation in the second equation is not performed.

Gas recycling has been enhanced to allow the specification of fuel and sales rates, and consequently the expressions for gas recycling rates are more complicated. A volumetric gas re-injection target is calculated in IMEX as follows:

$$\begin{aligned} qinj &= qrecyc + qmake_up \\ qmake_up &= \min(\text{max_makeup_rate}, \text{total_gas_rate} - qrecyc) \\ qrecyc &= \min(frecyc \times (\text{qprod} - \text{fuel} - \text{sales}), \text{qrecycmax}) \\ \text{fuel} &= \min(\text{ffrfuel} \times \text{qprod}, \text{max_fuel_rate}) \\ \text{sales} &= \min(\text{frsales} \times (\text{qprod} - \text{fuel}), \text{max_sales_rate}) \end{aligned}$$

In the above, qprod 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. max_makeup_rate is the maximum gas make-up rate entered after frecyc on the *RECYCLE *GAS line, or separately using the *GMKMAXR keyword. total_gas_rate is a target gas injection rate which make-up gas will be injected to meet until the maximum make-up rate is reached. total_gas_rate can be entered only using the *GMKUPTO keyword. If the user does not specify a value for total_gas_rate using the *GMKUPTO keyword, 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. frrfuel is a maximum fraction of the gas production rate which can be 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 is the gas production rate 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 an upper limit to the rate of diversion of produced gas to sales; it can be entered with the *GSALESMAXR keyword for the producing group.

Pressure Maintenance: *PMAINT specifies one special group control which instructs a group or groups to adjust the production or injection rates (at reservoir 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 Q_{targ} of bottom-hole fluid in reference to that of the group injection (or production), Q_{ref} :

$$Q_{\text{targ}} = Q_{\text{ref}} \pm \Delta Q$$

where ΔQ is the bottom-hole correction rate needed to reach or maintain a certain pressure. The sign \pm designates for production (+) or injection (-) target. ΔQ has been set to zero for normal voidage replacement constraints (*VREP under *GCONP / *GCONI). 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

$$\Delta Q = c_1 \left((P - P_{\text{targ}}) + c_2 \left(\frac{dP}{dt} \right) + c_3 \frac{1}{\Delta t} \int_{t-\Delta t}^t (P - P_{\text{targ}}) dt \right)$$

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.

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 'Entire 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).

Example:

This is an example using voidage replacement. Here 50% of the voidage is being replaced by solvent injection and the other 50% by 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  
*APPOR-METHOD *SOLI 'Group1' *IP  
*APPOR-METHOD *GASI 'Group1' *IP
```

Example:

Use IIP to distribute injection

```
*GCONI 'Group1'  
      *GTARGET *STW 1000.0  
*APPOR-METHOD *WATI 'Group1' *IP
```

Example:

Use gas phase guide rates to distribute injection.

```
*GCONI 'Group1'  
      *GTARGET *STG  50000.0  
*APPOR-METHOD *GASI 'Group1' *GUIDE  
*GUIDEI  *STG   'well-1'  'well-2'  'well-3'  
           2000.0    1500.0    6500.0
```

Gas Make-up Target for Group Gas Recycling (Optional)

*GMKUPTO

PURPOSE:

*GMKUPTO is used to specify a total group surface gas injection rate which make-up gas, in general up to a certain maximum rate, will supplement the recycled gas to meet.

In other words the gas rate specified for injection is the rate on the *GMKUPTO keyword. The rate of make-up gas actually injected may be reduced by the rate of gas being recycled and may account for the maximum make-up gas rate specified on the *GMKMAXR keyword.

The rate specified on the *GMKUPTO keyword will also be referred to as the “make_up_to” rate. The maximum gas make-up rate can be specified either with the *GMKMAXR keyword or on the *RECYCLE *GAS line following *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_list* *total_gas_rates*

DEFINITIONS:

group_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 (m^3/day | SCF/day | cm^3/min). 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 (“make_up_to” rate) entered using the *GMKMAXR keyword. If neither a maximum rate using *GMKMAXR nor a “make_up_to” *GMKUPTO rate 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” *GMKUPTO 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.

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_list max_rates*

DEFINITIONS:

group_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 (m^3/day | SCF/day | cm^3/min). 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 rate entered on the *GMKUPTO keyword. If neither a maximum rate nor a "make_up_to" *GMKUPTO rate 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 the rates entered using the *GMKUPTO keyword; if no rate is entered using the *GMKUPTO keyword 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 Producing Group for Group Recycling/Voidage Replacement (Optional)

*GPRODGROUP

PURPOSE:

*GPRODGROUP is used to specify the group from which gas production rates 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_list* *FROM '*producing_group_name*'

DEFINITIONS:

group_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_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 of one group to determine the gas injection rates 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 re-injection targets. Further information is also available from the manual pages for the other gas recycling keywords, which are *GSALESFRAC, *GSALESMAXR, *GFUELFrac, *GFUELMAXR, *GMKMAXR, *GMKUPTO, and *GRECYMAXR.

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 allowed 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_list* *FROM ‘*injection_group_name*’

DEFINITIONS:

production_group_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 allowed 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
```

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_list* *max_recyc_rates*

DEFINITIONS:

group_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum gas 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 | cm³/min). 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.

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_list fuel_fractions**

DEFINITIONS:

group_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.

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_list* *max_fuel_rates*

DEFINITIONS:

group_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 (m^3/day | SCF/day | cm^3/min). 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.

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_list* *sales_fractions*

DEFINITIONS:

group_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.

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_list* *max_sales_rates*

DEFINITIONS:

group_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 (m^3/day | SCF/day | cm^3/min). 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.

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.

In other words the water rate specified for injection is the rate on the *WMKUPTO keyword. The rate of make-up water actually injected may be reduced by the rate of water being recycled and also may account for the maximum make-up water rate specified on the *WMKMAXR keyword.

The rate specified on the *WMKUPTO keyword will also be referred to as the “make_up_to” rate. 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_list* *total_water_rates*

DEFINITIONS:

group_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 (m^3/day | stb/day | cm^3/min). 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 (“make_up_to” rate) entered using the *WMKMAXR keyword. If neither a maximum rate nor a “make_up_to” *WMKUPTO rate 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” *WMKUPTO 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 water injection target 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.

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_list max_rates*

DEFINITIONS:

group_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 (m^3/day | stb/day | cm^3/min). 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 rate entered on the *WMKUPTO keyword. If neither a maximum rate nor a "make_up_to" *WMKUPTO rate 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.

```
*GMKMAXR 'Group1' 'Group2' 300.
```

Here the maximum rate value is, but need not have been, on the same line as the group names.

Water Producing Group for Group Recycling (Optional)

***WPRODGROUP**

PURPOSE:

*WPRODGROUP is used to specify the group from which water/total production rates 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_list* *FROM '*producing_group_name*'

DEFINITIONS:

group_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 based on water/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 water recycling or voidage replacement by water injection targets calculated using the water/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 water rates for recycling injection or voidage replacement by water 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_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 target involves using the water production rates of one group to determine the water injection rate 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 water re-injection targets. Further information is also available from the manual pages for the other water recycling keywords, which are *WMKMAXR, *WMKUPTO, and *WRECYMAXR

Example : Group1 and Group2 have their water recycling injection targets calculated using the gas produced by Group3.

```
*WPRODGROUP 'Group1' 'Group2' *FROM 'Group3'
```

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_list* *max_recyc_rates*

DEFINITIONS:

group_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 (m^3/day | stb/day | cm^3/min). 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 rate 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.

Solvent Make-up Target for Group Solvent Recycling (Optional)

*SMKUPTO

PURPOSE:

*SMKUPTO is used to specify a total group surface solvent injection rate which make-up solvent, in general up to a certain maximum rate, will supplement the recycled solvent to meet.

In other words, the solvent rate specified for injection is the rate on the *SMKUPTO keyword. The rate of make-up solvent actually injected may be reduced by the rate of solvent being recycled and also may account for the maximum make-up solvent rate specified on the *SMKMAXR keyword.

The rate specified on the *SMKUPTO keyword This will also be referred to as the “make_up_to” rate. The maximum solvent make-up rate can be specified either with the *SMKMAXR keyword or on the *RECYCLE *SOLVENT line following *GCONI. The total solvent injection rate can be entered only with *SMKUPTO. The rates entered here have an effect only if a solvent recycling injection target is in force. Solvent recycling injection targets are specified for groups using the *GCONI keyword. The total injection rate specified here differs from a group STS target in that the target entered here is not be met if the make-up solvent must exceed a specified maximum rate in order to meet the total rate.

FORMAT:

*SMKUPTO *group_list* *total_inj_rates*

DEFINITIONS:

group_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the total solvent rates specified after the list. The list need not be contained on a single line.

total_inj_rates

A set of non-negative total injected solvent rate values (m^3/day | SCF/day | cm^3/min). 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 solvent make-up rate is controlled completely by the maximum make-up rate (“make_up_to” rate) entered using the *SMKMAXR keyword. If neither a maximum rate nor a “make-up to” *SMKUPTO rate is specified, then the make-up rate is taken to be zero. The total solvent injection 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 solvent make-up rate is controlled by maximum rates entered using the *SMKMAXR keyword; if no maximum rate is entered either, then no make-up solvent 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 *SMKUPTO. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the total solvent injection rate enters the calculation of a solvent recycling injection target, please see the pages of this manual describing the *GCONI keyword. It is important to note that the “make-up to” *SMKUPTO rate is associated with the group re-injecting the solvent, and not with the group producing it. A specified total injected solvent rate has no effect unless a group solvent recycling target is currently in force (see the *GCONI entry in this manual.). The group solvent injection rate will be set to the total solvent rate unless this total rate exceeds the injection rate of recycled solvent by more than a maximum make-up rate, in which case the total solvent injection rate is set to the recycling rate plus the maximum make-up rate.

Example : Group1 and Group2 both acquire the single specified total solvent injection rate.

```
*SMKUPTO 'Group1' 'Group2' 5.0D+7
```

Here the total solvent injection rate value is, but need not have been, on the same line as the group names.

Maximum Make-up Solvent Rate for Group Solvent Recycling (Optional)

***SMKMAXR**

PURPOSE:

*SMKMAXR is used to specify the maximum surface rate of make-up solvent injected as part of a group solvent recycling target to supplement the recycled fluid. This information can also be specified after the recycling fraction on the *RECYCLE *SOLVENT line after *GCONI. The two methods of entering the make-up rate can over-write each other. Solvent recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *SMKMAXR have no effect unless a group solvent recycling injection target is currently in force.

FORMAT:

SMKMAXR *group_list* *max_rates

DEFINITIONS:

group_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 (m^3/day | SCF/day | cm^3/min). 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 by the rate entered on the *SMKUPTO keyword. If neither a maximum rate nor a "make_up_to" *SMKUPTO rate 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 *SMKUPTO keyword; if no target rate is entered either, then no make-up solvent 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 *SMKMAXR. 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 solvent 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 solvent, and not with the group producing it.

Example : Group1 and Group2 both acquire the single specified maximum make-up solvent injection rate.

```
*SMKMAXR 'Group1' 'Group2' 1.0D+7
```

Here the maximum rate value is, but need not have been, on the same line as the group names.

Solvent Producing Group for Group Recycling (Optional)

***SPRODGROUP**

PURPOSE:

*SPRODGROUP is used to specify the group from which solvent production rates or total production rates will be taken to calculate a solvent recycling injection target or voidage replacement by solvent target specified for another group or groups using the *GCONI keyword.

FORMAT:

*SPRODGROUP *group_list* *FROM '*producing_group_name*'

DEFINITIONS:

group_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will have solvent recycling injection targets or voidage replacement by solvent target based on solvent/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 solvent recycling injection or voidage replacement by solvent targets calculated using the solvent/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 solvent rates for recycling injection targets imposed upon the listed groups. This group name may be but need not be on the same line as the *FROM subkeyword. 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 and VRI_GROUP for more detail.

DEFAULTS:

Optional keyword. If a group's name never appears in a group list following *SPRODGROUP, then by default that group's own solvent/total production rate is used to compute solvent recycling targets or voidage replacement by solvent 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 *SPRODGROUP.

The fraction of solvent recycled to each group in the *group_list* must sum to no more than 1.0. It is possible to reinject more solvent than is produced from the “solvent producing” group if this is not ensured (The fraction of solvent recycled for each group is entered on its *GCONI keyword after the *RECYCLE *SOLVENT sub-keyword).

EXPLANATION:

The imposition of a recycling rate/voidage replacement by solvent target involves using the solvent/total production rates of one group to determine the solvent injection rate 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 solvent re-injection/voidage replacement targets. Further information is also available from the manual pages for the other solvent recycling keywords, which are *SMKMAXR, *SMKUPTO, and *SRECYMAXR.

Example : Group1 and Group2 have their solvent recycling injection targets calculated using the gas produced by Group3.

```
*SPRODGROUP 'Group1' 'Group2' *FROM 'Group3'
```

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.

Maximum Re-injection Rate for Group Solvent Recycling (Optional)

*SRECYMAXR

PURPOSE:

*SRECYMAXR is used to specify a maximum rate at which solvent is re-injected under a group recycling target. Solvent recycling injection targets are specified for groups using the *GCONI keyword. The numbers entered under *SRECYMAXR have no effect unless a group solvent recycling injection target is currently in force.

FORMAT:

*SRECYMAXR *group_list* *max_recyc_rates*

DEFINITIONS:

group_list

A list of one or more valid group names, enclosed in single quotes. The groups named in this list will acquire the maximum solvent 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 | cm³/min). 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 solvent is re-injected under a group solvent 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 *SRECYMAXR, then by default its maximum solvent 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 *SRECYMAXR. Negative values are rejected as being in error.

EXPLANATION:

For a full explanation of how the maximum recycling rates enter the calculation of a solvent 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 solvent, and not with the group producing it. The recycling volumetric injection rate is set principally by the recycling fraction entered with the *RECYCLE *SOLVENT line

following *GCONI, but if a maximum has been imposed with the *SRECYMAXR 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.

```
*SRECYMAXR  '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.

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_list
(*GOR)      value      (action_max)
(*WCUT)
(*WGR)
(*MAXGAS)
(*MAXSTW)
(*MINOIL)    value      (action_min)
(*MINGAS)
(*MINBHF)
```

where

action_max = *STOP | *SHUTMOW | *SHUTMOL | * SHUTMOLDOWN |
 *SHUTMOLUP | *SHUTMOWS | *SHUTALL

action_min = *STOP | *SHUTALL

DEFINITIONS:

group_list

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.

*GOR

This subkeyword identifies a maximum gas-oil ratio (m^3/m^3 | scf/stb | cm^3/cm^3) monitor for group production. Actions specified by action_max are valid for this monitor.

*WCUT

This subkeyword identifies a maximum water-cut (fraction) monitor for group production. Actions specified by action_max are valid for this monitor.

*WGR

This subkeyword identifies a maximum water-gas ratio (m^3/m^3 | stb/scf | cm^3/cm^3) monitor for group production. Actions specified by action_max are valid for this monitor.

***MAXGAS**

This subkeyword identifies a maximum surface gas rate (m^3/day | scf/day | cm^3/min) monitor for group production. Actions specified by action_max are valid for this monitor. 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 (m^3/day | stb/day | cm^3/min) monitor for group production. Actions specified by action_max are valid for this monitor. The “most offending” well or layer for this constraint is deemed to be the one with the highest WCUT rather than the one with the highest water rate.

***MINOIL**

This subkeyword identifies a minimum oil rate (m^3/day | stb/day | cm^3/min) monitor for group production. Actions specified by action_min are valid for this monitor.

***MINGAS**

This subkeyword identifies a minimum gas rate (m^3/day | scf/day | cm^3/min) monitor for group production. Actions specified by action_min are valid for this monitor.

***MINBHF**

This subkeyword identifies a minimum bottom-hole fluid rate (rm^3/day | rbb/day | rcm^3/min) monitor. Actions specified by action_min are valid for this monitor.

value

Constraint value -- units are given under *GOR, *WCUT, *WGR, *MAXGAS, *MAXSTW, *MINOIL, *MINGAS and *MINBHF above.

***STOP**

Action subkeyword indicating that if the constraint is violated then the simulation should be stopped. This is the default action for all constraints if no action is entered explicitly.

***SHUTMOL**

This specifies that if a GOR, WCUT, WGR, MAXGAS, or MAXSTW monitor is violated then the most offending layer (MOL) in the most offending well (the one with the greatest GOR, WCUT, or WGR) should be shut in.

***SHUTMOLDOWN**

This specifies that if a GOR, WCUT, WGR, MAXGAS, or MAXSTW monitor 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, or WGR) should be shut in.

***SHUTMOLUP**

This specifies that if a GOR, WCUT, WGR, MAXGAS, or MAXSTW monitor is violated then the most offending layer (MOL) and the layers above it in the most offending well (the one with the greatest GOR, WCUT, or WGR) should be shut in.

***SHUTMOW**

This specifies that if a GOR, WCUT, WGR, MAXGAS, or MAXSTW monitor is violated, then the most offending well (MOW – the well with the highest GOR, WCUT, or WGR) should be shut.

***SHUTMOWS**

This specifies that if a GOR, WCUT, WGR, MAXGAS, or MAXSTW monitor is violated then a list of prioritized most offending wells (MOWS, the ones with the greater GOR, WCUT, or WGR) should be shut.

***SHUTALL**

This specifies that if the monitored constraint is violated for a group, then all currently open wells in the group 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 values.

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 termination of simulation when the group oil production rate falls below a prescribed limit:

```
*GCONM 'GROUP1'  
*MINOIL 100.0 *STOP
```

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. A number of criteria are available to stipulate the duration of each cycle part.

FORMAT:

```
*GCONCYCLE_START 'group_names'  
*CYCSTREAMS stream_name1 ... stream_namenstreams  
*NPARTS nparts  
*TARGETTYPES  
    stream_name1 type1,1 ... type1,nparts  
    stream_name2 type2,1 ... type2,nparts  
*MAXRATES  
    stream_name1 mxrt1,1 ... mxrt1,nparts  
    stream_name2 mxrt2,1 ... mxrt2,nparts  
*VREPPFR  
    stream_name1 vrep1,1 ... vrep1,nparts  
    stream_name2 vrep2,1 ... vrep2,nparts  
(*MAXCUMS  
    stream_name1 value1,1 ... value1,nparts  
    stream_name2 value2,1 ... value2,nparts)  
(*TOTHEAT  
    stream_name1 value1,1 ... value1,nparts  
    stream_name2 value2,1 ... value2,nparts)  
(*MAXBHP  
    stream_name1 bhp1,1 ... bhp1,nparts  
    stream_name2 bhp2,1 ... bhp2,nparts)  
(*MAXTIMES ntime1 ... ntimenpart)  
(*MINTIMES ntime1 ... ntimenpart)  
(*DTWCYC dtime1 ... dtimenpart)  
(*NCYCLES ncycles)  
(*MINBHP bhp1 ... bhpnparts)  
(*MINQOIL minoil1 ... minoilnparts)  
(*DEPNDX depndx1 ... depndxnparts)  
*GCONCYCLE_END
```

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. The target type and maximum rate (or voidage replacement fraction) provide sufficient information to impose a group target rate for each cycle part for each stream. This keyword is required 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 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 required.

STREAM	IMEX	GEM	STARS
Production ('PROD')	Stock tank oil ('STO') Stock tank gas ('STG') Stock tank water ('STW')	Stock tank oil ('STO') Stock tank gas ('STG') Stock tank water ('STW')	Stock tank oil ('STO') Stock tank gas ('STG') Stock tank water ('STW')

	Bottom hole fluid ('BHF') – includes oil, gas and water phase production at reservoir or bottom hole conditions as well as production of the 4 th phase, if defined such as solvent or polymer	Bottom hole fluid ('BHF') – includes oil, gas and water phase production at reservoir or bottom hole conditions	Bottom hole fluid ('BHF') – includes oil, gas and water phase production at reservoir or bottom hole 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	
Water injection ('WATI')	Stock tank water ('STW') or bottom hole water ('BHW')	Stock tank water ('STW') or bottom hole water ('BHW')	Stock tank water ('STW') or bottom hole water ('BHW')
Gas injection ('GASI')	Stock tank gas ('STG') or bottom hole gas ('BHG')	Stock tank gas ('STG') or bottom hole gas ('BHG')	Stock tank gas ('STG') or bottom hole gas ('BHG')
Solvent injection ('SOLI')	Stock tank solvent ('STS') or bottom hole 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 negative value 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. For example to specify a voidage replacement fraction for the water injection stream and a maximum rate group target for the solvent stream:

```
*cycstreams 'WATI' 'SOLI'
*maxrates
'WATI' -0.5 -0.5
'SOLI' 0.    250.

*vrepfr
'WATI' 1.0 0.0
'SOLI' -0.5 -0.5
```

A negative number specified with *MAXRATES indicates that the group target will be based on voidage replacement fraction. Accordingly a voidage replacement fraction needs to be specified using the keyword *VREPFR for that stream. Similarly a 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. If maximum rate group targets are to be imposed for all streams then the *VREPFR 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 keyword, data for all streams must be specified 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.

This keyword and/or *VREPFR 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	M ³ /day	Bbl/day	Cm ³ /min
WTG, STG, BHG, STS, BHS	M ³ /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, GAS1 or SOL1), 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 reservoir condition is replaced by injection. For a given stream, specify either a maximum rate or a voidage replacement fraction. Since for keywords *MAXRATES and *VREPFR values must be entered for all streams, if a voidage replacement fraction is to be imposed for a given stream, then enter a negative value for that stream with the keyword *MAXRATES. 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 optional.

*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 bottom hole 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 bottom hole pressure will be compared with the actual bottom hole 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 off 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 bottom hole 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 bottom hole pressure will be compared with the actual bottom hole 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 if 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	M ³ /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)	M ³ /day	Bbl/d	Cm ³ /min
GASI (STG/BHG)	M ³ /day	Ft ³ /day	Cm ³ /min
SOLI (STS/BHS) IMEX only	M ³ /day	Ft ³ /day	Cm ³ /min

DEFAULTS:

The default is to continue applying values originally specified with *GCONCYCLE_START.

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_list
  (Any of *TARGETTYPES, *MAXCUM, *MAXTIMES, *NCYCLES,
   *MAXBHP, *MININJ, *MINBHP, *MINOIL, *DEPNDX, *TOTHEAT
  *GCONCYCR_END
```

DEFUALTS:

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_list* (*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'
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 off
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_list cycle_part_start  
*CYCPRT_END   group_list 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 *GCONCYCLE *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 off
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
```

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 stream group_list (*ON | *OFF)
      stream = *PROD | *GASI | *WATI | *SOLI
```

DEFINITIONS:

*PROD

Indicates that group production should be treated as going through a manifold.

*GASI, *WATI, *SOLI

Indicates that group gas, water or solvent 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 four 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 *PWBLORE 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 *IWBLORE 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 stream *group_list* (*ON | *OFF)
stream = *PROD | *GASI | *WATI | *SOLI

DEFINITIONS:

*PROD

Indicates that pressure translation should be applied for pressure constraints upon the production manifolds of the listed groups.

*GASI, *WATI, *SOLI

Indicates that pressure translation should be applied for manifold group gas, water or solvent 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 four 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 stream *group_list* *table_list*
stream = *PROD | *GASI | *WATI | *SOLI

DEFINITIONS:

***PROD**

Indicates that the pressure table numbers are being specified for production manifolds.

***GASI, *WATI, *SOLI**

Indicates that the pressure table numbers are being specified for gas, water or solvent 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_list

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 four 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. The hydraulics table to be used is identified using the *GPTABLE keyword.

FORMAT:

*GPHYDDEP stream *group_list depths*
stream = *PROD | *GASI | *WATI | *SOLI

DEFINITIONS:

***PROD**

Indicates that the depths are being specified for production manifolds.

***GASI, *WATI, *SOLI**

Indicates that the depths are being specified for gas, water or solvent 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 | cm | m) 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 four 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 stream *group_list* *alq_values*
stream = *PROD | *GASI | *WATI | *SOLI

DEFINITIONS:

*PROD

Indicates that the ALQ values are being specified for production manifolds.

*GASI, *WATI, *SOLI

Indicates that the ALQ values are being specified for gas, water or solvent injection manifolds.

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 four 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 *GASI 'Group-1' 'Group-2'  
'Group-3' 1. 0.5 0.3
```

This specifies that the gas injection 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 compressor.

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

One or more well numbers or quoted well names assigned with the *WELL keyword to which the current ALQ specification applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

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

PURPOSE:

*DRILLQ allows the specification of the order in which wells must be drilled automatically.

*DRILLT allows the specification of the minimal time intervals between auto-drilling producers and injectors, respectively.

FORMAT:

*DRILLQ *IPP | *IIP | *well_list*
*DRILLT *tinv_p tinv_j*

DEFINITIONS:

*IPP

Specifies that the instantaneous production potential of wells with status *AUTODRILL will be used to determine the order in which these wells will be opened. *IPP applies to producers only.

*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 opened. *IIP applies to injectors only.

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword 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. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding.

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.

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.

CONDITIONS:

*DRILLQ and *DRILLT 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.

EXPLANATION:

This optional keyword is used to specify the order in which wells that have *AUTODRILL as their status will be drilled. One may either use the instantaneous injection / production potentials 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. If not all wells are on the list then those on the list will be opened first and the rest will be opened according to their IIP or IPP.

Example:

```
*GCONP 'Group1'  
*MAX *STO 1000.0  
*APPOR-METHOD *PROD 'Group1' *GUIDE  
*GUIDEP *STO 'well-1' 'well-2' 'well-3'  
          200.0    150.0    650.0  
*DRILLQ 'well-4' 'well-5' 'well-6'  
*DRILLT 10.0 0.0 ** No delay for injectors
```

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:

FORMAT:

*GAPPOR	<i>group_list</i>
*AUTODRILL	(*ON *OFF stream)

DEFINITIONS:

group_list

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 apportioned using one of the available apportionment methods specified by *APPOR-METHOD.

*AUTODRILL (*ON | *OFF | stream)

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

stream = *STO | *STG | *STW | *STS | *STL | *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 *GTARGET under *GCONP or *GCONI.

FORMAT:

*APPOR-METHOD stream *group_list* method
stream = *PROD | *GASI | *WATI | *SOLI
method = *IP | *GUIDE | *INGUIDE | *PRIOR

DEFINITIONS:

*PROD

Indicates that group production targets should be apportioned with the specified method.

*GASI, *WATI, *SOLI

Indicates that group gas, water or solvent 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 also 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 remainder 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 does not present. Unlike the user-supplied guide rate (*GUIDE), the internally-generated guide rates are time dependent. However, any well violating the constraint is still cut back to its maximum rate as in *GUIDE.

***PRIORITY**

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 does 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.

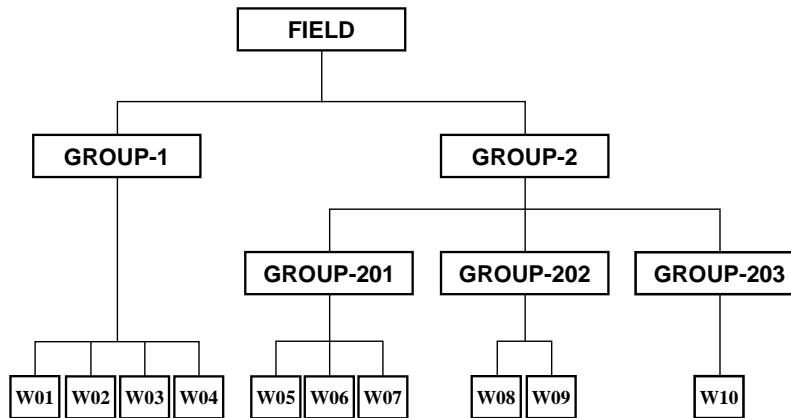
CONDITIONS:

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    'Group-2'
  *PRIOR-NUMER  0.    1.    0.    0.    0.    0.    0.
  *PRIOR-DENOM  0.    0.    1.    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. One 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 data sets. However, the backward compatibility is still maintained and any old data set 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
              *APPOR-METHOD *SOLI 'all_groups' *IP (for IMEX only)

```

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 data set. Therefore, old data sets containing *GUIDEP and *GUIDEI will run exactly the same.

```

*GUIDEPold 'ref stream' guide_rates
      = *APPOR-METHOD *PROD 'all_groups' *GUIDE
      *GUIDEP 'ref stream' guide_rates

*GUIDEIold *STG guide_rates
      = *APPOR-METHOD *GASI 'all_groups' *GUIDE
      *GUIDEI *STG guide_rates

*GUIDEIold *STW guide_rates
      = *APPOR-METHOD *WATI 'all_groups' *GUIDE
      *GUIDEI *STW guide_rates

*GUIDEIold *STS guide_rates          (for IMEX only)
      = *APPOR-METHOD *SOLI 'all_groups' *GUIDE
      *GUIDEI *STS 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 stream group_list
  (*PRIOR-RATE    *MRC | *BHP (bhp_val) | *WBHP)
  (*PRIOR-CTRL   freq tcr_min tcr_max)
  (*PRIOR-NUMER  A0 A1 ... Anph)
  (*PRIOR-DENOM  B0 B1 ... Bnph)
  stream = *PROD | *GASI | *WATI | *SOLI
```

DEFINITIONS:

*PROD

Indicates that the priority formula is applied to the group production targets with the priority ranking method.

*GASI, *WATI, *SOLI

Indicates that the priority formula is applied to the group gas, water or solvent injection targets with the priority ranking method.

group_list

A list of group names to which the priority formula is applied. Wildcard characters may also 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 bottom hole pressure, *bhp_val*. If *bhp_val* 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 no 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 ($i = 1, nph$). ‘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 ($i = 1, nph$).

DEFUALTS:

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-DEMON, 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 formula 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

$$\text{Priority}(iw) = \frac{A_0(ig) + \sum_{i=1}^{nph} A_i(ig)Q_i(iw)}{B_0(ig) + \sum_{i=1}^{nph} B_i(ig)Q_i(iw)}$$

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 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 bottom hole 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.

IMEX (Production: nph = 6)

i	CONST	*STO	*STG	*STW	*STS	*STL	*BHF
NUMER	A_0	A_1	A_2	A_3	A_4	A_5	A_6
DEMON	B_0	B_1	B_2	B_3	B_4	B_5	B_6

IMEX (Injection: nph = 3)

i	CONST	*STG	*STW	*STS
NUMER	A_0	A_2	A_3	A_4
DEMON	B_0	B_2	B_3	B_4

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 shifting 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.

Examples:

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 order of 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    STS    STL    BHF
*PRIOR-NUMER 0.      1.      0.      0.      0.      0.
*PRIOR-DENOM 0.      0.      1.      0.      0.      0.
```

Wells can also be opened in order of decreasing WCUT:

```
*PRIOR-NUMER 0.      1.      0.      1.      0.      0.      0.
*PRIOR-DENOM 0.      0.      0.      0.      1.      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.      0.      0.      0.      0.
*PRIOR-DENOM 0.      0.      0.      0.01 1.      0.      0.
```

In conjunction with the trigger action (*TRIGGER), appropriate priority formulae can also be used alternatively to minimize more than one stream. The following example shows the flexibility to reduce the gas production using the priority formula based on gas oil ratio. One can switch to that based on the water cut when *WCUT > 0.5 and *GOR < 200.0 through nested trigger actions.

```
*GCONP 'FIELD' *GTARGET *STO 500.0
*APPOR-METHOD *PROD 'FIELD' *PRIOR

*PRIOR-FORM 'FIELD'
    *PRIOR-CTRL 30.  0.  1.
    *PRIOR-NUMER 0.  1.  0.  0.  0.  0.  0.
    *PRIOR-DENOM 0.  0.  1.  0.  0.  0.  0.

*TRIGGER 'REDUCE_WCUT1' *ON_GROUP 'FIELD' *WCUT > 0.5 APPLY_TIMES 5
    *TRIGGER 'REDUCE_WCUT2' *ON_GROUP '@' *GOR < 200.0
        *PRIOR-FORM '@'
            *PRIOR-NUMER 0.  1.  0.  1.  0.  0.  0.
            *PRIOR-DENOM 0.  0.  0.  1.  0.  0.  0.

        *END_TRIGGER
    *END_TRIGGER
```

Guide Rates for Groups or Wells

*GUIDEP, *GUIDEI

PURPOSE:

*GUIDEP or *GUIDEI specifies the use of guide rates to distribute the production or injection rates to groups or wells so as to meet the production or injection target.

FORMAT:

```
*GUIDEP | *GUIDEI      type  list  guide_rates  
list   = group_list | well_list  
type   = *STO | *STG | *STW | *STS | *STL | *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. Not valid for use with *GUIDEI. Rates should be entered as (m³/day | stb/day | cm³/min).

*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 as (m³/day | scf/day | cm³/min).

*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 as (m³/day | stb/day | cm³/min).

*STS

This subkeyword identifies that the solvent stream is the reference stream to which the guide rates apply; i.e., that the guide rate values entered should be interpreted as solvent rates or as being proportional to solvent rates. Rates should be entered as (m³/day | scf/day | cm³/min).

*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 as (m³/day | stb/day | cm³/min).

***BHF**

Identifies that guide rate values are bottom-hole fluid rates. Rates should be entered in (rm³/day | rbbl/day | rcm³/min). 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. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

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.

If only one value of *guide_rates* is input, all wells or groups defined in the *well_list* or *group_list* will be set to that single value.

DEFAULTS:

Optional keywords. Default is to use the internal guide rate apportionment method (*APPOR-METHOD *INGUIDE) to determine rate distributions among groups or wells to meet the target rate.

CONDITIONS:

*GUIDEP AND *GUIDEI must be located in the Well and Recurrent Data keyword group, and must follow the *GCONI, *GCONP or *GCONM keywords.

EXPLANATION:

Guide rates for each of the wells specified in *well_list* or each group specified by the *group_list*. If guide rates are being used then all producers or injectors connected to that group must have their *guide_rates* specified. 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.

When guide rates are used, the contributing wells or groups receive a fraction of the target which is proportional to its guide rate. If this assignment exceeds the well's or the group's maximum rate, then the maximum rate is assigned and the remainder of the target is apportioned among the other contributors, with the contributions still proportional to the guide rates but with the contributors set at maximum left out of the apportionment.

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. 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 | *GCPON      list  
-or-  
*GCIOFF | *GCION      stream      list  
list = group_list | well_list  
stream = *GAS | *WATER | *SOLVENT
```

DEFINITIONS:

***GAS, *WATER, *SOLVENT**

Indicates under or not under group apportionment for gas, water or solvent injection calculations (stream identifier *GASI, *WATI or *SOLI is also acceptable).

group_list

A list of group names. No wildcarding is allowed for group names.

well_list

A list of well names. See ‘Using Wildcards in Well Lists’ in the Tutorial Section for more information about using wildcarding.

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'
```

Allows for a Set of Keywords to be Processed When a Specified Condition (Trigger) is Satisfied (Optional) *TRIGGER, *END_TRIGGER

PURPOSE:

*TRIGGER allows the user to specify certain actions which are implemented when a specific condition or trigger is satisfied during the simulation.

FORMAT:

```
*TRIGGER 'trig_name' trig_def
  (*APPLY_TIMES napt)
  (*INCREMENT rinc)
  (*AVRGTIME avrgt)
  (*TEST_TIMES ntestt)
  (*TEST_AFTER_TIMER rtimedr)
  (*TEST_AFTER_TIMEA rtimedra)
  {list of actions}
*END_TRIGGER
```

where **trig_def** takes the form of

trig_def = OBJECT condition operator condition_value

with the testing OBJECT being one of

```
(*ON_WELL      'well_names')
(*ON_GROUP    'group_names')
(*ON_LAYER    'well_name' layer_UBA)
(*ON_SECTOR   'sector_name')
(*ON_FIELD    'FIELD')
(*ON_ELAPSED  'TIME')
(*ON_REPGRP   'reporting_group_name')
(*ON_CYCGRP   'cycling_group_name')
(*ON_CTRLLUMP 'control_lump_name')
(*ON_RPTLUMP  'report_lump_name')
```

The {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.

*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).

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.

condition

Enter a single keyword identifying **one** of the following variables shown in Table 1 below: This is a required token and must immediately follow the well name or well list.

Table 1: Well quantities:

Keyword	Meaning
STO-RP	Oil production rate at surface conditions (m^3/D stb/D).
STO-CP	Oil cumulative production at surface conditions (m^3 stb).
STO-RI	Oil injection rate at surface conditions (m^3/D stb/D).
STO-CI	Oil cumulative injection at surface conditions (m^3 stb).
STW-RP	Water production rate at surface conditions (m^3/D stb/D).
STW-CP	Water cumulative production at surface conditions (m^3 stb).
STW-RI	Water injection rate at surface conditions (m^3/D stb/D).
STW-CI	Water cumulative injection at surface conditions (m^3 stb).
STG-RP	Gas production rate at surface conditions (m^3/D scf/D).
STG-CP	Gas cumulative production at surface conditions (m^3 scf).
STG-RI	Gas injection rate at surface conditions (m^3/D scf/D).
STG-CI	Gas cumulative injection at surface conditions (m^3 scf).
BHF-RP	Oil + water + gas + solvent production rate at bottom hole conditions (m^3/D bbl/D).
BHF-CP	Oil + water + gas + solvent production cumulative at bottom hole conditions (m^3/D bbl/D).
BHF-RI	Oil + water + gas + solvent injection rate at bottom hole conditions (m^3/D bbl/D).
BHF-CI	Oil + water + gas + solvent injection rate at bottom hole conditions (m^3/D bbl/D).
STL-RP	Oil + water production rate at surface conditions (m^3/D bbl/D).
STL-CP	Oil + water production cumulative at surface conditions (m^3/D bbl/D).

STS-RP	Solvent production rate at surface conditions (m^3/D scf/D).
STS-CP	Solvent cumulative production at surface conditions (m^3 scf).
STS-RI	Solvent injection rate at surface conditions (m^3/D scf/D).
STS-CI	Solvent cumulative injection at surface conditions (m^3 scf).
STR-RP	Solvent production rate at reservoir conditions (m^3/D ft^3/D).
STR-CP	Solvent cumulative production at reservoir conditions (m^3 ft^3).
STR-RI	Solvent injection rate at reservoir conditions (m^3/D ft^3/D).
STR-CI	Solvent cumulative injection at reservoir conditions (m^3 ft^3).
PLY-RP	Polymer mass production rate (kg/D lbm/D).
PLY-CP	Polymer mass production cumulative (kg lbm).
PLY-RI	Polymer mass injection rate (kg/D lbm/D).
PLY-CI	Polymer mass injection cumulative (kg lbm).
BHP	Bottom hole pressure (kPa psi).
THP	Tubing head pressure (kPa psi).
GOR	Gas oil ratio at surface conditions (m^3/m^3 scf/stb).
WCUT	Ratio of surface production rate of water phase divided by total surface liquid (water + oil) production rate.
WGR	Ratio of water production rate at surface conditions divided by surface gas production rate (m^3/m^3 stb/scf).
GLR	Ratio of gas production rate at surface conditions divided by surface liquid (water + oil) production rate (m^3/m^3 scf/stb).
BKFLOW	If any layer 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
GLIFTR	Gas lift rate (m^3/day scf/day).
GLIFTC	Gas lift cumulative volume (m^3 scf).
GGLOR	Gas lift rate + gas production rate divided by oil production rate (m^3/m^3 scf/stb).
GGLIFTR	Gas lift rate + gas production rate (m^3/day scf/day).
GGLIFTC	Gas lift cumulative + gas produced cumulative (m^3 scf).
GGLLIQR	Gas lift rate plus gas production rate divided by liquid (oil + water) production rate (m^3/m^3 scf/stb).
GLOILR	Gas lift rate divided by oil production rate (m^3/m^3 scf/stb).
GLLIQR	Gas lift rate divided by liquid (oil + water) production rate (m^3/m^3 scf/stb).
WTHTEMP	Tubing-head temperature (C° F° $^\circ\text{C}$), see keyword *TWELLBORE and *PTUBE1 *WHT.

*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.

group_names	<p>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.</p> <p>Note: wildcards may be used in the 'group_names' string as follows:</p> <ul style="list-style-type: none"> * 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'). <p>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.</p> <p>At least one group name must immediately follow the *ON_GROUP keyword. Required token.</p>
-------------	---

condition	<p>The valid list of conditions for groups and for the field is shown in Table 2 below:</p>
-----------	---

Table 2: Group and Field quantities:

Keyword	Meaning
STO-RP	Oil production rate at surface conditions (m ³ /D stb/D).
STO-CP	Oil cumulative production at surface conditions (m ³ stb).
STO-RI	Oil injection rate at surface conditions (m ³ /D stb/D).
STO-CI	Oil cumulative injection at surface conditions (m ³ stb).
STW-RP	Water production rate at surface conditions (m ³ /D stb/D).
STW-CP	Water cumulative production at surface conditions (m ³ stb).
STW-RI	Water injection rate at surface conditions (m ³ /D stb/D).
STW-CI	Water cumulative injection at surface conditions (m ³ stb).
STG-RP	Gas production rate at surface conditions (m ³ /D scf/D).
STG-CP	Gas cumulative production at surface conditions (m ³ scf).
STG-RI	Gas injection rate at surface conditions (m ³ /D scf/D).
STG-CI	Gas cumulative injection at surface conditions (m ³ scf).
BHF-RP	Oil + water + gas + solvent production rate at bottom hole conditions (m ³ /D bbl/D).
BHF-CP	Oil + water + gas + solvent production cumulative at bottom hole conditions (m ³ /D bbl/D).
BHF-RI	Oil + water + gas + solvent injection rate at bottom hole conditions (m ³ /D bbl/D).
BHF-CI	Oil + water + gas + solvent injection rate at bottom hole conditions (m ³ /D bbl/D).
STL-RP	Oil + water production rate at surface (m ³ /D bbl/D).

STL-CP	Oil + water production cumulative at surface conditions (m^3/D bbl/D).
STS-RP	Solvent production rate at surface conditions (m^3/D scf/D).
STS-CP	Solvent cumulative production at surface conditions (m^3 scf).
STS-RI	Solvent injection rate at surface conditions (m^3/D scf/D).
STS-CI	Solvent cumulative injection at surface conditions (m^3 scf).
STR-RP	Solvent production rate at reservoir conditions (m^3/D cf/D).
STR-CP	Solvent cumulative production at reservoir conditions (m^3 cf).
STR-RI	Solvent injection rate at reservoir conditions (m^3/D cf/D).
STR-CI	Solvent cumulative injection at reservoir conditions (m^3 cf).
PLY-RP	Polymer mass production rate (kg/D lbm/D).
PLY-CP	Polymer mass production cumulative (kg lbm).
PLY-RI	Polymer mass injection rate (kg/D lbm/D).
PLY-CI	Polymer mass injection cumulative (kg lbm).
GOR	Gas oil ratio at surface conditions (m^3/m^3 scf/stb).
WCUT	Ratio of surface production rate of water phase divided by total surface liquid (water + oil) production rate.
WGR	Ratio of water production rate at surface conditions divided by surface gas production rate (m^3/m^3 stb/scf).
GLR	Ratio of gas production rate at surface conditions divided by surface liquid (water + oil) production rate (m^3/m^3 scf/stb).
GLIFTR	Gas lift rate (m^3/day scf/day).
GLIFTC	Gas lift cumulative volume (m^3 scf).
GGLOR	Gas lift rate plus gas production rate divided by oil production rate (m^3/m^3 scf/stb).
GGLIFTR	Gas lift rate + gas production rate (m^3/day scf/day).
GGLIFTC	Gas lift cumulative + gas produced cumulative (m^3 scf).
GGLLIQR	Gas lift rate plus gas production rate divided by liquid (oil+water) production rate (m^3/m^3 scf/stb).
GLOILR	Gas lift rate divided by oil production rate (m^3/m^3 scf/stb).
GLLIQR	Gas lift rate divided by liquid (oil + water) production rate (m^3/m^3 scf/stb).
VOIDRPG	Voidage replacement ratio by gas injection
VOIDRPW	Voidage replacement ratio by water injection
VOIDRPS	Voidage replacement ratio by solvent injection
VOIDRPV	Voidage replacement ratio by gas + solvent injection
VOIDRPT	Voidage replacement ratio by all injection streams

*ON_LAYER

This subkeyword identifies 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. For layer rate, a positive number entered for a layer belonging to an injection well implies flow is expected to be from well bore 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 well bore.

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: $i_1 j_1 k_1 / i_2 j_2 k_2 / \dots$ please review the manual pages on the *PERF keyword for more information. Do NOT encapsulate the user block address in quotes.

condition
The valid list of conditions for layers is shown in table 3 below:

Table 3 : Layer quantities:

Keyword	Meaning
STO-R	Oil phase rate at surface conditions for a layer (m^3/D stb/D). If a negative number is entered, then the user is testing for a back flowing layer. The magnitude of the number indicates the severity of the back flow. Example 1: <code>*TRIGGER 'trig1' *ON_LAYER 'well1' 1 1 1 *STO-R > 300.0</code> 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 well bore (production) and the oil flow rate is greater than 300.0. Example 2: <code>*TRIGGER 'trig2' *ON_LAYER 'well1' 1 1 1 *STO-R > -300.0</code> The condition will be satisfied if the layer is back flowing, for an injector flow is from the reservoir to the well bore 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 well bore 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 (m^3/D stb/D).
STG-R	Gas phase rate at surface conditions (m^3/D scf/D).
STS-R	Solvent phase rate at surface conditions (m^3/D scf/D).
GOR	Gas oil ratio at surface conditions (m^3/m^3 scf/stb).
WCUT	Ratio of surface production rate of water phase divided by total surface liquid (water + oil) production rate.
WGR	Ratio of water production rate at surface conditions divided by surface gas production rate (m^3/m^3 stb/scf).
GLR	Ratio of gas production rate at surface conditions divided by surface liquid (water + oil) production rate (m^3/m^3 scf/stb).

BHP	Bottom hole pressure (kPa psi).
DWN	Difference in absolute value well layer pressure minus pressure of well block where well layer is completed (kPa psi).
*ON_SECTOR	<p>This subkeyword identifies 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'.</p> <p>Any other sector name used with a trigger must be previously defined.</p>
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' ('Entire Field') sector is created by default. The name must be on the same line as the *TRIGGER keyword.

condition
The valid list of conditions for sectors is shown in table 4 below:

Table 4 : Sector quantities:

Keyword	Meaning
PAVE	Average pressure calculated on the basis of total pore volume (kPa psi).
SOAVE	Average oil saturation calculated on the basis of total pore volume.
SWAVE	Average water saturation calculated on the basis of total pore volume.
SGAVE	Average gas saturation calculated on the basis of total pore volume.
STOIP	Oil in place at standard conditions (m^3 stb).
STWIP	Water in place at standard conditions (m^3 stb).
STFGIP	Gas in place based on reservoir gas phase alone at standard conditions (m^3 scf).
STGIP	Gas in place based on both reservoir gas and reservoir oil phase at standard conditions (m^3 scf).
PMAX	Maximum in pressure (kPa psi).
PMIN	Minimum in pressure (kPa psi).
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 identifies 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.

condition

The valid list of conditions for field is identical to that of groups as shown in Table 2.

*ON_ELAPSED

This subkeyword identifies 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.

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.

condition

The valid list of quantities for reporting groups is shown in Table 5 below:

Table 5: Reporting Group quantities:

Keyword	Meaning
STO-RP	Oil production rate at surface conditions (m^3/D stb/D).
STO-CP	Oil cumulative production at surface conditions. (m^3 stb).
STO-RI	Oil injection rate at surface conditions (m^3/D stb/D).
STO-CI	Oil cumulative injection at surface conditions. (m^3 stb).
STW-RP	Water production rate at surface conditions (m^3/D stb/D).
STW-CP	Water cumulative production at surface conditions (m^3 stb).
STW-RI	Water injection rate at surface conditions (m^3/D stb/D).
STW-CI	Water cumulative injection at surface conditions (m^3 stb).
STG-RP	Gas production rate at surface conditions (m^3/D scf/D).
STG-CP	Gas cumulative production at surface conditions (m^3 scf).
STG-RI	Gas injection rate at surface conditions (m^3/D scf/D).
STG-CI	Gas cumulative injection at surface conditions (m^3 scf).
BHF-RP	Oil + water + gas + solvent production rate at bottom hole conditions (m^3/D bbl/D).
BHF-CP	Oil + water + gas + solvent production cumulative at bottom hole conditions (m^3/D bbl/D).
BHF-RI	Oil + water + gas + solvent injection rate at bottom hole conditions (m^3/D bbl/D).
BHF-CI	Oil + water + gas + solvent injection cumulative at bottom hole conditions (m^3/D bbl/D).
STL-RP	Oil + water production rate at surface (m^3/D bbl/D).
STL-CP	Oil + water production cumulative at surface conditions (m^3/D bbl/D).
STS-RP	Solvent production rate at surface conditions (m^3/D scf/D).
STS-CP	Solvent cumulative production at surface conditions (m^3 scf).
STS-RI	Solvent injection rate at surface conditions (m^3/D scf/D).
STS-CI	Solvent cumulative injection at surface conditions (m^3 scf).
STR-RP	Solvent production rate at reservoir conditions (m^3/D cf/D).
STR-CP	Solvent cumulative production at reservoir conditions (m^3 cf).
STR-RI	Solvent injection rate at reservoir conditions (m^3/D cf/D).
STR-CI	Solvent cumulative injection at reservoir conditions (m^3 cf).
GOR	Gas oil ratio at surface conditions group (m^3/m^3 scf/stb).
WCUT	Ratio of surface production rate of water phase divided by total surface liquid (water + oil) production rate.
WGR	Ratio of water production rate at surface conditions divided by surface gas production rate (m^3/m^3 stb/scf).
GLR	Ratio of gas production rate at surface conditions divided by surface liquid (water + oil) production rate (m^3/m^3 scf/stb).

*ON_CTRLLLUMP, *ON_RPTLUMP

This subkeyword indicates that the test condition is to be applied to a control / reporting lump. The control / reporting lump must be previously defined to enable control / reporting lump based triggers to be used. This token must immediately follow the trigger name string. For layer lump rate, + for injection and – for production.

lump_name	A lump name (in quotes) to specify the lump to which this trigger applies. The lump must be previously defined. There are no lumps created by default. The lump name must immediately follow the *ON_CTRLLUMP or *ON_RPTLUMP keyword. Required token.
condition	The valid list of conditions for lumps is shown in Table 6 below.

Table 6: Control and Report Lump quantities:

Keyword	Meaning
OILSRFR	Oil surface flow rate (all layers in lump) (m ³ /day stb/day).
OILSRFC	Oil surface cumulative (all layers in lump) (m ³ stb).
STO-RP	Oil surface production rate (producer layers) (m ³ /day stb/day).
STO-CP	Oil surface production cumulative (producer layers) (m ³ stb).
STO-RI	Oil surface injection rate (injector layers) (m ³ /day stb/day)
STO-CI	Oil surface injection cumulative (injector layers) (m ³ stb).
WATSRFR	Water surface flow rate (all layers in lump) (m ³ /day stb/day).
WATSRFC	Water surface cumulative (all layers in lump) (m ³ stb).
STW-RP	Water surface production rate (producer layers) (m ³ /day stb/day).
STW-CP	Water surface production cumulative (producer layers) (m ³ stb).
STW-RI	Water surface injection rate (injector layers) (m ³ /day stb/day).
STW-CI	Water surface injection cumulative (injector layers) (m ³ stb).
GASSRFR	Gas surface flow rate (all layers in lump) (m ³ /day stb/day).
GASSRFC	Gas surface cumulative (all layers in lump) (m ³ stb).
STG-RP	Gas surface production rate (producer layers) (m ³ /day stb/day).
STG-CP	Gas surface production cumulative (producer layers) (m ³ stb).
STG-RI	Gas surface injection rate (injector layers) (m ³ /day stb/day).
STG-CI	Gas surface injection cumulative (injector layers) (m ³ stb).
4THSRFR	Solvent (m ³ /day scf/day) / Light oil (m ³ /day stb/day) / Polymer flow (kg/day lbm/day) rate (all layers in lump).
4THSRFC	Solvent (m ³ scf) / Light oil (m ³ stb) / Polymer (kg lbm) cumulative (all layers in lump).
4TH-RP	Solvent (m ³ /day scf/day) / Light oil (m ³ /day stb/day) / Polymer (kg/day lbm/day) surface production rate (producer layers).
4TH-CP	Solvent (m ³ scf) / Light oil (m ³ stb) / Polymer (kg lbm) surface production cumulative (producer layers).
4TH-RI	Solvent (m ³ /day scf/day) / Light oil (m ³ /day stb/day) / Polymer (kg/day lbm/day) surface injection rate (injector layers).
4TH-CI	Solvent (m ³ scf) / Light oil (m ³ stb) / Polymer (kg lbm) surface injection cumulative (injector layers).
LIQSRFR	Water + oil surface flow rate (all layers in lump) (m ³ /day stb/day).
LIQSRFC	Water + oil surface cumulative (all layers in lump) (m ³ stb).
STL-RP	Water + oil surface production rate (producer layers) (m ³ /day stb/day).
STL-CP	Water + oil surface production cumulative (producer layers) (m ³ stb).

STL-RI	Water + oil surface injection rate (injector layers) (m^3/day stb/day).
STL-CI	Water + oil surface injection cumulative (injector layers) (m^3 stb).
BHFSRFR	Water + oil + gas + solvent reservoir flow rate (all layers in lump) (m^3/day stb/day).
BHFSRFC	Water + oil + gas + solvent reservoir cumulative (all layers in lump) (m^3 stb).
BHF-RP	Water + oil + gas + solvent reservoir production rate (producer layers) (m^3/day stb/day).
BHF-CP	Water + oil + gas + solvent reservoir production cumulative (producer layers) (m^3 stb).
BHF-RI	Water + oil + gas + solvent reservoir injection rate (injector layers) (m^3/day stb/day).
BHF-CI	Water + oil + gas + solvent reservoir injection cumulative (producer layers) (m^3 stb).
GOR	Gas oil ratio at surface conditions (producer layers) (m^3/m^3 scf/stb).
WCUT	Water cut ratio in percent at surface conditions (producer layers)
WGR	Water gas ratio at surface conditions (producer layers) (m^3/m^3 stb/scf).
GLR	Gas liquid ratio at surface conditions (producer layers) (m^3/m^3 scf/stb).
OGR	Oil gas ratio at surface conditions (producer layers) (m^3/m^3 stb/scf).

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 trigger condition and unit system selected for the simulation.

*APPLY_TIMES *napt*

Subkeyword used to specify the maximum number of times that the actions specified with the trigger can be taken. An integer number (*napt*) must immediately follow this subkeyword. This subkeyword is optional. If no value is entered, then the default is 1 and 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 *rinc*

Subkeyword used to specify the increment to the trigger value. A single real number (*rinc*) must follow this subkeyword. This subkeyword is optional. The trigger increment can be a negative number. 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 $\text{value_new} = \text{existing_value} + \text{increment}$. The new value is then used in testing the trigger condition for subsequent times until the trigger condition is once again satisfied.

***AVRGTIME** *avgrt*

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 (*avgrt*) must follow this subkeyword. This subkeyword is optional. 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** *ntestt*

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 (*ntestt*) must follow this subkeyword. This subkeyword is optional. 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** *rtimedr*

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 (*rtimedr*) must follow this subkeyword. This subkeyword is optional. If no value is entered, then the default is to assume a time delay of zero. In days for FIELD/SI/MODSI units and in minutes for laboratory (LAB) unit system.

***TEST_AFTER_TIMEA** *rtimeda*

Subkeyword used to specify the time delay in days for FIELD/SI/MODSI units and in minutes for laboratory (LAB) units, 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 (*rtimeda*) must follow this subkeyword. This subkeyword is optional. If no value is entered, then the default is to assume a time delay of zero. In days for FIELD/SI/MODSI units and in minutes for laboratory (LAB) unit system.

{list of action}

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.

*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.

*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.

DEFAULTS:

*TRIGGER is an optional keyword.

CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group.
The entire *TRIGGER keyword must appear on one line followed by a list of actions and then by the *END_TRIGGER.

EXPLANATION:

This keyword allows for certain actions to be implemented when a specified condition(s) 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, *REFINE, *AMALGAMATE, *DEALMAGAMATE, *DEREFINE, *REREFINE, *AMALGAMATE-C, *DEREFINE-C, *AIMSET, *TRANSI, *TRANSJ, *TRANSK, *TRANLI, *TRANLJ and *TRANLK, *RANGE, *TRANSIJ+, *TRANSIJ-, *TRANSIK-, *TRANSIK+, *TRANSIENT, *THTYPE, any other valid keywords that can otherwise be used in the WELL and RECURRENT data section of a input data file can be used with triggers.

Note: some of the above keywords exceptions are not valid with the IMEX simulator.

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 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 IMEX 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 gas rate = 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 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 'well2' *STO 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 steps. 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 restart runs are now handled differently. With previous versions, 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 simulator 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.

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) by applying a multiplying factor or an increment.

FORMAT:

*WTMULT | *WTINCR con_type well_list value_list

DEFINITIONS:

well_list

One or more well numbers or quoted well names assigned with the *WELL keyword to which this specification of pressure gradient applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding.

con_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.

value_list

One number for each well identified by *well_list* specifying a multiplying factor or an increment to be applied. The new value of the specified constraint = old value \times 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
```

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:

```
*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
*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 Constraint Multiplier or Increment (Optional)

***GCONPMULT, *GCONPINCR**

PURPOSE:

*GCONPMULT/*GCONPINCR is used to modify existing group production target controls with the use of multipliers/increments.

FORMAT:

GCONPMULT | *GCONPINCR *group_list* *con_type* *value

DEFINITIONS:

group_list

Are the groups to which the following constraint multiplier/increment applies.

con_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.

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 *group_list* *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 *group_list* *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 Constraint Multiplier or Increment (Optional)

***GCONIMULT, *GCONIINCR**

PURPOSE:

*GCONIMULT/*GCONIINCR is used to specify multipliers/increments which modify existing group injection target controls.

FORMAT:

*GCONIMULT | *GCONIINCR *group_list* *con_type* *value*

DEFINITIONS:

group_list

Are the groups to which the following constraint multiplier/increment applies.

con_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.

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 solvent voidage replacement fraction is reduced by half whereas the gas voidage replacement fraction is increased by a factor of 2.

```
*GCONIMULT 'Group1'  
  *VREP *SOLVENT 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 *SOLVENT 0.6  
*GCONI 'Group-1' 'Group-2' *VREP *GAS 0.4
```

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 16 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

One or more well numbers or quoted well names assigned with the *WELL keyword. See '**Using Wildcards in Well Lists**' in the Tutorial Section for more information about using wildcarding. The reporting group will include all wells in the list.

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/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:

$$\begin{aligned} Q_{\text{average}} &= Q_{\text{instant}} * \text{OTF_input} && (\text{relative}) \\ Q_{\text{actual}} &= Q_{\text{instant}} * \text{OTF_actual} && (\text{absolute}) \end{aligned}$$

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

$$\begin{aligned} \text{OTF_actual} &= \text{OTF_input} \\ Q_{\text{actual}} &= Q_{\text{instant}} * \text{OTF_actual} && (\text{absolute}) \end{aligned}$$

for all levels. Moreover, $Q_{\text{actual}} = Q_{\text{instant}}$, 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

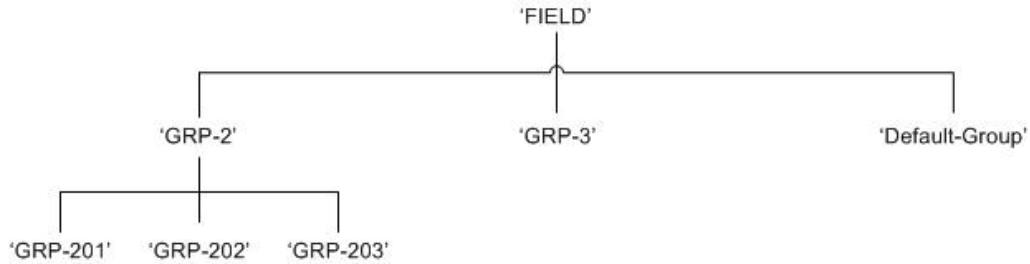
factors are meant to model situations in which all wells in a group are brought down at the same time and restored to service at the same time; however, wells are still allowed to have individually a non-unit on-time factor when the group factors are in force.

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:

Group Hierarchy	On-Time Factor (OTF)		Rates (Q)		
	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 data set 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

Hydraulic Pressure Table (Conditional)

*PTUBE1, *ITUBE1

PURPOSE:

*PTUBE1 or *ITUBE1 introduces 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) ... flo(nflo)

Gfr
gfr(1) ... gfr(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)*
: : : : : :

*DEPTH

This subkeyword introduces input of the reference depth (m | ft | cm).

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 | cm³/min);
*LIQ Liquid (oil+water) production rate (m³/day | stb/day | cm³/min);
*GAS Gas production rate (m³/day | scf/day | cm³/min);
For *ITUBE1, it is identified by one of the following subkeywords:

*GAS Gas injection rate (m³/days | scf/day | cm³/min);

	*WAT	Water injection rate (m^3/day stb/day cm^3/min);
	*SOL	Solvent injection rate (m^3/day scf/day cm^3/min);
Gfr		Independent Gas Fraction variable (*PTUBE1 only), identified by one of the following subkeywords:
	*GOR	Gas-oil ratio (m^3/m^3 scf/stb cm^3/cm^3);
	*GLR	Gas-liquid ratio (m^3/m^3 scf/stb cm^3/cm^3);
	*OGR	Oil-gas ratio (m^3/m^3 stb/scf cm^3/cm^3);
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 (m^3/m^3 bbl/scf cm^3/cm^3);
Add		Independent Additional look-up variable (*PTUBE1 only), identified by one of the following subkeywords:
	*LFG	Lift gas injection rate (m^3/day scf/day cm^3/min);
	*LFR	Injection gas-liquid ratio for gas lift (m^3/m^3 scf/stb cm^3/cm^3);
	*LFT	Total gas-liquid ratio for gas lift (m^3/m^3 scf/stb cm^3/cm^3);
	*DNO	Surface mass density of oil (kg/m^3 lb/ft^3 g/cm^3). It takes the volumetric-weighted average value for a manifold group;
	*DNG	Surface mass density of gas (kg/m^3 lb/ft^3 g/cm^3). 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.
*WHP		This keyword specifies the well-head (or manifold outlet) pressures (kPa psi kg/cm^2).
*BHP		This keyword introduces input of the bottom-hole (or manifold inlet) pressure table (kPa psi kg/cm^2).

***WHT**

Mutually exclusive with *BHP, this keyword introduces input of the well-head temperature table ($^{\circ}\text{C}$ | $^{\circ}\text{F}$ | $^{\circ}\text{C}$) 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 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* 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 well bore 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 in the Add variable item, apply to the manifold groups.

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. 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

```

The acceptable range of values for reference depth is:

	SI m	Field ft	Lab cm
Min	-1.0E+4	-32,808.0	-1.0E+6
Max	1.0E+4	32,808.0	1.0E+6

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 (ioqr)	*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.

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

One or more well numbers or quoted well names assigned with the *WELL keyword to which this specification applies. See ‘**Using Wildcards in Well Lists**’ in the Tutorial Section for more information about using wildcarding. The well list must be on the same line as *NWHYFS and must end with the end of the line. If more wells than can be listed on one line must have NWHYFS specified, a new *NWHYFS line must be entered.

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.

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

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

DEFUALTS:

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.

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 6 under *TRIGGER keyword 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_1_1' on oil production rate greater than 5000.0 the syntax is:

```
*TRIGGER 'trig1' *ON_CTRLLUMP 'c_1_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 *ALL
```

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 SPECIAL line for each quantity in the I/O section of the data file. The syntax is:

```
*OUTSRF *SPECIAL 'lump_name' *CRL RATE *{SURFACE|PROD|INJ}  
{OIL|WATER|4TH|LIQUID|BFH}
```

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 *SPECIAL 'lump_1' *CRL RATE *SURFACE *OIL
```

For light oil (API tracking) assuming this option is turned on the syntax for volumetric rate at STC assuming production stream is desired is

```
*OUTSRF *SPECIAL 'lump_1' *CRL RATE *PROD *4TH
```

For cumulative use

```
*OUTSRF *SPECIAL 'lump_1' *CRLLCUM *SURFACE *OIL
```

For light oil (API tracking) assuming this option is turned on the syntax for volumetric cumulative at STC assuming production stream is desired is

```
*OUTSRF *SPECIAL 'lump_1' *CRLLCUM *PROD *4TH
```

For the gas phase volumetric rate at STC the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLGRATE *{SURFACE|PROD|INJ}
```

For the gas phase volumetric cumulative at STC the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLGCUM *{SURFACE|PROD|INJ}
```

For the solvent phase volumetric rate at STC the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLSLVR *{SURFACE|PROD|INJ}
```

For the solvent phase volumetric cumulative at STC the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLSLVC *{SURFACE|PROD|INJ}
```

For the polymer phase mass rate the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLPLMR *{SURFACE|PROD|INJ}
```

For the polymer mass cumulative the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLPLMC *{SURFACE|PROD|INJ}
```

For gas oil ratio at surface conditions the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLGOR
```

For water cut ratio in percent at surface conditions the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLWCUT
```

For water gas ratio at surface conditions the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLWGR
```

For gas liquid ratio at surface conditions the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLGLR
```

For oil gas ratio at surface conditions the syntax is

```
*OUTSRF *SPECIAL 'lump_1' *CRLOGR
```

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

$i_f \ j_f \ k_f (/ \ ... (/ i_m \ j_m \ k_m) ...)$

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 6 under *TRIGGER keyword.

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 *AYERCLUMP.

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

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

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-LAYERCLUMP removes membership of a layer from the named report lump

FORMAT:

```
*RM-LAYERCLUMP      report_lump_name
'well name 1' layer_id_1
'well name 2' layer_id_2
:
'well name n' layer_id_n
report_lump_name
```

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

$i_f \ j_f \ k_f$ (/ ... (/ $i_m \ j_m \ k_m$) ...)

DEFUALTS:

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_n
  Name of the control lump.

parameter_value_1, parameter_value_n
  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 Versus Actual Rates (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.

Dynamic Grid Amalgamation Control (Optional)

***DYNAGRID, *DYNAGRID-WLN-A, *DYNAGRID-WLN-V**

PURPOSE:

Create or remove amalgamated grid cells and refinements statically.

FORMAT:

***DYNAGRID option
 { $i_1:i_2 \ j_1:j_2 \ k_1:k_2$ }**

where

option = ***AMALGAMATE *INTO $n_{ir} \ n_{jr} \ n_{kr}$ |
 *Derefine | *DEAMALGAMATE | *REREFINE**
DYNAGRID-WLN-A *iwlna
DYNAGRID-WLN-V *ilwnv

DEFINITIONS:

***AMALGAMATE *INTO $n_{ir} \ n_{jr} \ n_{kr}$**

Over the specified block region, amalgamate each regular group of $n_{ir} \times n_{jr} \times n_{kr}$ 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. Please note that well blocks cannot be amalgamated.

***Derefine**

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.

***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.

{ $i_1:i_2 \ j_1:j_2 \ k_1:k_2$ }

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.

DYNAGRID-WLN-A *iwlna

Specify how many blocks away from active wells to allow amalgamation and de-refinement in the areal directions. Integer *iwlna* refers to number of blocks on the finest grid.

DYNAGRID-WLN-V *iwlsv

Specify how many blocks away from active wells to allow amalgamation and de-refinement in the vertical direction. Integer *iwlsv* refers to number of blocks on the finest grid.

DEFAULTS:

All these keywords are optional.

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.

Well Blocks themselves (i.e. grid blocks containing wells) cannot be amalgamated.

None of these keywords may be used with any of the following grid options in the Reservoir Description section: *DUALPOR, *DUALPERM, *SUBDOMAIN and *MINC.

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 a static manner only. The grid alterations are static, in that they are applied immediately at a time specified by a *DATE or *TIME keyword.

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).

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.

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.

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.

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. Data after the *STOP keyword is ignored.

For example, to complete the input-data-file in the normal manner at a specific date, use:

```
:  
*DATE 1997 10 22  
*STOP  
For example, to terminate simulation after 7 years,  
without deleting the remainder of the input-data-file,  
use:  
:  
*TIME 2555.0  
*STOP
```

Appendix A

Theoretical Outline

Four Component Black-Oil Model with Dual Porosity

In this appendix, the governing equations for IMEX are discussed. Section A.1 discusses the conservation equations used, including those for the dual-porosity model, the polymer option and the pseudo-miscible option. Section A.2 describes the equations for the dual-porosity, dual-permeability model. Section A.3 describes the subdomain method for dual-porosity modelling, and the matrix-fracture coefficient. Finally, in section A.4, the independent and dependent variables are given.

This version of IMEX is a four-component dual-porosity model which can be used to simulate miscible flood performances and naturally fractured reservoirs. It is also a three phase adaptive implicit black-oil simulator.

For the dual-porosity model, IMEX uses either the Multiple-Interacting Continua (MINC) approach or the subdomain approach. The reservoir is discretized into a primary grid which represents the interconnecting fractures. The rock mass is represented by secondary grids discretized in a nested format (MINC) or in a layered format (Subdomain) as shown in Figure A.1. Besides the MINC approach and the subdomain approach, a dual-permeability option is also available.

IMEX can be run in two-component (oil-water), three-component (oil-water-gas or oil-water-polymer), or four-component (oil-water-gas-solvent or oil-water-gas-polymer) mode. The model solves the mass conservation equations (A.1)-(A.10). Note that the polymer and solvent equations can not both be solved for the same problem. For the MINC dual porosity option it is assumed that we have nested matrix blocks from 1 to J as shown in Figure A.1. Thus, for the outer-most matrix block, k=J, the matrix subblock equations (A.6) to (A.10), include the matrix-fracture transfer terms. Note that for single-porosity systems, the equations are the same as the fracture equations, (A.1) to (A.5), without the matrix-fracture transfer term, and in terms of matrix (reservoir) variables rather than fracture variables.

Conservation of oil in the fracture:

$$\Delta T_{of}^x \left(\Delta p_{of}^{n+1} - \gamma_{of}^x \Delta D \right) + T_{omf}^x \left(p_{om_J}^{n+1} - p_{of}^{n+1} \right) + q_o^{n+1} - \frac{V_b}{\Delta t} \left[\left(\frac{\phi S_o}{B_o} \right)_f^{n+1} - \left(\frac{\phi S_o}{B_o} \right)_f^n \right] = 0 \quad (A.1)$$

Conservation of water in the fracture:

$$\Delta T_{wf}^x \left(\Delta p_{wf}^{n+1} - \gamma_{wf}^x \Delta D \right) + T_{wmf}^x \left(p_{wm_j}^{n+1} - p_{wf}^{n+1} \right) + q_w^{n+1} - \frac{V_b}{\Delta t} \left[\left(\frac{\phi S_w}{B_w} \right)_f^{n+1} - \left(\frac{\phi S_w}{B_w} \right)_f^n \right] = 0 \quad (\text{A.2})$$

Conservation of gas in the fracture:

$$\begin{aligned} & \Delta T_{of}^x R_{sf}^x \left(\Delta p_{of}^{n+1} - \gamma_{of}^x \Delta D \right) + \Delta T_{gf}^x \left(\Delta p_{gf}^{n+1} - \gamma_{gf}^x \Delta D \right) \\ & + T_{omf}^x R_{smf}^x \left(p_{om_j}^{n+1} - p_{of}^{n+1} \right) + T_{gmf}^x \left(p_{gm_j}^{n+1} - p_{gf}^{n+1} \right) + R_{sf}^x q_o^{n+1} \\ & + q_g^{n+1} - \frac{V_b}{\Delta t} \left[\phi_f^{n+1} \left(\frac{R_{sf} S_{of}}{B_{of}} + E_{gf} S_{gf} \right)^{n+1} - \phi_f^n \left(\frac{R_{sf} S_{of}}{B_{of}} + E_{gf} S_{gf} \right)^n \right] = 0 \quad (\text{A.3}) \end{aligned}$$

Conservation of polymer in the fracture:

$$\begin{aligned} & \Delta T_{wf}^x C_{pf}^x \left(\Delta p_{wf}^{n+1} - \gamma_{wf}^x \Delta D \right) + T_{wmf}^x \left(p_{wm_j}^{n+1} - p_{wf}^{n+1} \right) \\ & + \Delta \left(\frac{\phi_p S_w D_e}{B_w} \right)_f^x \Delta C_{pf}^{n+1} + \left(\frac{\phi_p S_w D_e}{B_w} \right)_{mf}^x \left(C_{pm_j}^{n+1} - C_{pf}^{n+1} \right) \\ & + q_w C_p^* - \frac{V_b}{\Delta t} \left[\left(\frac{\phi_p S_w C_p}{B_w} + \phi A_d \right)_f^{n+1} - \left(\frac{\phi_p S_w C_p}{B_w} + \phi A_d \right)_f^n \right] = 0 \quad (\text{A.4}) \end{aligned}$$

Conservation of solvent in the fracture:

$$\begin{aligned} & \Delta T_{wf}^x R_{ssf}^x \left(\Delta p_{wf}^{n+1} - \gamma_{wf}^x \Delta D \right) + \Delta T_{sf}^x \left(\Delta p_{sf}^{n+1} - \gamma_{sf}^x \Delta D \right) \\ & + T_{wmf}^x R_{smf}^x \left(p_{wm_j}^{n+1} - p_{wf}^{n+1} \right) + T_{smf}^x \left(p_{sm_j}^{n+1} - p_{sf}^{n+1} \right) + R_{ssf}^x q_w^{n+1} \\ & + q_s^{n+1} - \frac{V_b}{\Delta t} \left[\phi_f^{n+1} \left(\frac{R_{ssf} S_{wf}}{B_{wf}} + E_{sf} S_{sf} \right)^{n+1} - \phi_f^n \left(\frac{R_{ssf} S_{wf}}{B_{wf}} + E_{sf} S_{sf} \right)^n \right] = 0 \quad (\text{A.5}) \end{aligned}$$

Conservation of oil in the matrix subblock k:

$$T_{om_{k-1/2}}^x \left(p_{om_{k-1}}^{n+1} - p_{om_k}^{n+1} \right) + T_{om_{k+1/2}} \left(p_{oy}^{n+1} - p_{om_k}^{n+1} \right) - \frac{V_{b_x}}{\Delta t} \left[\left(\frac{\phi S_o}{B_o} \right)_{m_k}^{n+1} - \left(\frac{\phi S_o}{B_o} \right)_{m_k}^n \right] = 0 \quad (\text{A.6})$$

Conservation of water in the matrix subblock k:

$$T_{wm_{k-1/2}}^x \left(p_{wm_{k-1}}^{n+1} - p_{wm_k}^{n+1} \right) + T_{wm_{k+1/2}} \left(p_{wy}^{n+1} - p_{wm_k}^{n+1} \right) - \frac{V_{b_k}}{\Delta t} \left[\left(\frac{\phi S_w}{B_w} \right)_{m_k}^{n+1} - \left(\frac{\phi S_w}{B_w} \right)_{m_k}^n \right] = 0 \quad (\text{A.7})$$

Conservation of gas in the matrix subblock k:

$$\begin{aligned}
 & T_{om_{k-1/2}}^x R_{sm_{k-1/2}}^x \left(p_{om_{k-1}}^{n+1} - p_{om_k}^{n+1} \right) + T_{om_{k+1/2}}^x R_{sz}^x \left(p_{o_y}^{n+1} - p_{om_k}^{n+1} \right) \\
 & + T_{gm_{k-1/2}}^x R_{sm_{k-1/2}}^x \left(p_{om_{k-1}}^{n+1} - p_{gm_k}^{n+1} \right) + T_{gm_{k+1/2}}^x \left(p_{g_y}^{n+1} - p_{gm_k}^{n+1} \right) \\
 & - \frac{V_{b_k}}{\Delta t} \left[\phi_{m_k}^{n+1} \left(\frac{R_s S_o}{B_o} + E_g S_g \right)_{m_k}^{n+1} - \phi_{m_k}^n \left(\frac{R_s S_o}{B_o} + E_g S_g \right)_{m_k}^n \right] = 0 \quad (A.8)
 \end{aligned}$$

Conservation of polymer in the matrix subblock k:

$$\begin{aligned}
 & T_{wm_{k-1/2}}^x C_{pm_{k-1/2}}^x \left(p_{wm_{k-1}}^{n+1} - p_{wm_k}^{n+1} \right) + T_{wm_{k+1/2}}^x C_{pmz}^x \left(p_{wy}^{n+1} - p_{wm_k}^{n+1} \right) \\
 & + \left(\frac{\phi_p S_w D_e}{B_w} \right)_{m_{k-1/2}}^x \left(C_{pm_{k-1}}^{n+1} - C_{pm_k}^{n+1} \right) + \left(\frac{\phi_p S_w D_e}{B_w} \right)_z^x \left(C_{py}^{n+1} - C_{pm_k}^{n+1} \right) \\
 & - \frac{V_{b_k}}{\Delta t} \left[\left(\frac{\phi_p S_w C_p}{B_w} + \phi A_d \right)_{m_k}^{n+1} - \left(\frac{\phi_p S_w D_p}{B_w} + \phi A_d \right)_{m_k}^n \right] = 0 \quad (A.9)
 \end{aligned}$$

Conservation of solvent in the matrix subblock k:

$$\begin{aligned}
 & T_{wm_{k-1/2}}^x R_{ssm_{k-1/2}}^x \left(p_{wm_{k-1}}^{n+1} - p_{wm_k}^{n+1} \right) + T_{wm_{k+1/2}}^x R_{ssz}^x \left(p_{w_y}^{n+1} - p_{wm_k}^{n+1} \right) \\
 & + T_{sm_{k-1/2}}^x \left(p_{sm_{k-1}}^{n+1} - p_{sm_k}^{n+1} \right) + T_{sm_{k+1/2}}^x \left(p_{s_y}^{n+1} - p_{sm_k}^{n+1} \right) \\
 & - \frac{V_{b_k}}{\Delta t} \left[\phi_{m_k}^{n+1} \left(\frac{R_{ss} S_w}{B_w} + E_s S_s \right)_{m_k}^{n+1} - \phi_{m_k}^n \left(\frac{R_{ss} S_w}{B_w} + E_s S_s \right)_{m_k}^n \right] = 0 \quad (A.10)
 \end{aligned}$$

In the above expressions, the symbols have the following meanings:

- A_d polymer adsorption
- B_o oil formation volume factor
- B_w water formation volume factor
- C_p polymer concentration
- C_p^* polymer concentration for well: C_{pf}^{n+1} for producers, input C_p for injectors depth
- D depth
- D_e dispersion coefficient. Input dispersion coefficient times geometric factor as in transmissibilities times transmissibility multiplier
- E_g gas expansion factor
- E_s solvent expansion factor
- p_l pressure in phase 1 = oil, gas, water, solvent

q_l	well rates for phase 1 = oil, gas, water, solvent
R_s	solution gas-oil ratio
R_{ss}	solution solvent-water ratio
S_l	saturation of phase l = oil, gas, water, solvent
T_l	transmissibilities of phase l = oil, gas, water, solvent
t	time
γ	specific weight of phase l = oil, gas, water, solvent
ϕ	porosity
ϕ_p	polymer accessible pore volume

Superscripts

n	old time level
$n+1$	new time level
x	old or new time level

Subscripts

f	fracture
g	gas
J	nested matrices are numbered from 1 to J where J is the outermost matrix block connected to fracture
k	submatrix block index between 1 to J
m	matrix
o	oil
w	water
s	solvent
y	equal to f when $k=J$, it is equal to m_{k+1} when $k \neq J$
z	equal to m_k , m_{k+1} , or f, depending on which is the upstream

The fracture transmissibility T_{of}^x in the direction X for the flow term from i to $i+1$ is:

$$T_{of_{i+1/2}}^x = \left[\frac{\Delta Y \Delta Z}{\Delta X} k_e \right]_{i+1/2} \bullet \left[\frac{k_{ro}}{\mu_o B_o} \right]_{f_k}^x \quad (\text{A.11})$$

where k is i or $i+1$ depending on the potential between the two blocks, such that the mobility is always upwind weighted. k_e is the effective fracture permeability which is $k_f \phi_f$. The matrix-to-fracture transmissibility T_{omf}^x can be written as:

$$T_{om_{j+1/2}}^x = \sigma \left[\frac{A_{m_j}}{d_{m_j}/2} k_{m_j} \right] \bullet \left[\frac{k_{ro}}{\mu_o B_o} \right]_t^x \quad (\text{A.12})$$

where t is equal to f or m_j depending on whether the fracture or the matrix is upstream. The matrix-to-matrix transmissibility T_{om} for the flow term from matrix block j to matrix block $j+1$ can be written as:

$$T_{om_{j+1/2}}^x = \sigma \left[\frac{A_m}{d_m} k_m \right]_{j+1/2} \bullet \left[\frac{k_{ro}}{\mu_o B_o} \right]_{m_k}^x \quad (\text{A.13})$$

σ is the matrix-fracture transfer coefficient which will be discussed later. A_m and d_m are the surface areas and nodal distances for the nested matrix elements. The nodal distance for the innermost matrix element is determined by assuming quasi-steady state flow within the block as described in Warren and Root ("The Behavior of Naturally Fractured Reservoirs," SPEJ, Sept. 1963, pp. 245-253). k_m is the matrix permeability.

The gas saturation is related to the water and oil saturations through the saturation constraint

$$S_g = 1 - S_w - S_o \quad (\text{A.14})$$

or when the pseudo-miscible option is used, the saturation constraint becomes

$$S_g = 1 - S_w - S_o - S_s \quad (\text{A.15})$$

The individual phase pressures are related through the capillary pressures P_{ci} :

$$p_w = p_o - p_{cw} \quad (\text{A.16})$$

$$p_g = p_o + p_{cg} \quad (\text{A.17})$$

The equations are discretized into finite-difference form using first order backward differences in time, and central differences in space with upstream mobilities.

The polymer formulation is further described in Appendix E.

Dual Permeability

The dual-permeability model is an enhancement to the standard dual-porosity model. In this model, the communication between the matrices (the intergranular void space which is also referred to as the primary porosity) is not assumed to be negligible. The differences in terms of interblock communication for the standard dual porosity model, the dual permeability model, and the MINC model are shown in Figure A.2.

The equations for matrix flow (A.6-A.10) need to be modified to include matrix-matrix flow terms as follows:

Conservation of oil in the matrix:

$$\Delta T_{om}^x (\Delta p_{om}^{n+1} - \gamma_{om}^x \Delta D) + T_{omf}^x (p_{of}^{n+1} - p_{om}^{n+1}) - \frac{V_b}{\Delta t} \left[\left(\frac{\phi S_o}{B_o} \right)_m^{n+1} - \left(\frac{\phi S_o}{B_o} \right)_m^n \right] = 0 \quad (\text{A.18})$$

Conservation of water in the matrix:

$$\Delta T_{wm}^x (\Delta p_{wm}^{n+1} - \gamma_{wm}^x \Delta D) + T_{wmf}^x (p_{of}^{n+1} - p_{om}^{n+1}) - \frac{V_b}{\Delta t} \left[\left(\frac{\phi S_w}{B_w} \right)_m^{n+1} - \left(\frac{\phi S_w}{B_w} \right)_m^n \right] = 0 \quad (\text{A.19})$$

Conservation of gas in the matrix:

$$\begin{aligned} & \Delta T_{om}^x R_{sm}^x (\Delta p_{om}^{n+1} - \gamma_{om}^x \Delta D) + \Delta T_{gm}^x (\Delta p_{gm}^{n+1} - \gamma_{gm}^x \Delta D) \\ & + T_{omf}^x R_{smf}^x (p_{of}^{n+1} - p_{om}^{n+1}) + T_{gmf}^x (p_{gf}^{n+1} - p_{gm}^{n+1}) + R_{sm}^x q_o^{n+1} \\ & - \frac{V_b}{\Delta t} \left[\phi_m^{n+1} \left(\frac{R_{sm} S_{om}}{B_{om}} + E_{gm} S_{gm} \right)^{n+1} - \phi_m^n \left(\frac{R_{sm} S_{om}}{B_{om}} + E_{gm} S_{gm} \right)^n \right] = 0 \quad (\text{A.20}) \end{aligned}$$

Conservation of polymer in the matrix:

$$\begin{aligned} \Delta T_{wm}^x C_{pm}^x & \left(\Delta p_{om}^{n+1} - \gamma_{om}^x \Delta D \right) + T_{omf}^x \left(p_{of}^{n+1} - p_{om}^{n+1} \right) \\ & + \Delta \left(\frac{\phi_p S_w D_e}{B_w} \right)_m^n \Delta C_{pm}^{n+1} + \left(\frac{\phi_p S_w D_e}{B_w} \right)_{mf}^x \left(C_{pf}^{n+1} - C_{pm}^{n+1} \right) \\ & - \frac{V_b}{\Delta t} \left[\left(\frac{\phi S_w C_p}{B_w} + \phi A_d \right)_m^{n+1} - \left(\frac{\phi S_w C_p}{B_w} + \phi A_d \right)_m^n \right] = 0 \quad (A.21) \end{aligned}$$

Conservation of solvent in the matrix:

$$\begin{aligned} \Delta T_{wm}^x R_{ssm}^x & \left(\Delta p_{wm}^{n+1} - \gamma_{wm}^x \Delta D \right) + \Delta T_{sm}^x \left(\Delta p_{sm}^{n+1} - \gamma_{sm}^x \Delta D \right) \\ & + T_{wmf}^x R_{ssmf}^x \left(p_{wf}^{n+1} - p_{wm}^{n+1} \right) + T_{smf}^x \left(p_{sf}^{n+1} - p_{sm}^{n+1} \right) + R_{ssm}^x q_w^{n+1} \\ & - \frac{V_b}{\Delta t} \left[\phi_m^{n+1} \left(\frac{R_{ssm} S_{wm}}{B_{wm}} + E_{sm} S_{sm} \right)^{n+1} - \phi_m^n \left(\frac{R_{ssm} S_{wm}}{B_{wm}} + E_{sm} S_{sm} \right)^n \right] = 0 \quad (A.22) \end{aligned}$$

The fracture equations are the same as (A.1-A.5) except that the subscript J is not required.
The matrix transmissibility T_{om}^x in the direction X for the flow term from i to i+1 is

$$T_{om_{i+1/2}}^x = \left[\frac{\Delta Y \Delta Z}{\Delta x} k_m \right]_{i+1/2} \bullet \left[\frac{k_{ro}}{\mu_o B_o} \right]_{m_k}^x \quad (A.23)$$

where k is i or i+1 depending on the potential between the two blocks such that the mobility is always upwind weighted. k_m is the harmonic mean of the matrix absolute permeability.

Subdomain Model

Besides the MINC approach and dual permeability, another enhancement to the standard dual-porosity model is called the subdomain method. In essence, this model discretizes the matrix into a layered format to improve the representation of gravity effects in the matrix-fracture transfer calculation. More details of this method can be found in Fung and Collins, "An Evaluation of the Improved Dual-Porosity Model for the Simulation of Gravity Effects in Naturally Fractured Reservoirs," CIM paper no. 88-39-05 presented at the 39th Annual Technical Meeting of the Petroleum Society of CIM held in Calgary, June 12-16, 1988.

Matrix-Fracture Transfer Coefficient, sigma

The transfer coefficient, σ , in equations (A.12) and (A.13) is calculated as follows:

1. All matrix subdivisions of MINC except the innermost subdivision:

$$\sigma = \sigma_{MINC} = \frac{\Delta X \Delta Y \Delta Z}{L_x L_y L_z} \quad (A.24)$$

where ΔX , ΔY , and ΔZ are the grid block dimensions and L_x , L_y , and L_z are the matrix block dimensions.

2. For the innermost subdivision, or when one-matrix one-fracture system is defined, either (a) Gilman and Kazemi or (b) Warren and Root's approach is used:

- a) Gilman and Kazemi (1983) used the following matrix-fracture transfer coefficient, σ , ("Improvements in Simulation of Naturally Fractured Reservoirs," SPEJ, August 1983, pp. 695-707)

$$k\sigma_{KG} = 4 \left(\frac{k_x}{Lx^2} + \frac{k_y}{Ly^2} + \frac{k_z}{Lz^2} \right) \Delta X \Delta Y \Delta Z \quad (A.25)$$

when substituted into (A.12) and (A.13) give:

$$T_{omf}^x = 4 \left(\frac{k_x}{Lx^2} + \frac{k_y}{Ly^2} + \frac{k_z}{Lz^2} \right) \Delta X \Delta Y \Delta Z \left[\frac{k_{ro}}{\mu_o B_o} \right]_t^x$$

$$T_{om_{j+1/2}}^x = 4 \left(\frac{k_x}{dx^2} + \frac{k_y}{dy^2} + \frac{k_z}{dz^2} \right) dx dy dz \sigma_{MINC} \left[\frac{k_{ro}}{\mu_o B_o} \right]_t^x$$

- b) Warren and Root (1963) assumed quasi-steady state flow and arrived at the following form of shape factors for 3 orthogonal sets of plane fractures with constant matrix absolute permeability in all direction ("The Behavior of Naturally Fractured Reservoirs," SPEJ, Sept. 1963, pp. 245-255).

$$\sigma_{WR} = \frac{20}{3} \left[\frac{1}{Lx} + \frac{1}{Ly} + \frac{1}{Lz} \right]^2 \Delta X \Delta Y \Delta Z \quad (A.26)$$

when substituted into (A.12) and (A.13) give:

$$T_{omf}^x = \frac{20}{3} \left(\frac{1}{Lx} + \frac{1}{Ly} + \frac{1}{Lz} \right)^2 \Delta X \Delta Y \Delta Z k_m \left[\frac{k_{ro}}{\mu_o B_o} \right]_t^x$$

$$T_{om_{j+1/2}}^x = \frac{20}{3} \left(\frac{1}{dx} + \frac{1}{dy} + \frac{1}{dz} \right)^2 dx dy dz \sigma_{MINC} k_m \left[\frac{k_{ro}}{\mu_o B_o} \right]_t^x$$

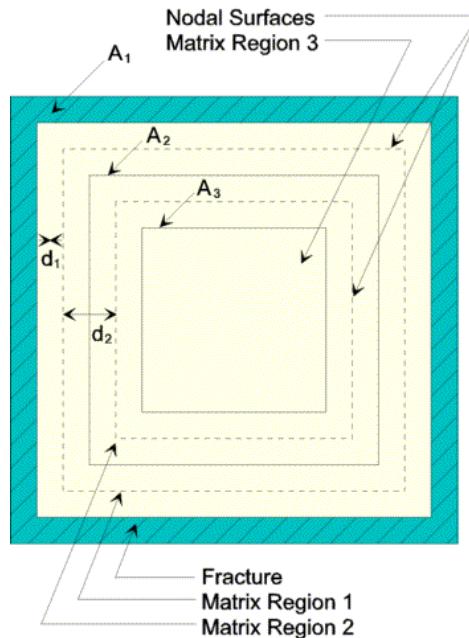
where dx , dy , and dz are the dimensions of the innermost subdivision of MINC.

Independent and Dependent Variables

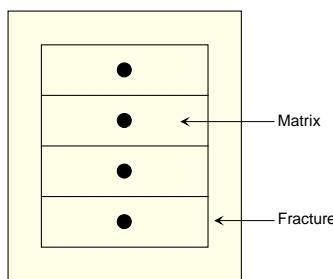
The independent variables at each grid node are oil pressure p_o , water saturation S_w , oil saturation S_o in the presence of free gas when the gas option is on, polymer concentration C_p when the polymer option is on, and solvent saturation S_s in the presence of solvent when the pseudo-miscible option is on. When no free gas is present p_B , saturation (bubble point) pressure, replaces S_o as an independent variable. Similarly, when no free solvent is present p_{Bs} , saturation pressure of solvent in water, replaces S_s as an independent variable. All other variables are functions of these two, three, or four independent variables. The gas saturation is a dependent variable, as can be seen from equation (A.11). The gas, solvent, and polymer options are set using the *MODEL keyword in the component properties section.

Each independent variable is associated with one of the conservation equations. In the two-component mode (oil-water) the oil and water equations are used with p_o and S_o as primary variables. For the three-component modes (oil-water-gas, or oil-water-polymer) the equations are oil, water and gas; or oil, water and polymer with primary variables p_o , S_w , S_o or p_b ; or p_o , S_o , C_p . In the four-component modes, the equations are oil, water, gas and solvent or oil, water, gas and polymer with primary variables p_o , S_w , S_o or p_b , and S_s or p_{bs} ; or p_o , S_w , S_o or p_b , and C_p .

In dual-porosity simulation, the fracture equations are (A.1)-(A.5) and the corresponding variables are p_{of} , S_{wf} , S_o or p_{Bf} , C_{pf} and S_{sf} or p_{Bsf} . The matrix equations are (A.5)-(A.8) and the corresponding variables are p_{om} , S_{wm} , S_o or p_{Bm} , C_{pm} and S_{sm} or p_{Bsm} . If nested matrix blocks are used, each matrix node will be represented by a set of equations (A.6)-(A.10).



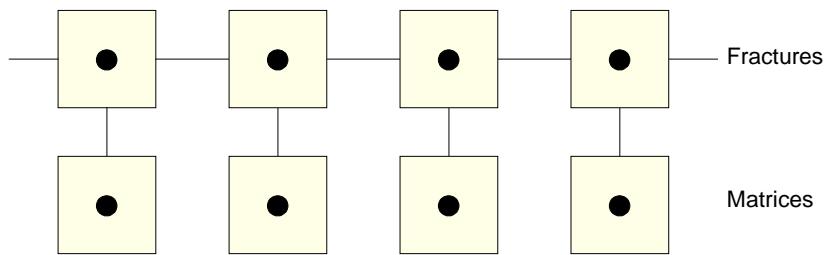
(a): MINC-partitioning



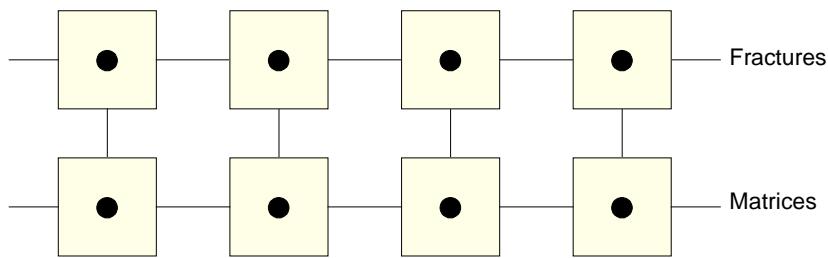
(b): Subdomain partitioning

Figure A-1

(a) Standard dual-porosity model



(b) Dual-permeability model



(c) MINC model

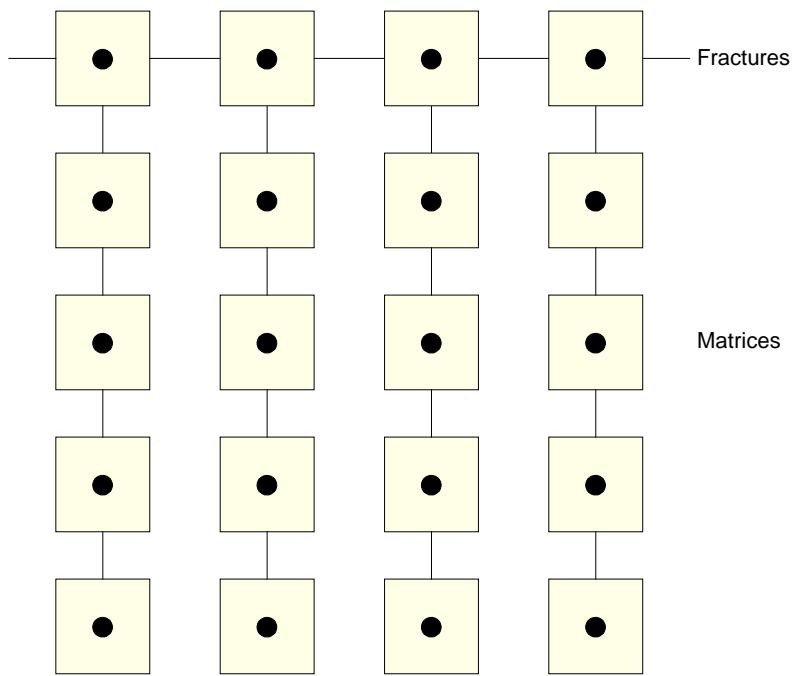


Figure A-2

Schematic diagrams of connectivity for the standard dual-porosity model, the dual-permeability model, and the MINC model

Appendix B

Well Model

There are two types of injection wells available. Each is capable of injecting oil, (free) gas, (free) water, and solvent. An injector well may be mobility weighted or unweighted. There is also a production well model. This accounting leads to nine distinct well types, indicated by the choice of keywords available: *PRODUCER, *INJECTOR, and *INCOMP.

Total Mobility Weighted Injectors

*INJECTOR *MOBWEIGHT

The well rate for the single injected phase is calculated as follows, where 1 denotes oil, water, gas, or solvent.

$$q_1 = \sum_{\text{ily}} [\text{WI} \lambda_T (p_{BH} + \text{head} - p_i) / B_1] \quad j = o, w, g, s$$

Unweighted Injectors

*INJECTOR *UNWEIGHT

The well rate for the single injected phase is calculated as follows, where 1 denotes oil, water, gas, or solvent.

$$q_1 = \sum_{\text{ily}} [\text{WI} (p_{BH} + \text{head} - p_i)]$$

Well Injectivity

The well injectivity is user set with any of the well completion keywords, *GEOMETRY, *PERF, and *PERFV and has the units of (fluid volume)/(time)(pressure). The equation used to calculate well injectivity is:

$$\text{WI} = 2\pi k h f_f h / (\ln(r_e / r_w) + S)$$

*GEOMETRY keywords are used with mobility weighted injectors and producers only. Well indices are entered directly by the user if unweighted injectors are used.

Producers

*PRODUCER

The well rate for the four phases oil (o), gas (g), water (w), solvent (s) are calculated as follows:

$$q_o = \sum_{ily} [WP\lambda_o(p_{BH} + head - p_i)/B_o]$$

$$q_g = \sum_{ily} [WP\lambda_g(p_{BH} + head - p_i)E_g + R_s q_o]$$

$$q_w = \sum_{ily} [WP\lambda_w(p_{BH} + head - p_i)/B_w]$$

$$q_s = \sum_{ily} [WP\lambda_s(p_{BH} + head - p_i)E_s + R_{ss} q_w]$$

The well productivity is calculated using the formula:

$$WP = 2\pi k h f_{fh} / (\ln(r_e / r_w) + S)$$

In the above,

q_l	surface flow rate of phase l=o,g,w
\sum_{ily}	sum over blocks in the well completion
p_{BH}	bottom hole well flowing pressure
head	wellbore head
p_i	block pressure
$\lambda_l = (k_{rl} / \mu_l)$	mobility of phase l=o,w,g,s in the block
$\lambda_T = \lambda_o + \lambda_g + \lambda_w + \lambda_s$	total block mobility
R_s	solution gas/oil ratio
R_{ss}	solution solvent/water ratio

Once a well type has been set, the blocks in the completion must be determined along with the well injectivities and productivities. You may either allow IMEX to calculate these internally by using *GEOMETRY to enter the well geometry elements and use any of the well completion keywords with *GEO, or you use the well completion keywords by themselves and enter your own injectivities and productivities.

Wellhead Pressure Wellbore Model

A simplified wellbore model is included in order to apply wellhead pressure constraints. A single phase wellbore model is used for injectors similar to Aziz et al (1972). For producers a table look-up approach is used.

INJECTORS

A single phase wellbore model is used for injectors.

The equation governing the change in pressure in the direction of flow used for the wellbore model is

$$\Delta p = \Delta P_H - \Delta p_{KE} - \Delta p_F$$

where

$$\Delta P_H = \rho g \Delta z \quad (\text{Hydrostatic head over the length } \Delta z)$$

$$\Delta p_F = \frac{2fv^2\rho}{D} \Delta z \quad (\text{Friction gain over a length } \Delta z)$$

$$\Delta p_{KE} = -\rho v^2 \ln\left(\frac{p_2}{p_1}\right) \quad (\text{Kinetic energy gain})$$

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

PRODUCERS

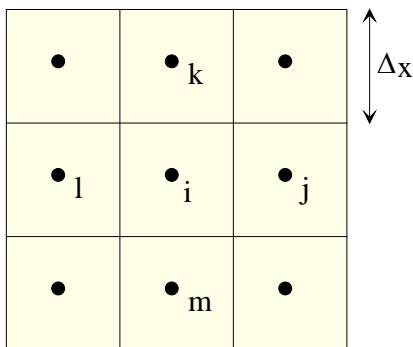
For producers, a table look-up procedure is used. Aziz et al (1972) was not used for producers as the phase behavior and fluid properties in that method are inconsistent with those used within the simulator. Aziz et al used simple analytic correlations while, in general, black oil simulators use isothermal tables to calculate fluid properties and phase behavior. Within the wellbore, however, thermal effects may be very important. Thus for producers, a table look-up is used where bottom hole pressure is calculated as a function of wellhead pressure, oil rate, GOR and water cut. Also, a reference depth is given for each table and a head adjustment is made in order to account for the difference between table depth and the bottom hole well depth.

REFERENCES

Aziz, K., Govier, G.W., and Fogarasi, M., "Pressure Drop in Wells Producing Oil and Gas," JCPT, July-Sept., 1972.

Appendix C

Calculation of Well Parameters



Assume incompressible, single-phase, steady state flow, with no skin effects.

If it is assumed that the pressure calculated at the points j, k, l and m are exact then these pressures may be related to the pressure in the wellbore by the radial flow equation (derived from radial form of Darcy's law).

$$p_j - p_w = [q\mu / 2\pi kh] \ln(\Delta x / \text{rad}) \quad (\text{C.1})$$

where

p_j	=	pressure at node j
p_w	=	pressure at wellbore
q	=	flow rate
μ	=	viscosity
k	=	permeability
h	=	height of grid block
rad	=	wellbore radius

The finite-difference equation for inflow into grid block i (derived from the linear form of Darcy's law) can be written:

$$(p_j - p_i) + (p_k - p_i) + (p_l - p_i) + (p_m - p_i) = q\mu / kh \quad (\text{C.2})$$

Noting that from symmetry considerations $(p_j - p_i) = (p_k - p_i) = (p_l - p_i) = (p_m - p_i)$, then the above can be written as:

$$(p_j - p_i) = q\mu / 4kh \quad (\text{C.3})$$

Eliminating p_j , we obtain

$$p_i - p_w = [q\mu / 2\pi kh] (\ln \Delta x / r_w - \pi / 2) = [q\mu / 2\pi kh] \ln e^{-\pi/2} \Delta x / \text{rad} \quad (\text{C.4})$$

Comparing this to the well model we see that

$$r_e = e^{-\pi/2} \Delta x \approx 0.21 \Delta x$$

Therefore, since

$$r_e = \text{geofac} \sqrt{\Delta x \Delta y / (\text{wfrac} \pi)} = \text{geofac} \Delta x / \sqrt{\text{wfrac} \pi},$$

for this case, geofac $\approx .37$.

Similar arguments show that geofac $\sim .5$ for a radial coning problem.

The appropriate geometric factor can be derived for other geometries using a similar procedure and the method of images. The results for some common cases are shown in Figure C.1. For more details, see Peaceman, 1978 ("Interpretation of Well Block Pressures in Numerical Reservoir Simulation," SPEJ, June 1978, pp. 183-194).

EXAMPLE: WELL INJECTIVITY INDEX

If mobility weighted injectors are used, then the following are required:

k

absolute permeability of the grid block calculated as $k = (k_1 - k_2)^{1/2}$ where k_1 and k_2 are the two absolute permeabilities in the directions normal to the wellbore

h

grid block (formation) thickness

wfrac

well fraction (1 for wells near block centres, 1/2 for half wells on boundaries and 1/4 for wells at block corners)

geofac

geometric factor

r_e

effective well radius is given as $r_e = \text{geofac} (\text{area} / \text{wfrac} \pi)^{1/2}$ where area denotes the area of the block face normal to the wellbore and CC is a geometric factor described in Appendix C

rad

wellbore radius

skin

skin factor

Note that k , h , and r_e are calculated in IMEX once the direction normal to the well is known (set by *GEOMETRY or the set of *PERF keywords). The remaining parameters are user set.

If unweighted injectors are used, then a single injectivity is read and the injectivity is allocated to all layers with weights based on (kh) .

Often an estimate of the formation permeability is available, but no injectivity index has been measured. The well injectivity index can then be estimated as follows.

If a single fluid is injected, and assuming that only irreducible saturations of the other fluids are present in the vicinity of the wellbore, the well flow rate is given by

$$q_1 = \lambda_1 h(p_w - p_i) \frac{wfrac2\pi k}{\left(\ln \frac{r_e}{rad} + skin \right)} \quad (C.5)$$

Converting to surface conditions gives

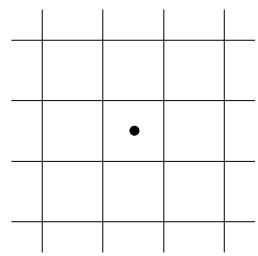
$$q_{ls} = \left(\frac{\rho_1}{\rho_{ls}} \right) \lambda_1 h(p_w - p_i) \frac{wfrac2\pi k}{\left(\ln \frac{r_e}{rad} + skin \right)} \quad (C.6)$$

Hence the well injectivity index is,

$$WI_l = \left(\frac{\rho_1}{\rho_{ls}} \right) \lambda_1 (\tilde{S}_l) h \frac{wfrac2\pi k}{\left(\ln \frac{r_e}{rad} + skin \right)} \quad (C.7)$$

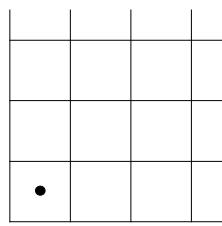
The terms in the above equations can be defined as follows:

ρ_{ls}	surface density of the injected liquid
λ_l	mobility of the injected liquid
\tilde{S}_l	saturation at steady state in the formation near the well
wfrac	well fraction
h	formation thickness
r_e	effective radius = geofac $\sqrt{\Delta x \Delta y / (wfrac \pi)}$
rad	wellbore radius
k	absolute permeability
skin	skin factor
p_w	wellbore pressure
p_i	grid block pressure



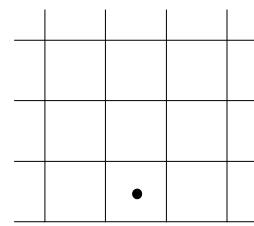
wfrac = 1
geofrac = 0.37

(a) Centre of Square Grid Block



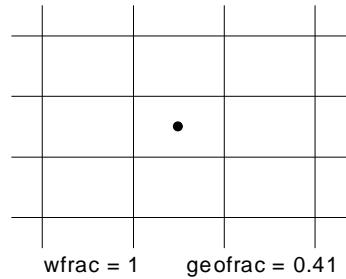
wfrac = 1
geofrac = 0.34

(b) Centre of Corner Block



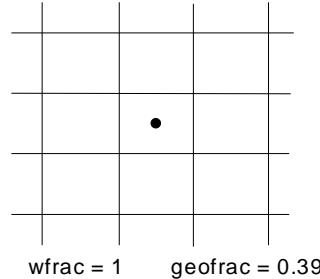
wfrac = 1
geofrac = 0.35

(c) Centre of Edge Block



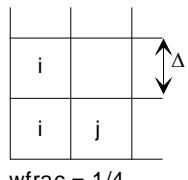
wfrac = 1
geofrac = 0.41

(d) Centre Rectangular Grid Block Ratio $\Delta x/\Delta y=2$



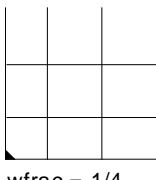
wfrac = 1
geofrac = 0.39

(e) Centre Rectangular Grid Block Ratio $\Delta x/\Delta y=3/2$



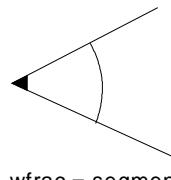
wfrac = 1/4
geofrac = 0.64

(f) Well in Corner



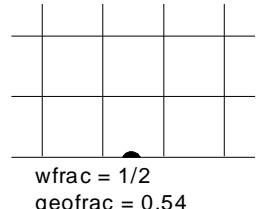
wfrac = 1/4
geofrac = 0.56

(g) Edge Blocks Half Size



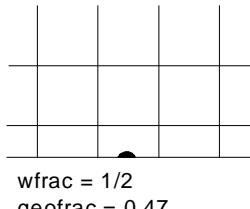
wfrac = segment
geofrac = 0.5

(h) Radial Segment
 2π radian wfrac = 1
 π radian wfrac = 1
 $1/2\pi$ radian wfrac = 1/4



wfrac = 1/2
geofrac = 0.54

(i) Well on Edge



wfrac = 1/2
geofrac = 0.47

(j) Edge Blocks Half Size

Appendix D

Figure D-1a: Numbering of the Grid (Non-Corner Point)

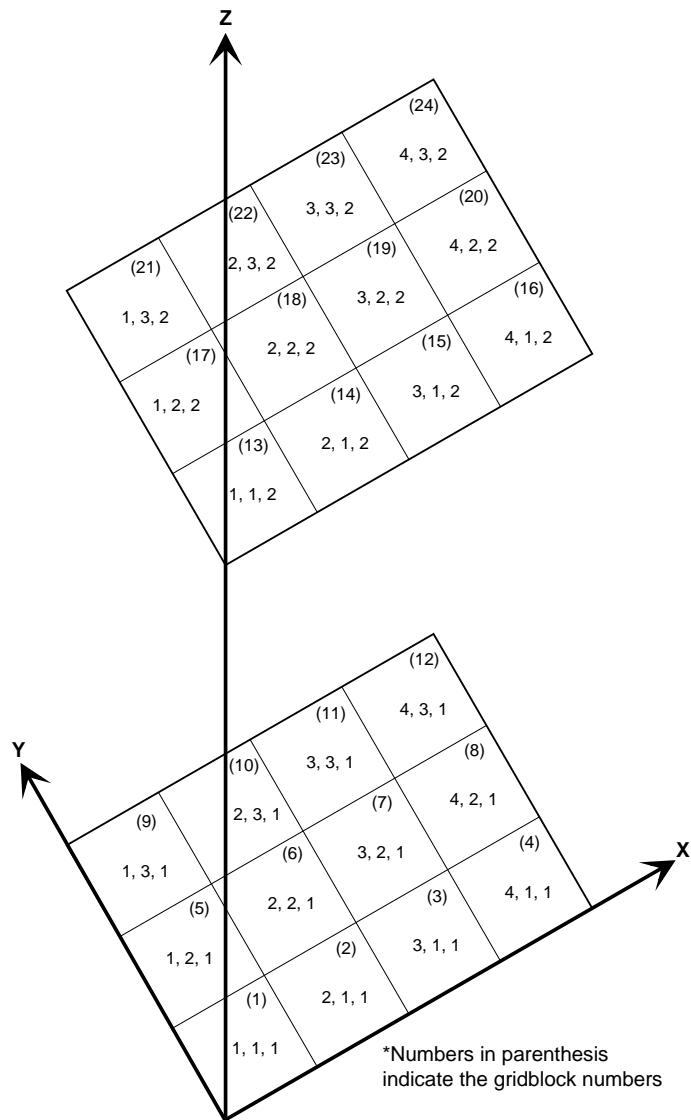


Figure D-1b: Numbering of the Grid (Corner Point)

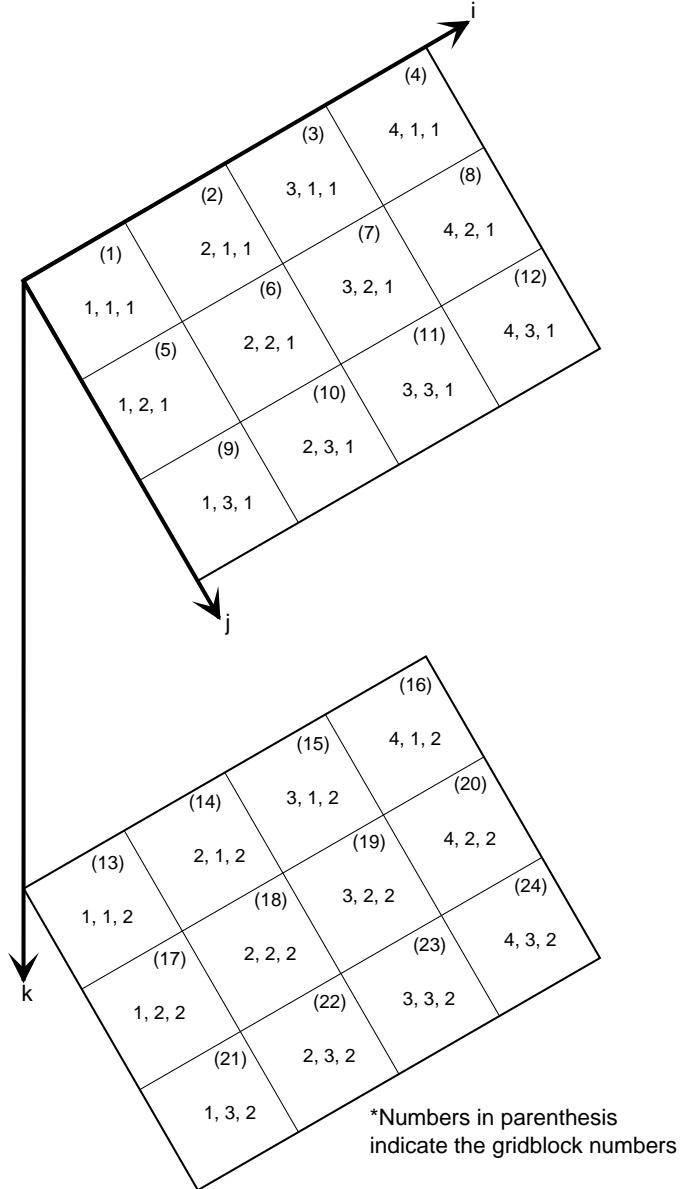


Figure D-2: Cylindrical Reservoir

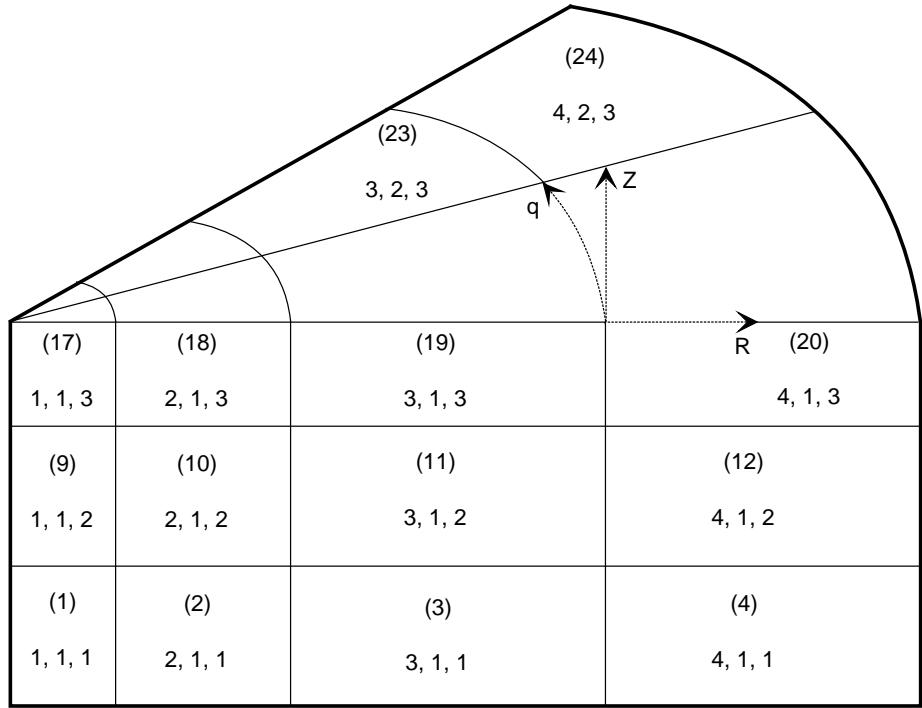
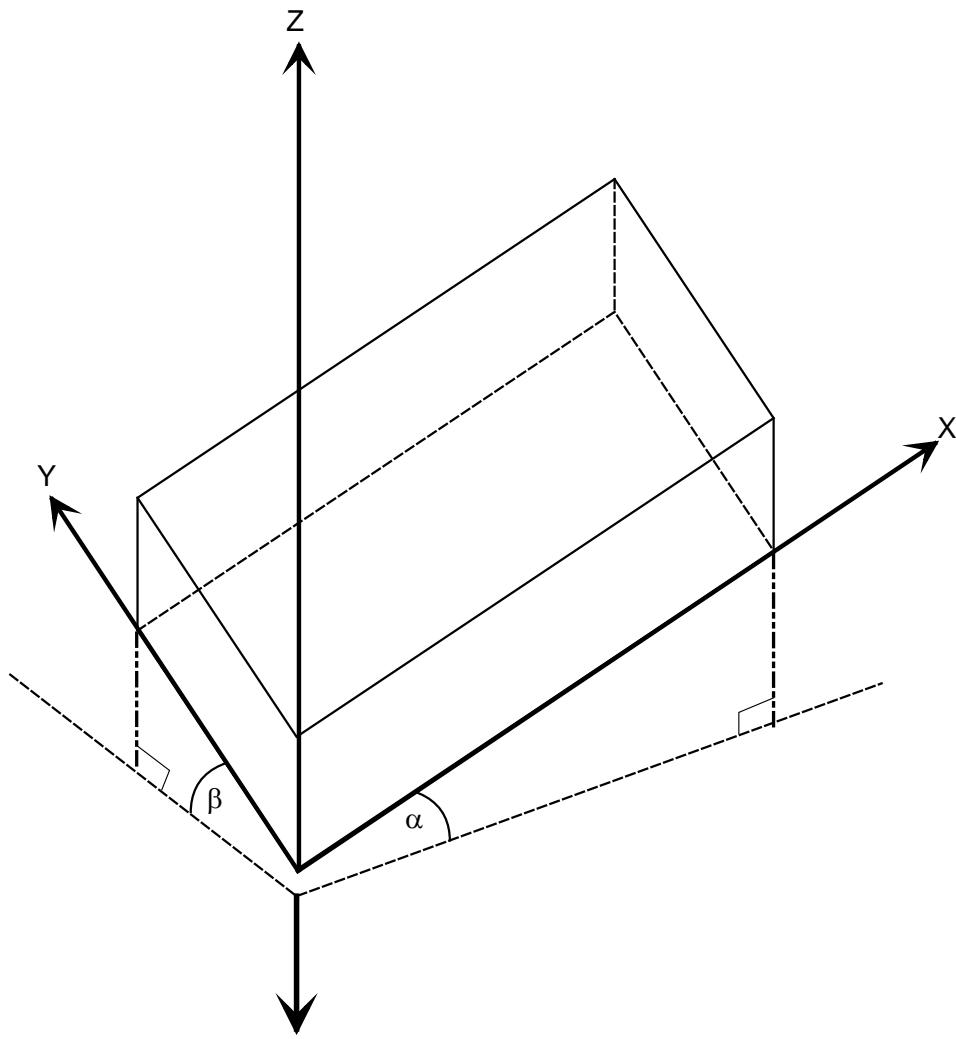


Figure D-3: Calculation of Gravity Components for a Tilted Grid System



α = dip in x - direction

β = dip in y - direction

$GX = -\sin \alpha$

$GY = -\sin \beta$

$GZ = -\sqrt{1 - \sin^2 \alpha - \sin^2 \beta}$

Figure D-4: Radial (Cylindrical) Coordinates

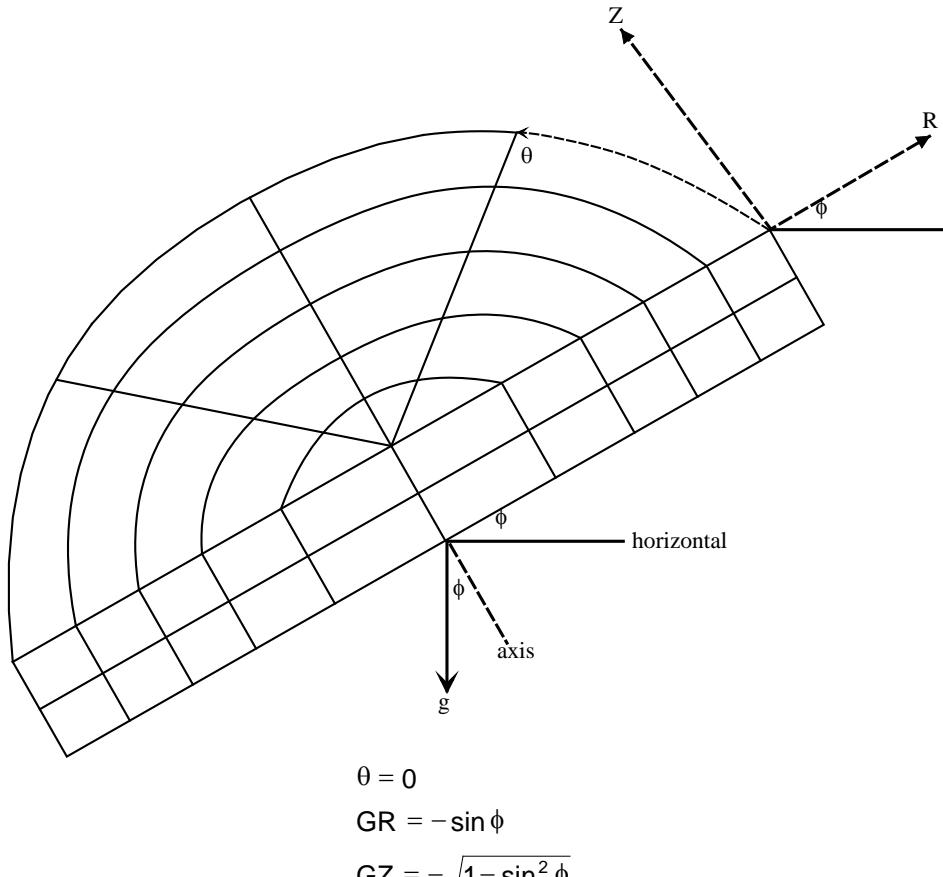
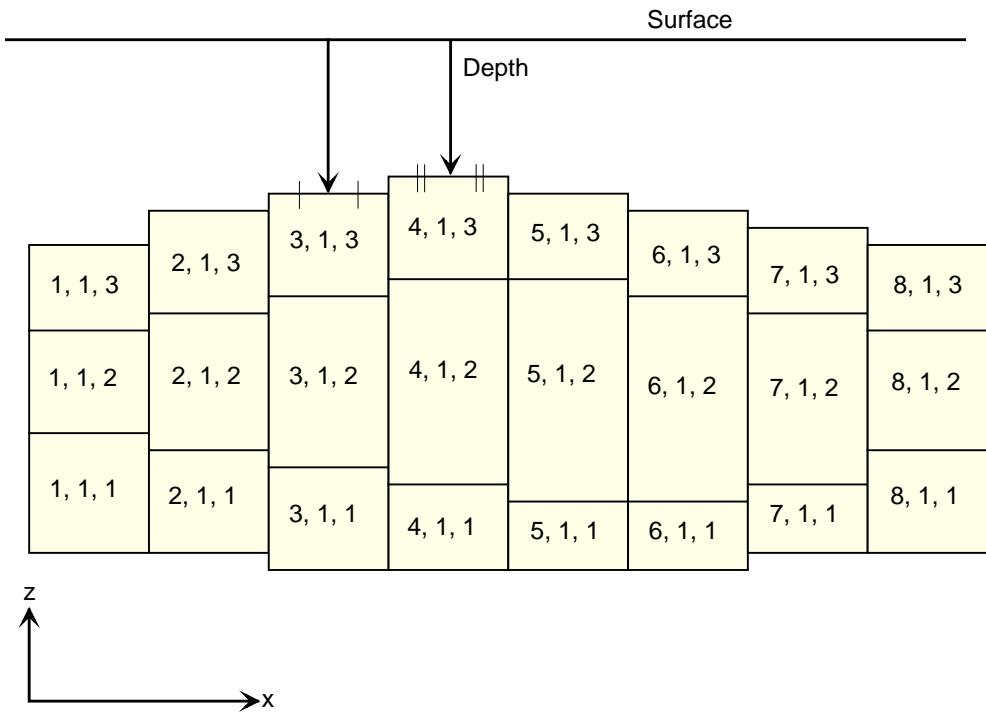


Figure D-5a: Depth Measurement



**Figure D-5b: 3-Dimensional Variable Thickness Grid System
(8x3x2)**

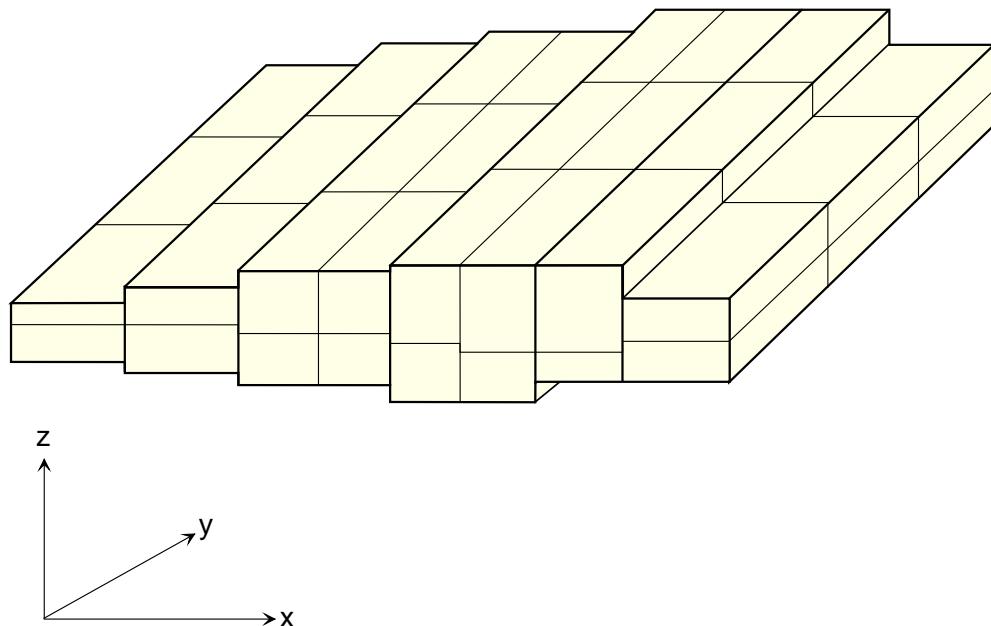


Figure D-6: Fault Throw Definition

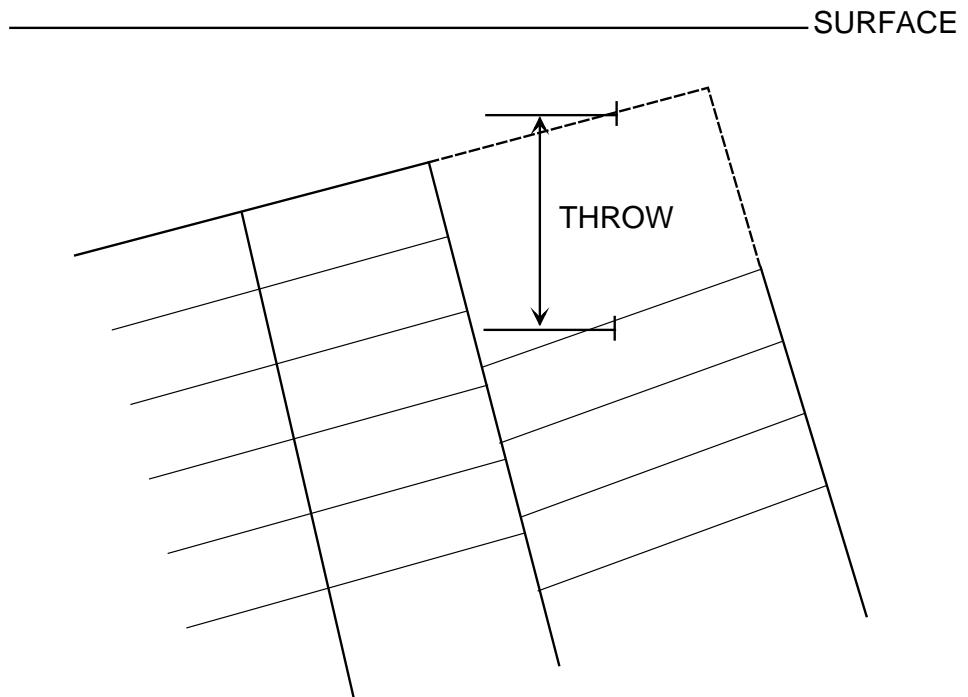


Figure D-7: Typical Data Curves

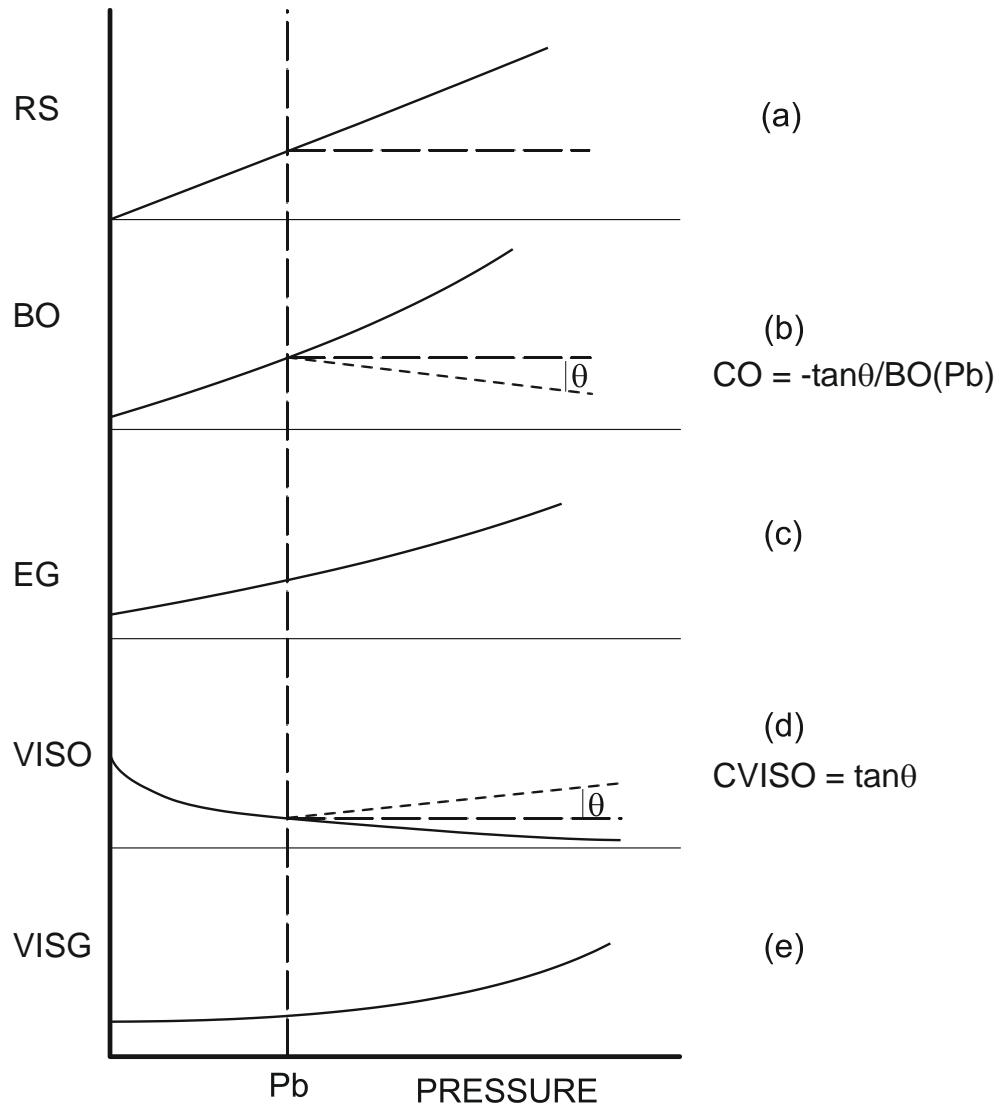
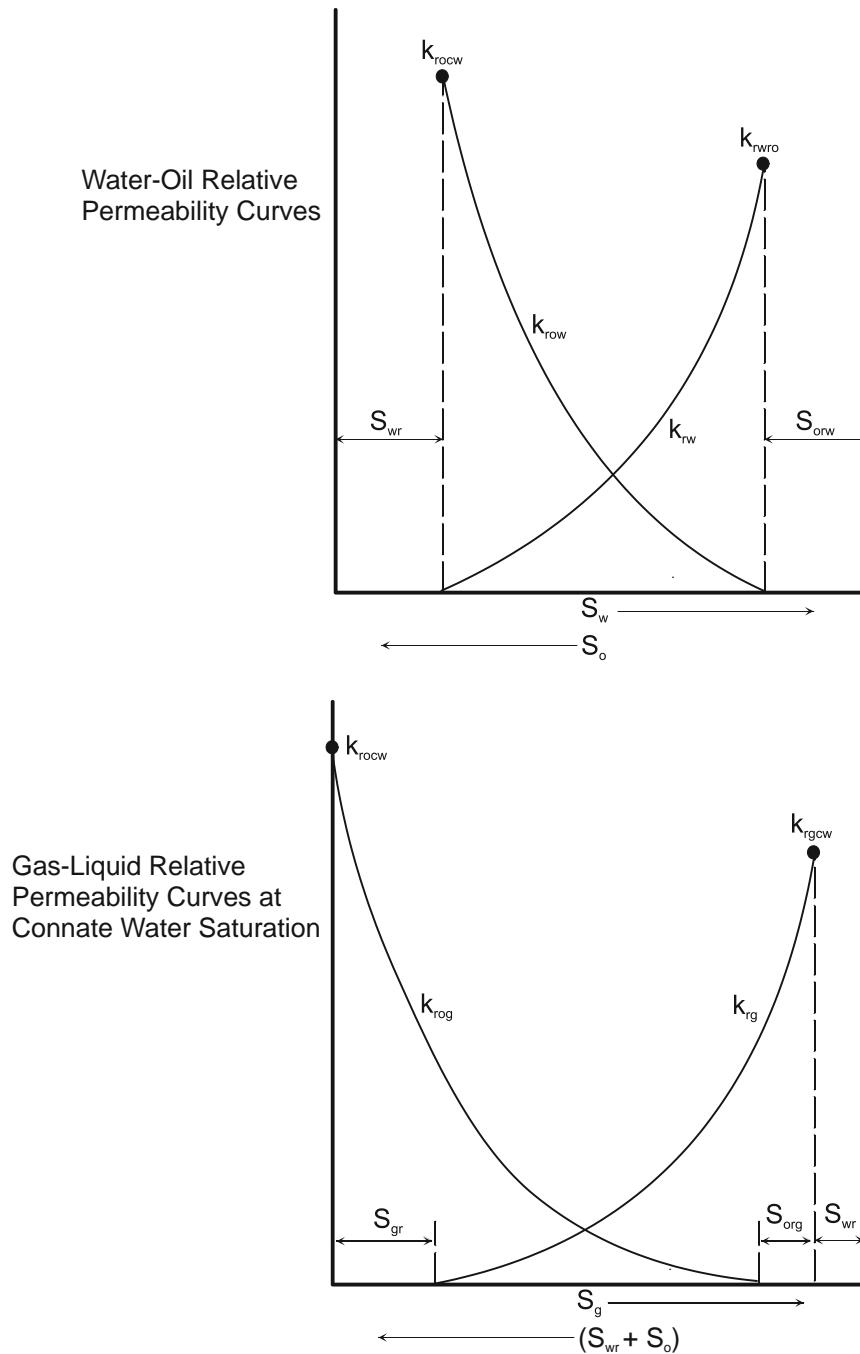


Figure D-8: Typical Relative Permeability Curves



Appendix E

Polymer Option

Conventional waterflooding operations can significantly increase the recovery of hydrocarbons from a reservoir. Due to unfavorable mobility ratios and reservoir heterogeneities, waterflooding can result in poor and incomplete sweeps of the reservoir volume. One method for improving waterflood performance is including additives in the injected water so that the viscosity of the injected water will increase and result in more favorable mobility ratios. The mobility ratio is defined as the mobility of the displacing phase to the mobility of the displaced phase. In waterflooding,

$$M_{w-o} = \frac{\lambda_w}{\lambda_o} = \frac{k_{rw}(S_w)/\mu_w}{k_{ro}(S_{wc})/\mu_o} \quad (E.1.1)$$

where $k_{rw}(S_w)$ is the relative permeability to water at the average water saturation behind the waterfront and $k_{ro}(S_w)$ is the relative permeability to oil ahead of the front at irreducible water saturation.

Several types of additives have been developed for use in waterflood projects. Water-soluble polymers are the most popular choice due to their lower cost compared to other types of additives. In addition to an increase in mobility ratio, other factors and mechanisms have been determined to influence the performance of polymer floods.

Factors and Mechanisms Affecting Polymer Floods

Figure E.1.1 shows a schematic representation of a polymer flood. A slug of polymer solution is injected into the reservoir which has already had a prior injection of low salinity brine or fresh water. The polymer slug is followed by an additional slug of fresh water and by continuous drive water injection. Note that the polymer solution slug is injected between the two fresh (low salinity) water buffers in order to reduce the direct contact with the saline reservoir water, since high water salinity can significantly reduce the viscosity of the polymer solution.

Experimental work from previous studies have shown that polymer flooding improves the recovery over conventional waterflooding by increasing the reservoir volume contacted. The polymer solution injected has no effect on the residual oil saturation. Also, polymer flooding accelerates oil production, resulting in a higher recovery at breakthrough. Polymer flooding is more successful when applied during the early stages of a waterflood process, when the mobile oil saturation is high. A polymer flood will not be useful when applied to uniformly waterflooded fields with low oil viscosity or for fields with high water saturation. Reservoirs

with high permeability variations or rapid water breakthrough are also good candidates for polymer floods but generally pose a high degree of risk.

There are five major aspects of a polymer flood that need to be rigorously represented in a numerical model. They are:

1. Mobility Control
2. Polymer Retention
3. Physical Dispersion
4. Inaccessible Pore Volume
5. Apparent Viscosity and Resistance Factor

These factors are all very important when predicting the performance of a polymer flood, however, the most important benefits are due to the decrease in mobility ratio and the increase in apparent viscosity due to polymer adsorption. The effectiveness of polymer floods is reduced due to dispersion and reservoir permeability variation. The five factors listed above are described in detail below.

Mobility Control

An important aspect of polymer flooding is the improvement of the volumetric sweep efficiency and the increase in the microscopic displacement efficiency. The role of water-soluble polymers is to increase the water viscosity and also to reduce the permeability of the rock to water. Directly, this results in the reduction of the water-oil mobility ratio given in (E.1.1) close to unity or less. Then, the volumetric sweep efficiency will be improved and a higher oil recovery will be achieved at breakthrough compared to conventional waterflooding. The reduction in permeability and higher water viscosity will result in an increase in the resistance to flow. This will divert the polymer solution toward areas unswept by water.

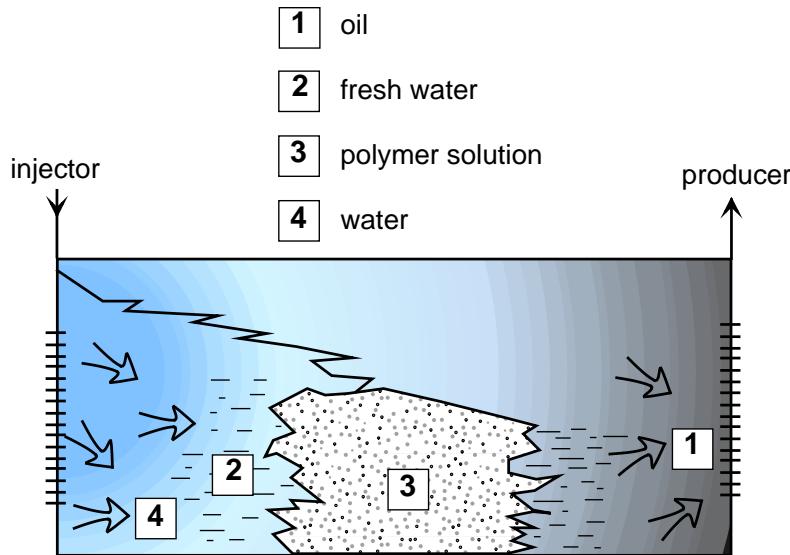


Figure E.1.1: Schematic View of Polymer Flood

The polymer mass balance equation given in (E.1.12) contain a term which represents the water-polymer transmissibility

$$\Delta T_w = \bar{\lambda}_w \left(\frac{A_f}{\Delta \ell} \right) \quad (E.1.2)$$

where A_f is flow area between adjacent grid blocks and $\Delta \ell$ is the distance between grid block centres. The term $\bar{\lambda}_w$ is a modified mobility to water in the presence of polymer defined as:

$$\bar{\lambda}_w = \frac{\bar{k} k_{rw}}{\bar{\mu}_w} \quad (E.1.3)$$

This modified mobility incorporates a reduced permeability \bar{k} due to polymer retention/loss as a result of rock interaction and a modified water viscosity $\bar{\mu}_w$ which accounts for the increase in viscosity of the polymer-water solution.

The reduction in permeability is due to the mechanism of polymer retention. As the polymer solution interacts with the reservoir rock, polymer is adsorbed or desorbed from the rock surface. The variation in rock permeability due to this process is given by

$$\bar{k} = \frac{k}{R_k}$$

where k is the absolute rock permeability prior to polymer flooding and R_k is a permeability reduction factor. R_k is a function of polymer adsorption and the residual resistance factor RRF. R_k is given by the expression

$$R_k = 1.0 + (RRF - 1.0) \frac{A_d}{A_{dmax}} \quad (E.1.4)$$

where A_d is the cumulative adsorption of polymer per unit volume of reservoir rock and A_{dmax} is the maximum adsorptive capacity of polymer per unit volume of reservoir rock. The maximum adsorptive capacity of the reservoir rock and the residual resistance factor are functions of reservoir rock permeability.

The polymer solution viscosity, $\bar{\mu}_w$, is the second parameter for mobility reduction. The modification of the viscosity accounts for the mixing of polymer and water. Note that the resulting viscosity is dependent on the concentration of polymer in the solution. Different mixing rules can be employed to calculate the mixed solution viscosity. Two commonly used mixing models are the Linear-Mixing Model which is given by

$$\bar{\mu}_w = \alpha \mu_p^o + (1 - \alpha) \mu_w \quad (E.1.5)$$

and a Nonlinear Mixing Model which is given by

$$\bar{\mu}_w = \mu_p^{o\alpha} \mu_w^{(1-\alpha)} \quad (E.1.6)$$

In the above expressions (E.1.5) and (E.1.6), μ_p^0 is a reference polymer solution viscosity at a reference polymer concentration. α is a parameter dependent on polymer concentration given as:

$$\alpha = \frac{C_p}{C_p^0}$$

where C_p is the current polymer concentration and C_p^0 is the reference polymer concentration.

Polymer Retention

The process of polymer retention consists of two separate mechanisms. These mechanisms, whose effects are difficult to separate are adsorption of the polymer on rock surfaces and entrapment of polymer molecules in small pore spaces. Both these mechanisms have the effect of increasing the resistance to flow by essentially a permeability reduction. It is difficult to quantify what percentage of injected polymer is adsorbed and what percentage is trapped in small pore spaces since only the produced polymer concentration can be measured. Both these mechanisms result in a loss of polymer to the reservoir. Of the two commonly used polymers for field applications, polyacrylamide polymers adsorb on the surface of most reservoir rock. Polysaccharides are not retained on rock surfaces and consequently, do not exhibit a residual resistance effect. Calcium carbonate has a greater affinity for polymer than does silica and will exhibit a higher adsorption. The porous space within reservoir rock offers a variety of opening sizes. The long chain of the polymer molecule can flow through into large pore openings but can become trapped when the other end has a smaller opening. Entrapment can also take place when the flow is restricted or stopped. When this happens, the polymer molecule loses its elongated shape and coils up. Desorption of the polymer from the reservoir rock can also take place if sufficient polymer has already been adsorbed above a residual sorption level. This residual sorption level is basically a function of the reservoir permeability. Assuming equilibrium sorption with the reservoir rock, the sorption phenomenon can be described as

$$A_d = f(C_p) \quad (E.1.7)$$

where $f(C_p)$ is a function of polymer concentration only. The function $f(C_p)$ is specified in the form of a table. It should be noted that field observations indicate the higher the polymer concentration before flowing through the porous space, the higher will be the adsorption on the rock surface. Consequently, A_d increases with increasing polymer concentration C_p .

Physical Dispersion

Movement of a polymer bank through the porous medium results in mixing, primarily due to physical dispersion. This mixing is characterized by longitudinal and transverse mixing dispersivities which are multiplied by the fluid velocities to give local dispersion coefficients. Molecular diffusion, although not very important, can also be included as a component of the local dispersion coefficient. The effective dispersion coefficient, which is the required input data, can be written by a two-parameter model of the form

$$\bar{D}_e = \frac{D_m}{\tilde{\ell}} + \alpha \bullet \bar{v}_w \quad (E.1.8)$$

Where

\overline{D}_e	=	effective dispersion coefficient
D_m	=	molecular diffusion coefficient
$\tilde{\ell}$	=	tortuosity
$\overline{\alpha}$	=	local dispersion parameter tensor
\overline{v}_w	=	water velocity vector

Note that since fluid motion is normally not parallel to one of the coordinate direction, the dispersion terms form tensor quantities. Because this introduces cross derivatives into the governing flow equations for the polymer solution, (E.1.8) can be simplified by neglecting the off-diagonal terms in the tensoral quantities:

$$D_{e,x,y,z} = \frac{D_m}{\tilde{\ell}} + \alpha_{x,y,z} \frac{|\overline{V}_w|}{\phi}$$

where $D_{e,x,y,z}$ and $\alpha_{x,y,z}$ are the effective dispersion coefficient and local dispersion parameter respectively in the x, y and z directions and \overline{V}_w is the average Darcy velocity.

The major purpose for incorporating dispersive mixing is to simulate the smearing of the trailing and leading edges of the injected polymer bank. The polymer slug propagates through the porous medium at a velocity different from that of the water. Adsorption tends to move the front edge of a polymer slug at a slower velocity than the water bank, and the inaccessible pore volume tends to move the polymer slug at a higher velocity than the water bank. The combination of these two effects results in a smaller slug that is shifted forward. Consequently, further simplification results in a constant value of the effective dispersion coefficient varying only spatially throughout the reservoir. Thus the dispersion becomes indirectly dependent on the rock properties, namely the porosity and absolute permeability.

Inaccessible Pore Volume

When the flow of polymer molecules through the porous medium is restricted in pores with small openings, only the passage of water or brine is permitted. These small openings not contacted by flowing polymer molecules form what is called inaccessible pore volume (IPV). This phenomenon of IPV was first reported by Dawson and Lautz (1972)¹. They showed that all the pore spaces may not be accessible to polymer molecules and that this allows polymer solutions to advance and displace oil at a faster rate than predicted on the basis of total porosity. Up to 30% of the total pore volume may not be accessible to polymer molecules. As a result, the effective porosity for a polymer solution is less than the actual reservoir porosity. A reduced polymer porosity ϕ_p , can be defined to represent the available pore volume to polymer solutions as:

$$\phi_p = (1.0 - \text{IPV})\phi \quad (\text{E.1.9})$$

where ϕ is the original porosity.

The IPV can have beneficial effects on field performance. The rock surface in contact with the polymer solution will be less than the total pore volume, thus decreasing the polymer adsorbed. More importantly, if connate water is present in the smaller pores inaccessible to the polymer, the bank of connate water and polymer-depleted injection water that precedes the polymer bank is reduced by the amount of inaccessible pore volume. One drawback however, is that movable oil located in the smaller pores will not be contacted by the polymer and therefore may not be displaced.

Apparent Viscosity and Resistance Factor

The effect of polymers in aqueous solutions is to increase the viscosity for better mobility control as mentioned earlier. Observations of the behavior of low concentration polymer solutions in a porous medium suggest a much higher viscosity than that measured in the lab. In fact, the viscosity of the polymer solution flowing in the reservoir is 5 to 25 times higher when calculated indirectly using Darcy's law, assuming the same effective permeability. This effect is shown in Figure E.1.2. In reality however, it is the effective permeability of the formation to the polymer solution which is lower than it is to water without polymer. It is difficult to separate the effects of permeability reduction from that of the viscosity increase. The total effect can be measured as a reduction in mobility. The measure of the mobility reduction, R can be expressed as:

$$R = \frac{\lambda_w}{\lambda_p} = \frac{k_{rw} \mu_p}{\mu_w k_{rp}} \quad (E.1.10)$$

where λ_w and λ_p are the water mobility and water-soluble polymer mobility respectively and the other quantities are as defined previously. A plot of the resistance factor, R, as a function of pore volume injected is shown in Figure (E.1.3) for a 300 ppm polymer solution flowing through a core sample. The plot shows that R rapidly increases and stabilizes to a constant value. This behavior is important since injection pressures will not have to be increased in order to sweep the polymer bank through the reservoir. Polymers with high resistance factors are useful for profile improvement by plugging more permeable streaks near injectors and reducing the variation in permeability.

The residual resistance factor R_r measures the reduction of the reservoir rock's permeability to water after polymer flow and is defined as:

$$R_r = \frac{(k_{rw} / \mu_w)_{\text{before polymer flow}}}{(k_{rw} / \mu_w)_{\text{after polymer flow}}} \quad (E.1.11)$$

This residual resistance effect is due to the alteration of the original rock permeability by adsorption of the polymer and mechanical entrapment of polymer molecules. The residual resistance effect also has economic importance. Expenditures for polymer occur only during the injection period. Long after the polymer injection phase, the residual resistance effect continues at no additional expense. It should be noted that polysaccharide polymers are not retained on rock surfaces and consequently do not exhibit the residual resistance effect.

Polymer Types

There are two principal types of polymers that are currently being used. The first type is a synthetic polymer, called hydrolyzed polyacrylamide's (HPAM) are obtained by the polymerization of the acrylamide monomer. Through hydrolysis, some of the acrylamide monomers are converted to carboxylate groups with a negative charge. This process results in a higher molecular weight and a linear chain molecular structure which helps ease the flow through the tortuous porous space of the rock reservoir. Electrolytes in saline water cause the long chain molecules to coil and obstruct the flow and reduce solution viscosity. Hydrolyzed polyacrylamide solutions, generally are sensitive to salts and dissolved solids and must be prepared with fresh water. To be effective, polymer solutions must remain stable for a long time at reservoir conditions. The presence of oxygen can be a source of instability and can result in loss of mobility reduction. Other factors for polymer instability are chemical, mechanical, thermal and microbial degradation. Mechanical degradation is a serious problem for polyacrylamides. The long chain molecules that make up the polymer are not resistant to shear stresses and can be easily broken up, especially at high velocities^{2,3,4}. Mechanical degradation for polyacrylamides becomes more severe in the presence of high salinity, high fluxes in low permeability rock and adverse temperatures. Both chemical and thermal degradation are aggravated under aerobic conditions. Recent studies have shown that biocides, such as formaldehyde need to be used to prevent microbial degradation.

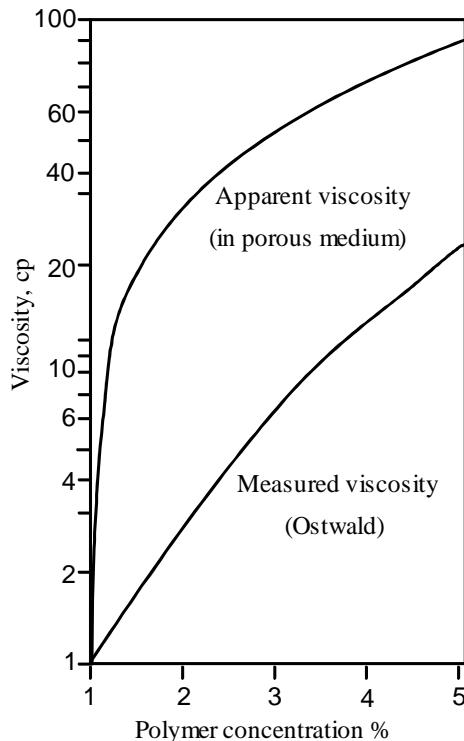


Figure E.1.2: Resistance Effect of Polymer Solution in Porous Media

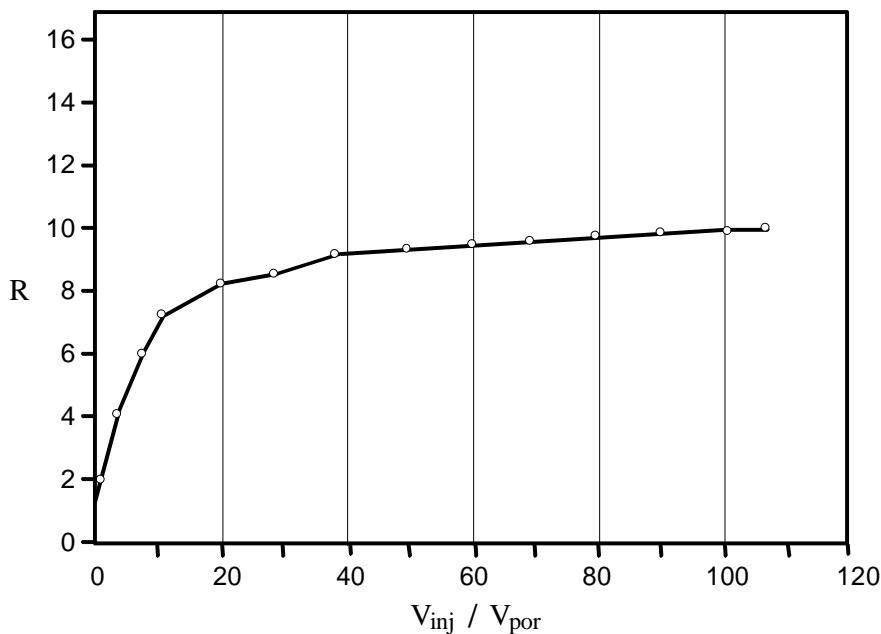


Figure E.1.3

Polysaccharide biopolymers are the second type of polymer commonly used. They are obtained from sugar in a fermentation process caused by the bacterium *Xanthomonas campestris*. The polysaccharide's molecular structure yields a chain of great stiffness which behaves like rigid-rod molecules. Consequently, in contrast to polyacrylamide polymer, the viscosity of a polysaccharide biopolymer solution is not affected by salinity, and shearing effects can be tolerated. Despite these advantages, a polysaccharide biopolymer's stability degrades at temperatures above 95°C (like polyacrylamide polymer) and is more susceptible to bio-degradation. It is also much more expensive than polyacrylamide polymer.

Mass Conservation Equations

Modelling the polymer solution requires correctly describing the increase in viscosity and mobility control as well as the polymer adsorption/desorption⁵. These properties are functions of both reservoir permeability, which varies spatially, and polymer concentration, which varies both spatially and with time. Therefore, a mass conservation is required which describes the flow of polymer solution through the porous media of the reservoir. The mass conservation equation combined with Darcy's law yields:

$$\begin{aligned} & \nabla \cdot (C_p T_w k (\nabla P_w - \gamma_w \nabla h)) + \nabla \left(\frac{\phi_p S_w D_e}{B_w} \right) \cdot \nabla C_p \\ & + q_w C_p = \frac{\partial}{\partial t} \left(\frac{\phi_p C_p S_w}{B_w} + \phi A_d \right) \end{aligned} \quad (E.1.12)$$

where

T_w	=	Transmissibility of the polymer solution
C_p	=	Polymer concentration
P_w	=	Pressure of the water phase
D_e	=	Effective dispersion coefficient
ϕ_p	=	Polymer accessible porosity
A_d	=	Polymer adsorption/desorption
k	=	Absolute permeability
γ_w	=	Water phase specific gravity
q_w	=	Water production/injection rate
S_w	=	Water saturation
B_w	=	Water formation volume factor

This equation can be written in difference form as

$$\Delta T_w^x C_p^x \left(\Delta P_w^{n+1} - \gamma_w^x \Delta D \right) + \Delta \left(\frac{\phi_p S_w D_e}{B_w} \right)^x \Delta C_p^{n+1} + q_w C_p \\ - \frac{V_b}{\Delta t} \left(\left(\frac{\phi_p S_w C_p}{B_w} + \phi A_d \right)^{n+1} - \left(\frac{\phi_p S_w C_p}{B_w} + \phi A_d \right)^n \right) = 0 \quad (E.1.13)$$

Note that the water phase transmissibility is modified in the manner described in equations (E.1.4, E.1.5, E.1.6) to incorporate the effects of increase of mobility due to the presence of the polymer.

Guidelines for Polymer Application

Polymer flooding is of practical use only in reservoirs of a certain characteristic. Due to polymer degradation, application of polymers should be limited to reservoirs with in-situ temperatures less than 95°C. The porosity of the reservoir rock should be medium to high in order to assure a good storage capacity. Also, the salinity of the connate water should be less than 10,000 ppm total dissolved solids in order to assure a stable polymer solution for polyacrylamide polymer. Reservoirs with moderate to good permeabilities are generally good candidates. Permeabilities that are too low will result in very high injection pressures and permeabilities that are high usually give good recoveries with conventional waterfloods. Reservoirs with large variations in permeability are also ideally suited for polymer flooding since the polymer solution has a tendency to divert toward unswept regions of the reservoir.

Polymer flooding is applied most effectively in the early stages of a waterflood while the mobile oil saturation is still high. Waterflooded oil reservoirs can also be good candidates provided that the high values of WOR are caused by either a high water-oil mobility ratio (viscous oil reservoirs) and water coning or by a low vertical sweep efficiency (heterogeneous reservoirs). Under these conditions, the mobile oil saturation will be high and the diverting effect of polymer flooding will be effective in reducing the mobile oil saturation. Polymer flooding, in general, should be avoided for reservoirs with a large gas-cap and/or extensive aquifers. The oil viscosity of candidate reservoirs should not be higher than 200 cp.

References for Polymer Option

1. Dawson, R. and Lautz, R, "Inaccessible Pore Volume in Polymer Flooding," SPE Journal, October 1972.
2. Maerker, J.M., "Mechanical Degradation of Partially Hydrolyzed Polyacrylamide Solutions," SPE Journal, August 1976.
3. Chang, H.L., "Polymer Flooding Technology - Yesterday, Today and Tomorrow," JPT, August 1978.
4. Needham, R.B. and Doe, P.H., "Polymer Flooding Review," JPT, December, 1987.
5. Vela, S., Peaceman, D.W. and Sandvik, E.I., "Evaluation of Polymer Flooding With Crossflow, Retention and Degradation," SPE Journal, April 1976.

Pseudo-Miscible Option

The objective of miscible displacement is to reduce the residual oil saturation through the complete elimination of the interfacial tension (IFT) between oil and the displacing fluid (solvent). This is achieved if oil and the displacing fluid are miscible, i.e. they mix together in all proportions to form one single-phase. Miscibility can be obtained on 'first contact' or through 'multiple contact'¹.

Miscible displacement is characterized by unstable frontal advances, in the form of either viscous fingering or gravity tonguing. These instabilities are caused by the highly adverse viscosity ratio and large density difference that generally exist between oil and the displacing solvent. Figure E.2.1 depicts the swept zone at solvent breakthrough at an adverse mobility ratio of 15, for a five-spot pattern flood as observed in the Hele-Shaw (parallel plate) model. The grid structure in Figure E.2.1 contains 121 grid blocks. In order to reproduce in detail the viscous fingering shown in the figure, the grid block sizes should be in the order of that of a finger width or smaller. Thus, about 10 to 50 times this number of blocks might be required for accurate numerical simulation of this single five-spot displacement.

Todd and Longstaff² proposed a method of simulating miscible displacement performance without reproducing the fine structure of the flow. Their method involves modifying the physical properties and the flowing characteristic of the miscible fluids in a three-phase black-oil simulator. They introduced a mixing parameter ω , which determines the amount of mixing between the miscible fluids within a grid block. A value of zero corresponds to the case of a negligible dispersion rate, whereas a value of one corresponds to complete mixing.

The following sections describe the equations used to form a 'four-component' model in CMG's IMEX simulator. The equations are written in a general form so that immiscible displacement can also be modeled if there is a loss of miscibility.

Mass Conservation Equations

It is assumed that solvent dissolves in water but not in oil. Hence, the mass conservation equation for solvent can be expressed as follows:

$$\begin{aligned} \frac{\partial}{\partial t} \left[\phi \left(\frac{R_{\text{sol}} S_w}{B_w} + E_s S_s \right) \right] + R_{\text{sol}} q_w + q_s \\ - \nabla \left[\frac{R_{\text{sol}} \tau_w}{B_w} k (\nabla p_w - \rho_w \bar{g}) + E_s \lambda_s k (\nabla p_s - \rho_s \bar{g}) \right] = 0 \end{aligned} \quad (\text{E.2.1})$$

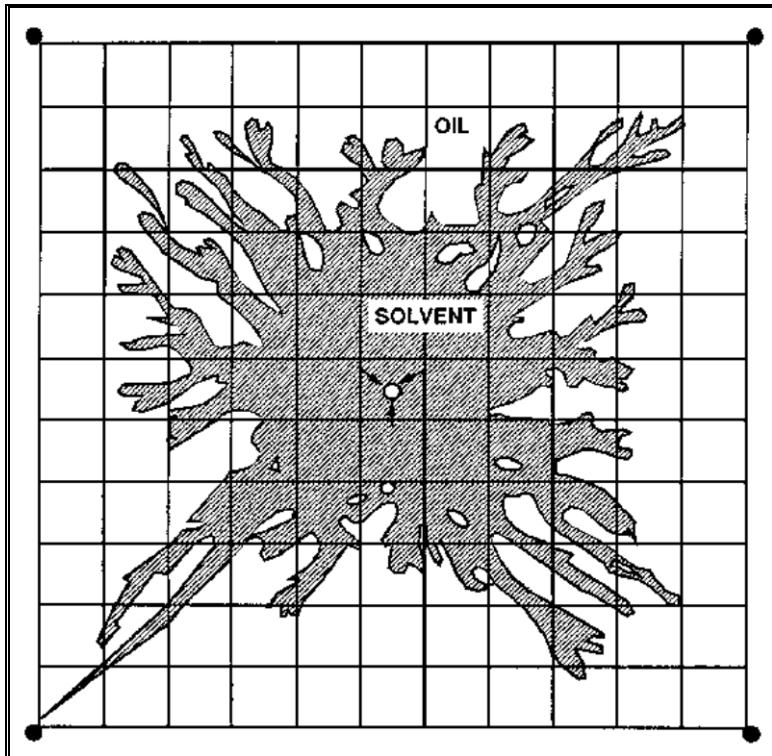


Figure E.2.1: Swept Zone at Solvent Breakthrough (from Todd and Longstaff, 1972)

The chase gas, which is usually injected following a solvent slug, may dissolve in oil but not in water. Thus, the original gas equation in the black-oil model can be used to describe the flow of the chase gas.

Care must be exercised in the PVT data preparation if chase gas is present. R_s is now the solubility of the chase gas in the reservoir at various pressures. E_g and μ_g represent the expansion factor and viscosity of the chase gas. Furthermore, the bubble point pressure for the oil/chase gas system in the reservoir should be set at a low pressure where R_s is zero. For accurate modelling, the block pressures should not be allowed to drop below the MMP.

If there is no chase gas injection, the original gas equation is used to describe the flow of the free gas. In this case, the block pressures can drop below the bubble point pressure and the simulation can still be properly modeled.

Equations for both immiscible and miscible displacements are discussed in the next subsection.

Pseudo-Miscible Considerations

Consider a miscible slug displacement process in which chase gas is injected following a solvent slug. It is assumed the solvent is miscible with the oil and the chase gas. The chase gas, however, is not miscible directly with the oil.

Figure E.2.2 depicts a typical fluid distribution within a grid block. The mixing of solvent and oil is controlled by a pressure-dependent mixing parameter, $\omega_o(P)$, as shown in Figure E.2.3. When the block pressure is so much lower than the minimum miscibility pressure (MMP) that $\omega_o(P) = 0.0$, solvent is displacing oil immiscibly. As the block pressure increases, this mixing parameter reaches its maximum value $\omega_{o\max}$ at the MMP. The maximum value $\omega_{o\max}$, however, cannot be estimated adequately. There is only a limited amount of published material to aid in this estimation. Several authors^{3,4} have attempted to estimate a value for the mixing parameter by history-matching field pilot tests. This history matches mixing parameter is then used judiciously for the predictions of full-scale performance. When no better data is available, the limited work to date suggests a value in the range of 0.5 to 0.7 as a first approximation.

The mixing of solvent and chase gas (or free gas) is governed by ω_g , which is assumed pressure independent. ω_g is bounded by zero and one. Since solvent/gas has a lower mobility ratio than oil/solvent, ω_g is usually greater than $\omega_{o\max}$. Notice that in Figure E.2.2 it is possible for the oil/solvent and solvent/gas dispersion zones to merge, thereby miscibility is lost. This situation is assumed to have occurred when the local solvent saturation drops below a certain value (say, $S_{smin} = 0.01$). The model assumes that the remaining solvent is immobile and the oil and gas are treated as immiscible phases.

The presence of water may block the solvent from the oil, preventing the complete displacement of oil by solvent even under miscible conditions⁵. It has been observed that the water-blocking effect is represented by a function relating water saturations and irreducible oil saturations.

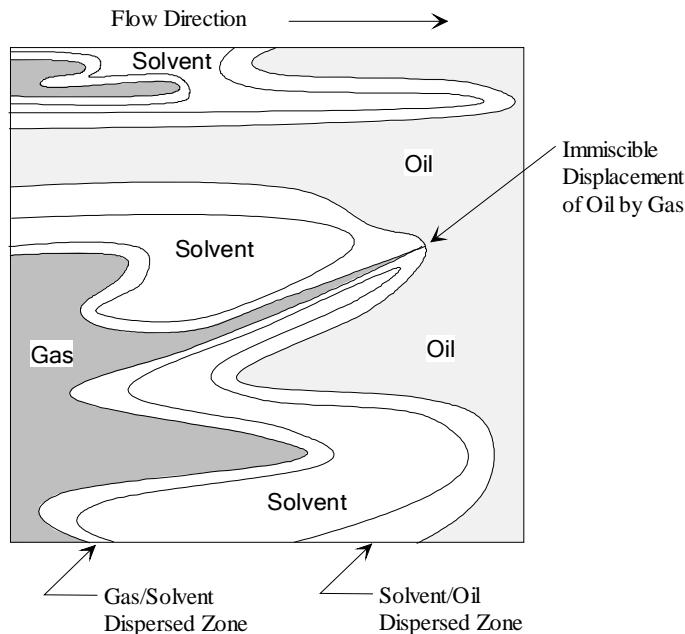


Figure E.2.2: Schematic Representation of a Three-Component Miscible Displacement in a Grid Block (from Todd and Longstaff, 1972)

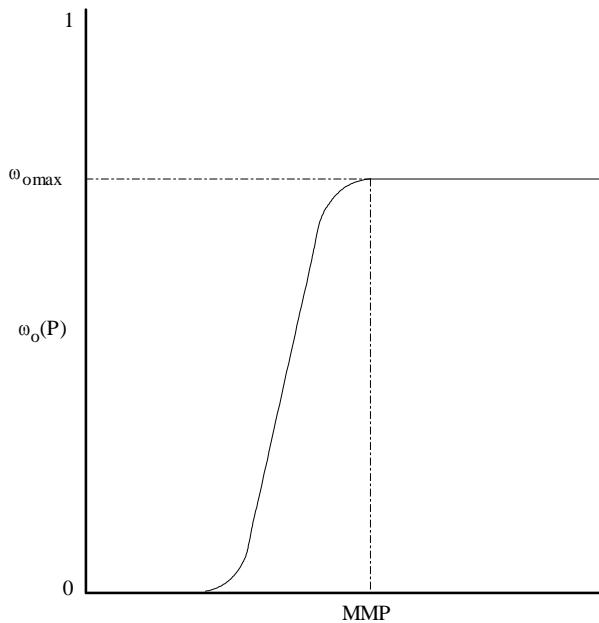


Figure E.2.3: WI Versus P

Relative Permeabilities Modifications

The relative permeabilities of the three hydrocarbon phases are divided into two parts, one for miscible flow and a second for immiscible flow. For each phase the effective relative permeability is calculated as

$$k_{rp}^{\text{eff}} = \frac{\omega_o(P)}{\omega_{o\max}} k_{rp}^m + \left(1 - \frac{\omega_o(P)}{\omega_{o\max}}\right) k_{rp}^{Im} \quad (\text{E.2.2})$$

The miscible portion of the effective relative permeability k_{rp}^m is calculated as:

$$k_{rp}^m = F_p^m k_{row}(S_w) \quad p = \text{oil, gas, solvent} \quad (\text{E.2.3})$$

where F_p^m is a miscible partitioning function which acts to partition $k_{row}(S_w)$ according to the saturation of each hydrocarbon phase weighted by each phase's mixing parameter ω_p .

The immiscible portion of the effective relative permeability k_{rp}^{Im} is calculated as

$$k_{ro}^{Im} = k_{ro}(S_w, S_L) \quad p = \text{oil}$$

where $S_L = S_o + S_w$ and k_{ro} is the immiscible three phase relative permeability.

$$k_{rp}^{Im} = F_p^{Im} k_{rg}(S_g + S_{sol}) \quad p = \text{gas, solvent}$$

where F_p^{Im} is an immiscible partitioning function which acts to partition $k_{rg}(S_g + S_{sol})$ according to the saturations of gas and solvent.

MISCIBLE PARTITIONING FUNCTION

The miscible partitioning function F_p^m is calculated as follows:

$$\text{Define } \omega_p^{\text{Frac}} = \frac{\omega_p}{\omega_{p\max}} \quad p = \text{oil, gas, solvent}$$

ω_o is a function of pressure and $\omega_{p\max}$ is defined as the maximum value of ω_p .

(The values of ω can vary from $\omega_{p\max}$ either due to pressure or due to the loss of miscibility as S_{sol} approaches the minimum solvent saturation.)

We obtain

$$F_o^m = \frac{\omega_o^{\text{Frac}} S_o^*}{\omega_o^{\text{Frac}} S_o^* + \omega_g^{\text{Frac}} S_g + \omega_{\text{sol}}^{\text{Frac}} S_{\text{sol}}} \quad (\text{E.2.4})$$

$$S_o^* = S_o - S_{\text{orm}}(S_w)$$

$$F_g^m = \frac{\omega_g^{\text{Frac}} S_g}{\omega_o^{\text{Frac}} S_o^* + \omega_g^{\text{Frac}} S_g + \omega_{\text{sol}}^{\text{Frac}} S_{\text{sol}}} \quad (\text{E.2.5})$$

$$F_{\text{sol}}^m = 1 - F_o^m - F_g^m$$

$\omega_{\text{sol}}^{\text{Frac}}$ is not entered as input in the model but is calculated as a function of the gas and oil parameters:

$$\omega_{\text{sol}}^{\text{Frac}} = \left(\frac{\omega_o^{\text{Frac}} S_o^*}{\omega_o^{\text{Frac}} S_o^* + \omega_g^{\text{Frac}} S_g} \right) \omega_o^{\text{Frac}} + \left(\frac{\omega_g^{\text{Frac}} S_g}{\omega_o^{\text{Frac}} S_o^* + \omega_g^{\text{Frac}} S_g} \right) \omega_g^{\text{Frac}} \quad (\text{E.2.6})$$

IMMISCIBLE PARTITIONING FUNCTION

The immiscible partitioning function F_p^{Im} $p = \text{gas, solvent}$ is calculated as:

$$F_g^{\text{Im}} = \left(\frac{S_g}{S_g + S_{\text{sol}}} \right)$$

$$F_{\text{sol}}^{\text{Im}} = 1.0 - F_g^{\text{Im}}$$

CAPILLARY PRESSURE

The capillary pressure between oil and gas is determined as:

$$P_{\text{cog}}^{\text{eff}} = \frac{\omega_o}{\omega_{o\max}} P_{\text{cog}}(S_g) + \left(1 - \frac{\omega_o}{\omega_{o\max}} \right) P_{\text{cog}}(S_g + S_{\text{sol}}) \quad (\text{E.2.7})$$

Density Modification

The mixture densities are functions of pure component densities and represented by the following expressions.

$$\rho_p^{\text{eff}} = \omega_p \rho_p^m + (1 - \omega_p) \rho_p^{\text{Im}} \quad p = \text{oil, gas, solvent}$$

$$\rho_o^m = \left(\frac{S_o^*}{S_o^* + S_{\text{sol}}} \right) \rho_o + \left(\frac{S_{\text{sol}}}{S_o^* + S_{\text{sol}}} \right) \rho_{\text{sol}} \quad (\text{E.2.8})$$

$$\rho_g^m = \left(\frac{S_g}{S_g + S_{\text{sol}}} \right) \rho_g + \left(\frac{S_{\text{sol}}}{S_g + S_{\text{sol}}} \right) \rho_{\text{sol}} \quad (\text{E.2.9})$$

$$\begin{aligned} \rho_{\text{sol}}^m = & \left(\frac{\omega_o S_o^*}{\omega_o S_o^* + \omega_g S_g + \omega_{\text{sol}} S_{\text{sol}}} \right) \rho_o + \left(\frac{\omega_g S_g}{\omega_o S_o^* + \omega_g S_g + \omega_{\text{sol}} S_{\text{sol}}} \right) \rho_g \\ & + \left(\frac{\omega_{\text{sol}} S_{\text{sol}}}{\omega_o S_o^* + \omega_g S_g + \omega_{\text{sol}} S_{\text{sol}}} \right) \rho_{\text{sol}} \end{aligned} \quad (\text{E.2.10})$$

$$\omega_{\text{sol}} = \left(\frac{\omega_o S_o^*}{\omega_o S_o^* + \omega_g S_g} \right) \omega_o + \left(\frac{\omega_g}{\omega_o S_o^* + \omega_g S_g} \right) \omega_g \quad (\text{E.2.11})$$

ρ_p^{Im} are the pure component densities $S_o^* = S_o - S_{\text{orm}}(S_w)$.

Viscosity Modification

The mixture viscosities are determined by a $1/4$ power fluid mixing rule and are represented by the following expressions.

$$\mu_p^{\text{eff}} = (\mu_p^m)^{\omega_p} \times (\mu_p^{\text{Im}})^{1-\omega_p} \quad \text{when } p = \text{oil, gas, solvent}$$

$$\mu_o^m = \frac{\mu_o \mu_{\text{sol}}}{\left(\left(\frac{S_o^*}{S_o^* + S_{\text{sol}}} \right) \mu_{\text{sol}}^{1/4} + \left(\frac{S_{\text{sol}}}{S_o^* + S_{\text{sol}}} \right) \mu_o^{1/4} \right)^4} \quad (\text{E.2.12})$$

$$\mu_g^m = \frac{\mu_g \mu_{\text{sol}}}{\left(\left(\frac{S_g}{S_g + S_{\text{sol}}} \right) \mu_{\text{sol}}^{1/4} + \left(\frac{S_{\text{sol}}}{S_g + S_{\text{sol}}} \right) \mu_g^{1/4} \right)^4} \quad (\text{E.2.13})$$

$$\mu_{\text{sol}}^m = \frac{\mu_g \mu_{\text{sol}} \mu_o}{\left(\frac{\omega_o S_o^*}{S_n} \right) \mu_{\text{sol}}^{1/4} \mu_g^{1/4} + \left(\frac{\omega_g S_g}{S_n} \right) \mu_o^{1/4} \mu_{\text{sol}}^{1/4} + \left(\frac{\omega_{\text{sol}} S_{\text{sol}}}{S_n} \right) \mu_o^{1/4} \mu_g^{1/4}} \quad (\text{E.2.14})$$

where

$$\omega_{\text{sol}} = \left(\frac{\omega_o S_o^*}{\omega_o S_o^* + \omega_g S_g} \right) \omega_o + \left(\frac{\omega_g S_g}{\omega_o S_o^* + \omega_g S_g} \right) \omega_g \quad (\text{E.2.15})$$

and

$$S_n = \omega_o S_o^* + \omega_g S_g + \omega_{\text{sol}} S_{\text{sol}}, \quad S_o^* = S_o - S_{\text{orm}}(S_w),$$

μ_p^{Im} are the pure component viscosities.

Estimation of the Mixing Parameter for Oil and Gas Above the Minimum Miscibility Pressure

Oil:

For the oil phase mixing parameter it is recommended that the Koval formula be used. This would represent a realistic maximum value of ω_o .

$$\omega_o = 1 - 4 \log(0.78 + 0.22 M^{1/4}) / \log(M) \quad (\text{E.2.16})$$

$M = \mu_{\text{oil}} / \mu_{\text{solvent}}$ at reservoir conditions

Gas:

We normally recommend a gas mixing parameter equal to or greater than ω_o . Equation (E2.16) can be used with $M = \mu_{\text{solvent}} / \mu_{\text{gas}}$ to estimate ω_g . This would result in gas mixing parameters very close to 0.78. A value of 1.0 would imply that the solvent and gas are completely miscible with no gas-solvent fingering.

LOSS OF MISCELLITY

In the IMEX model we allow the mixture to lose miscibility when the gas saturation is above an input threshold value (*SGTHRESH) and the solvent saturation is lower than an input minimum solvent saturation (*MINSS).

As IMEX is an adaptive implicit simulator and can be subject to numerical problems when abrupt changes occur, we apply a smoothed transition (*MINSS *SMOOTHEND) between the miscible and the immiscible range of solvent and gas saturations.

Pressure Dependence of Miscibility

ω_o is considered to be a function of pressure and is entered as such a function on the PVTS keyword. ω_g is assumed to be independent of pressure.

References for Pseudo-Miscible Option

1. Stalkup, F.I. Jr., "Miscible Displacement", First Printing, SPE Monograph, 1983.
2. Todd, M.R. and Longstaff, W.J., "The Development, Testing and Application of a Numerical Simulator for Predicting Miscible Flood Performance", Trans. AIME 253 (1972) pp. 874.

3. Bilhantz, H.L. et al, "A Method for Projecting Full-Scale Performance of CO₂ in the Willard Unit", paper SPE 7051 presented at the 1978 SPE Symposium on Improved Methods for Oil Recovery, Tulsa, April 16-19.
4. Youngren, G.K. and Charlson, G.S., "History Match Analysis of the Little Creek CO₂ Pilot Test", J. Pet. Tech., November 1980, pp. 2042-52.
5. Raimondi, P. and Torcaso, M.A., "Distribution of the Oil Phase Obtained Upon Imbibition of Water", SPEJ, March 1964, pp. 49-55.

Appendix F

The Use of IMEX in Well Testing Applications

INTRODUCTION

This document describes some guidelines for the use of IMEX in simulating well tests. Such tests typically call for a well to be operated at a sequence of different flowing conditions, usually involving high rate withdrawals followed by shut in periods, under careful pressure monitoring. This monitoring is done with (electronic or mechanical) down-hole gauges placed in the wellbore of the well undergoing the test, and sometimes in the wellbores of selected neighboring wells. The single well test is most often encountered (no neighbor monitoring, or no neighbors), although multiwell interference testing is also done. The resulting down-hole pressure history is analyzed to obtain an indication of the reservoir's character. The analysis hopes to recover at least the reservoir's permeability-thickness product as well as other information regarding well properties (such as damage and the presence of near wellbore fracturing). Certain reservoir parameters, such as the Warren and Root model's LAMBDA and OMEGA parameters for multiple porosity reservoirs, are also sought. The reservoir analysis is carried out with the aid of specialized graphical presentations. These graphical techniques have been well developed over time and are derived from the consideration of simplified reservoir and wellbore models.

Well test modelling assumes that a single phase fluid flows according to Darcy's law, and with known PVT properties, though a reservoir into a well. The reservoir is assumed homogeneous (or is sometimes assumed to be made up of homogeneous layers) of known porosity and thickness. The well may be damaged or stimulated (that is, may have positive or negative skin) and may possess some non-trivial volume (which leads to wellbore storage effects (after flow)). The reservoir may also manifest certain special features, such as a massive hydraulic fracture or multiple, connected porosities. Certain types of reservoir boundaries can be examined. The goal of well testing is to determine the numerical values of the parameters that characterize these phenomena in the well test model.

A black-oil reservoir simulator like IMEX can also model the above types of phenomena, as well as much more general situations. For instance, heterogeneous reservoir properties, reservoir zonation, multiphase flow effects (perhaps in the wellbore), realistic well perforation and operating strategies, and complete descriptions of reservoir geometry can be addressed by a simulator. These latter effects can only be hinted at in a conventional well test analysis. This document will describe how to prepare an IMEX data set so that IMEX can replicate the phenomena observed during a conventional well test.

Both well test software and IMEX start with the full equations describing the flow of fluids in a reservoir, and both types of software take the PVT description of the fluid into account. However, well test software then turns to semi-analytical techniques to solve simplified, linearized forms of the basic equations. These techniques sometime use specialized transformations (the pressure-squared gas formulation and the real gas pseudo-pressure, for example) that are not required in IMEX. IMEX uses sophisticated numerical techniques to solve the full reservoir flow equations without the need to resort to such methods. Because of the extreme flexibility gained through the use of numerical techniques, IMEX is the model of choice for doing forward predictions that involve the eventual appearance of multiple flowing phases (and alternate reservoir drive mechanisms), realistic multiple well operations and actual reservoir geometry. Well test software remains the method of choice for determining initial estimates for parameter values from field measurements using a simplified model.

A history match of a well test should be carried out before using IMEX to do more advanced field modelling. A well test is often the only historical information available for a new pool, and plays an important role in understanding reservoir behavior. History matches provide the necessary simulator calibration to give confidence in the prediction of pool depletion strategies, particularly with respect to new well locations. With a history match in hand, sensitivities to changes in parameters can also be estimated and confidence limits in the analysis of the original test can be established. This allows estimation of risk factors in further calculations. Note that matching must be carried out in an ad hoc manner as CMG does not currently supply the software to present IMEX pressure output in the forms most often used in well test analysis. The use of a simple well test package often facilitates the examination of IMEX output in well test situations.

With a match in hand, the engineer can go on to estimate well deliverability, production profiles, reserves, and, eventually, an optimal depletion strategy.

Well tests can be simulated before they are carried out to help design economic draw down and build up durations, along with rates. Even though these latter simulations must be carried out in IMEX using data from analogue pools or from other sources, invaluable insight is obtained regarding the set up of the proposed test.

Some guidelines will now be given for preparing data sets for well test simulations using IMEX. The results of simulations prepared using these guidelines have been calibrated with well test software and have been found to generate valid tests. They also have been used to match actual field tests. A sample data set, and a Log-Log plot of the output generated by IMEX using that data set, follow.

RESERVOIR DATA SECTION

The Radial grid option provides the simplest approach to grid design if the shape of the reservoir is generally unknown. This type of grid will account for the volume of the reservoir without requiring any detailed specifications for the reservoir boundary. A Cartesian or Variable Thickness grid can be used if seismic or other geological indications of pool geometry are available. However, the simplicity of the Radial grid option should not be overlooked, and its use should be carefully considered before more involved models are created. As well test modelling often assumes a simple homogeneous reservoir of uniform thickness, a single layer IMEX model usually suffices ($NK = 1$). Reservoir thickness can be derived from log analysis and entered into IMEX. Because of the overall simplicity of the model, NI (the number of

blocks in the radial direction) can be fairly large. This permits high accuracy, yet running times will not be excessive. Values for NI on the order of 25-50 should suffice.

Radial block sizes should begin as fractions of a metre, where the first block should be somewhat larger than twice the actual wellbore radius. Block sizes should increase so that the radial locations of the block centres are roughly in geometric progression. The sizes of the outer blocks should be adjusted so that the total reservoir size agrees with known geological estimates, or the information obtained about the reservoir's boundary location from long time behavior analysis of the well test. The IMEX adaptive implicit formulation ensures that the small, near wellbore grid blocks can be easily tolerated by the simulator. Note that the wellbore radius specified for the grid (RW) should be equal to the actual wellbore radius.

Sudden large increases in the radial block dimensions can have the effect of imposing an artificial, internal boundary in the simulation, causing boundary like slope changes in pressure profiles. If there is concern about grid block sizing, simulations can be always be re-run with smaller (halved) blocks, or with a more gradual increase in block size, and the results compared to those generated by the coarser grid. If important discrepancies appear, there should be cause for concern and grid block sizes should be reduced for subsequent runs.

Porosity and rock compressibility information should be set based on the best information available, duplicating what was used in the well test analysis. Reservoir permeability should be set based on the results of the well test analysis.

If the reservoir is naturally fractured, the double porosity reservoir modelling feature of IMEX should be used. Porosities are required for both the Fracture and Matrix systems in these situations. (IMEX regards these values as describing the fraction of bulk reservoir rock that is attributed to the appropriate system.) Matrix porosity will likely be determined from logs and will be the same value as was entered into the well test analysis. Fracture porosity will be backed out of the OMEGA parameter determined during the well test analysis and then entered into IMEX. OMEGA represents the ratio of storativities between the Fracture and Matrix systems and if equal compressibilities are assumed (an assumption likely predicated by lack of detailed information about the fractures), OMEGA is determined to a large extent by the simple ratio of porosities.

Like porosities, permeabilities also have to be entered for the both the Matrix and Fracture systems in IMEX. These values represent the permeabilities of their respective systems, as determined across a cross section of bulk reservoir rock. Fracture permeability is essentially the permeability determined from well test analysis (for instance, from Horner analysis), and it should be entered directly into IMEX. Matrix permeability should be backed out from the LAMBDA parameter determined from well test analysis. Note that a choice has to be made for matrix block geometry and fracture spacing before this calculation can be carried out.

Basically, any reasonable choice will do, as it is only the composite LAMDA that determines flow properties in both the well test model and IMEX. Nevertheless, these choices affect the derived value of matrix permeability, and if it is desirable to correlate the derived value with other indicators (such as core determined values), a reasonable set of parameter values should be used. The choices made should be the same for well test analysis and for IMEX.

If a multiple layer model is being examined, perhaps as a more geologically palatable way to explain multiple porosity behavior in a well test, it is easy to set up IMEX accordingly.

PVT AND INITIAL DATA SECTION

PVT properties should be entered in the usual way for the flowing phase. Simple, and reasonable, estimates should be made for the properties of the remaining two non-flowing phases present in IMEX. For instance, if gas flow is being modelled, constant oil formation volume factors and viscosities (= 1), and increasing, but small, solution gas-oil ratio values should be specified.

Relative permeabilities should be entered as simple straight lines, with unity endpoint values. (Thus, the oil-water table needs only two data lines.) The connate water saturation should be set to what exists in the field. Some type of residual oil saturation will need to be input, which means the liquid-oil table can have as few as three data lines.

The reservoir's initial conditions should be set with a (non zero) connate water saturation. The remaining pore space should be made up of the flowing phase of interest. If the reservoir is a gas reservoir, a tiny amount of saturated oil should be initialized. The initial reservoir pressure should be set to the same value as entered into the well test analysis.

WELL DATA

The well should be defined in the first block for a Radial grid, or suitably located in a more general Cartesian grid. The well radius should be set to be the actual radius. A skin value can be set as determined from the well test. Note that if a large negative skin is required, a more detailed model will have to be constructed incorporating an actual fracture in a Cartesian grid. Note also that skin can be modelled as a region of lowered permeability around the wellbore, if desired. This would involve the resetting of the permeability of a few of the blocks nearest the well. The extent of the region and the factor by which permeabilities should be lowered would have to be determined by matching.

The well should begin operation on the appropriate day and time at the required (high) rate. When the time comes for the well to be shut in, a tiny (but nonzero) rate should be set. (This rate could be a hundredth of a standard cubic metre per day for fluids, or a standard cubic metre per day for gas.) This maintains the well as an operating well in IMEX, and permits continued printing of its Bottom Hole Pressure (BHP) in the IMEX output. These latter values are interpretable as the required flowing BHP when the well is operating at the high rate, and as the shut in pressures when the well is on trickle. This approach allows consistent use of suitable software to obtain the BHPs throughout the simulation, regardless of the state of the well, and avoids having to activate block pressure output.

Note that IMEX does not allow for the specification of non-Darcy flow effects near the wellbore for gas wells (turbulent flow) other than by explicitly adjusting permeabilities and/or transmissibility multipliers near the wellbore, or by adjusting the skin. Rate dependent skins can be modelled by calculating the skin for the new rate and re-setting the skin at the changes.

WELL DATA, WELLBORE STORAGE

No provision has been made so far for wellbore storage in the well model. Such storage effects can be simulated in IMEX by the use of a block volume modifier (VOLMOD) to increase the volume of the well block. (Specify two VOLMODs using both *FRACTURE and *MATRIX if a Double Porosity model is being used. It is the Fracture volume that matters in these situations.) The first choice for the multiplier should be the wellbore (tubular) volume (the entire volume, from reservoir to surface) divided by the (Fracture) pore volume of the wellbore block. This value for the volume modifier can be checked by ensuring that the value for the

(Fracture) block volume that is reported for the well block in the IMEX output file corresponds to the wellbore volume. The value can than be adjusted to match the observed behavior of the well test at early times.

The procedure just described is based on the idea that wellbore storage arises from having a wellbore filled with compressible fluid, where the storage coefficient can be approximated as the wellbore volume multiplied by the fluid compressibility at the well block pressure. The volume provided by the VOLMOD-adjusted block volume accounts for the wellbore volume in this application and IMEX automatically handles the compressible nature of the fluid it contains. Note that multipliers on the order of hundreds or more may be required.

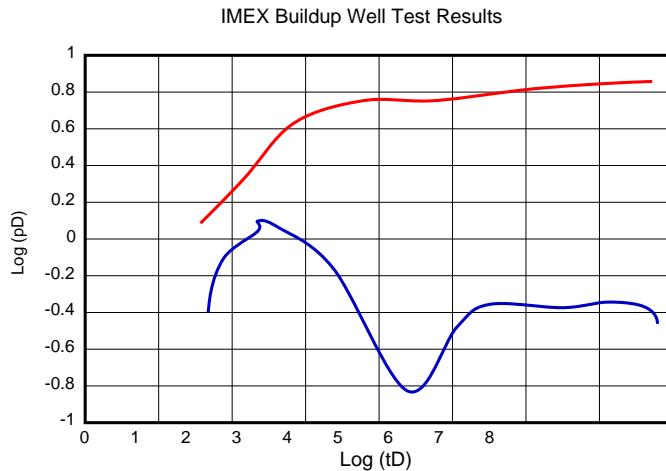
NUMERICAL PARAMETER CONTROL

As regards the numerical parameters, the simple nature of well test modelling allows specification of small timesteps without great run time penalties. The initial timestep can be of the order of a minute, or even less. A timestep of the order of a few seconds is not reasonable however, due to the lack of accurate wellbore modelling at length scale of a fraction of a metre. Timesteps will then be primarily controlled by pressure NORMs. Values of up to about half an atmosphere are reasonable here. Finally, as the simulation progresses, the maximum timestep will be limited by timestep size. A maximum timestep limit of a day (or a few days) should suffice. These type of parameters will give up to thousands of quickly converging timesteps, mimicking the quantity of data obtained from an electronic gauge.

Once a representative run is obtained with the timestep levels defined by the above conservative settings, it is easy to try a few sensitivity runs with larger limits. Such experiments could very possibly obtain accurate results in considerably less time, and thereby lead to suitable alterations in run time parameters for later trials.

SUMMARY

These guidelines should generate simulation results that will almost exactly overlay the well test results. Some tuning might be required if the well test analysis was not particularly accurate, as some of the data for IMEX requires well test derived results. Once a tuned IMEX data set is available, modifications can be made to simulated the more realistic operating conditions encountered in a developing pool, thereby starting the search for an optimal depletion strategy.



```

*****
* IMEX Example Data Set for Well Test Analysis.
* This is a dual porosity reservoir. With 40 radial blocks.
* The reservoir is a gas reservoir.
*****


*****
* I/O Section
*****


*TITLE1  'WELL TEST EXAMPLE DATA SET'
*TITLE2  'Gas Reservoir. Pressure Build-up Test.'
*INCLUDE 'caseid.dat'

*INUNIT *SI
*OUTUNIT *SI

*WPRN *WELL 0
*WPRN *GRID 0
*WPRN *ITER *BRIEF

*OUTPRN *WELL *BRIEF
*OUTPRN *GRID *PRES

*WSRF *WELL *DOUBLE
*WSRF *GRID *NEVER

*OUTSRF *WELL *ALL

*****
* Reservoir Description
*****
*GRID *RADIAL 40 1 1 *RW 0.10 ** Note that the well radius must be
                           ** same as that defined in the
                           ** well data section ( after *RUN).
*DI IVAR
    0.2   0.1   0.1   0.1   0.1   0.2   0.2   0.5   0.75
    1     1     2     2     5     5     7.5   7.5
   10    10    15    15    20    20    25    25
   30    40    50    70    90
  100   100   100   100   100   100   100   100

*DJ *CON 360
*DK *CON 100
*DEPTH 1 1 1 1000
*DUALPOR
*SHAPE *GK
*DIFRAC *CON 1
*DJFRAC *CON 1
*DKFRAC *CON 1

*POR *MATRIX *CON 0.050
*POR *FRACTURE *CON 0.005

*PRPOR *MATRIX 25000
*PRPOR *FRACTURE 25000

*VOLMOD *MATRIX *IJK 1 1 1 150 ** Model the wellbore storage by
*VOLMOD *FRACTURE *IJK 1 1 1 150 ** block volume multipliers.

```

```

*COPOR *MATRIX 0.1E-6
*COPOR *FRACTURE 0.1E-6
*PERMI *MATRIX *CON 0.00025
*PERMI *FRACTURE *CON 2.5
*PERMJ *MATRIX *CON 0.00025
*PERMJ *FRACTURE *CON 2.5
*PERMK *MATRIX *CON 0.00025
*PERMK *FRACTURE *CON 2.5

*****
* Fluid properties.
*****
*MODEL *BLACKOIL
*PVT 1
** P      RS   BO   EG   VISO  VISG
 11200  1.01  1   112  1   0.015
 13000  1.03  1   128  1   0.016
 13800  1.05  1   135  1   0.016
 14700  1.07  1   143  1   0.017
 15600  1.09  1   152  1   0.017
 16400  1.11  1   159  1   0.017
 17300  1.13  1   167  1   0.018
 18100  1.15  1   174  1   0.018
 19000  1.17  1   182  1   0.019
 19900  1.19  1   190  1   0.019
 20700  1.21  1   197  1   0.020
 21600  1.23  1   206  1   0.020
 22500  1.25  1   214  1   0.020
 23300  1.27  1   221  1   0.021
 24200  1.29  1   229  1   0.021
 25100  1.31  1   237  1   0.022

*DENSITY *OIL     800
*DENSITY *GAS     1
*DENSITY *WATER   1100

*CO 0.1E-6
*CVO 0

*BWI    1.0
*CW    0.1E-6
*REFPW 25000
*VWI    0.5
*CVW    0

*****
* Relative permeability data.
*****
*ROCKFLUID
*RPT 1 *STONE2
*SWT
** SW   KRW   KRO   PCOW
  0.1  0     1     0
  1     1     0     0

*SLT  *QUAD  *ON
** SL   KRG   KROG  PCOG
  0.1  1     0     0
  0.3  0.7   0     0
  1     0     1     0

```

```

*****
* Initialize reservoir.
*****
*INITIAL
*VERTICAL *OFF

*PRES *MATRIX *CON 25000
*PRES *FRACTURE *CON 25000
*PB *MATRIX *CON 25000
*PB *FRACTURE *CON 25000
*SO *MATRIX *CON 0.00001
*SO *FRACTURE *CON 0.00001
*SW *MATRIX *CON 0.1
*SW *FRACTURE *CON 0.1
*****
* Numerical controls.
*****
*NUMERICAL
*MAXSTEPS 9999
*NORM *PRESS 25
*DTMIN 0.00001
*****
* Well data section.
*****
*RUN
*DATE 1992 6 0.01
*DTWELL 0.001
*DTMAX 1.00
*WELL 1 'WELL'
*PRODUCER 1
*OPERATE *GAS 1000000          ** Produce at a large rate.
*OPERATE *MIN *BHP 100 *STOP
*GEOMETRY *K 0.1 0.5 1.0 0
*PERF *GEO 1
    1     1     1     1.0
*DATE 1992 6 3.00
*DTWELL 0.00005
*ALTER           ** Shut in the well. (Produce at a very-very small rate.)
    1
    1
*DATE 1992 9 0.01
*STOP

```

Appendix G

Analytical Aquifers

The role of aquifers in studying reservoir performance is usually quite large. The aquifers supply additional energy to a connected reservoir in the form of water influx. In reservoir simulation studies, aquifers can be represented as 1) Numerical Aquifers (additional grid blocks added) or 2) Analytical Aquifers.

Aquifers represented by Numerical Methods have the disadvantage of increasing the number of blocks, which increases both the required computer processing time (CPU) and storage. The Analytical Methods can often provide the same results as the Numerical Aquifer, while using less computer resources. Analytical Aquifers will fail to properly model reservoir fluids flowing back to the aquifer. In this case the use of Numerical Aquifers is preferred.

There are two well-known analytical methods, which are widely used in the simulation industry. The first method is known as the Carter-Tracy method, which is a modified version of the original Van-Everdingen and Hurst method. The second and more recent is the Fetkovich method. The first method is a rigorous mathematical solution based on the solution of the radial flow equation, while the second is a simple material balance equation. In most cases, both methods predict the water influx volumes within acceptable engineering accuracy.

Mathematical Background

Van-Everdingen and Hurst Method

This method is based upon solving the radial equation using the Laplace transform as applied for different boundary conditions. The radial flow equation can be written as:

$$\frac{\partial^2 P(t_d)}{\partial r^2} + \frac{1}{r} \frac{\partial P(t_d)}{\partial r} = \frac{\partial P(t_d)}{\partial t_d} \quad (1)$$

Where

$P(t_d)$ = Pressure at distance ‘ r ’ and dimensionless time ‘ t_d ’

‘ t_d ’ = Dimensionless time, defined as given in Equation (2)

$$t_d = \frac{k}{f\mu C_e R_o^2} \text{ (time)} \quad (2)$$

Time = Real simulation time

μ = Viscosity of fluids

f = Fraction of aquifer/reservoir connection

C_e = Effective fluid and rock compressibility

k = Absolute permeability

R_o = Effective reservoir/aquifer radius

In the reservoir simulation study the Constant Rate solution is used for finite or infinite aquifer extent.

Infinite Aquifer with Constant Rate Solution

The dimensionless pressure drop $P(t_d)$ at any dimensionless time (t_d) for the infinite aquifer and under a constant rate solution, is expressed by Equation (6) below:

$$P(t_d) = \frac{4}{\pi^2} \int_0^\infty \frac{(1 - e^{-u^2 t_d}) du}{-u^3 [J_1^2(u) + Y_1^2(u)]} \quad (3)$$

Where J and Y are Bessel functions and u is an operator. The numerical solution of the above integral provides the general solution of an infinite aquifer with constant Rate. The solution is tabulated in Table (1) for (t_d) values between $t_d = 0.01$ to $t_d = 1000$. The value of $P(t_d)$ at for (t_d) below 0.01 is computed from Equation (4), while for (t_d) greater than 1000.0, $P(t_d)$ is calculated from Equation (5).

$$P(t_d) = \frac{2}{\pi} t_d^{1/2}, \quad t_d < 0.01 \quad (4)$$

$$P(t_d) = \frac{1}{2} [\log t_d + 0.80907], \quad t_d > 1000 \quad (5)$$

Table(1) is saved in the CMG aquifer model, and the extrapolation beyond t_d of 1000, equation (5) is used.

Limited Aquifers with Constant Rate Solution

When Equation (3) above is applied to the limited aquifers with constant rate, a simpler solution is derived as given by Equation (6) below,

$$P(t_d) = \frac{2}{(R_d^2 - 1)} \left[\frac{1}{4} + t_d \right] - \frac{(3R_d^4 - 4R_d^4 \log R_d - 2R_d^2 - 1)}{4(R_d^2 - 1)^2} + 2 \sum_{\beta_1, \beta_2}^{\infty} \frac{e^{-\beta^2 t_d} J_1^2(\beta_n R_d)}{\beta_n^2 [J_1^2(\beta_n R_d) - J_1^2(\beta_n)]} \quad (6)$$

Where the B_1, B_2 are the roots of the following equation

$$[J_1(\beta_n R_d) Y_o(\beta_n R_d) - Y_1(\beta_n R_d) J_o(\beta_n R_d)] = 0 \quad (7)$$

The above equation is solved and tabulated for several values of R_d , where R_d is defined as R_e/R_o , where (R_e = external aquifer radius, R_o = external reservoir radius). Table (III) of the original work in Reference 1 gives values of $P(t_d)$ versus t_d for R_d values of 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 6.0, 7.0, 8.0, 9.0, and 10.0. The same table is attached in this report as Table (2) for reference. Figure (1) displays these results.

The last term in the above equation becomes insignificant at larger values of t_d . Thus, beyond the tabulated t_d values and for any R_d , Equation (8) is used for extrapolating t_d .

$$P(t_d) = \frac{2}{(R_d^2 - 1)} \left[\frac{1}{4} + t_d \right] - \frac{(3R_d^4 - 4R_d^4 \log R_d - 2R_d^2 - 1)}{4(R_d^2 - 1)^2} \quad (8)$$

In CMG models, equation (8) is not used directly, the last two entry points in the user input table are extrapolated for larger dimension time. The user has to be sure Equation (8) has approximate linearity for his last two values of t_d . The user should enter the entire table from the supplied tables (Table 2) to ensure this is the case.

Carter-Tracy Method

This is a modified Van-Everdingen and Hurst method, which doesn't require the superposition theorem. It changes the assumption from "a field producing at a constant hydrocarbon rate" to "an aquifer that influx water at a constant rate". This assumption simplifies the solution to that shown in Equation (9), which is implemented in the simulator.

$$W_e(t_{dj}) = W_e(t_{dj-1}) + \left[\frac{UU * \Delta P(t_{dj}) - W_e(t_{dj-1}) * P'(t_{dj})}{P(t_{dj}) - t_{dj-1} P'(t_{dj})} \right] (t_{dj} - t_{dj-1}) \quad (9)$$

$$UU = \text{Aquifer constant} = 1.119 * \phi * C_e * R_o^2 * h * \left[\frac{\theta}{360} \right] \quad (10)$$

$$\Delta P(t_{dj}) = \text{Change in boundary pressure} = P_{res}(t_j) - P_{res}(t_{j-1})$$

$$P(t_{dj}) = \text{Dimensionless pressure drop}$$

C_e = Effective compressibility = $C_w + C_f$

C_w = Water compressibility

C_f = Rock compressibility

$$P'(t_{dj}) = \frac{\partial P(t_{dj})}{\partial t_{dj}}$$

$W_e(t_{dj})$ = Cumulative water in flux at time (t_{dj})

H = Aquifer thickness

R_o = Reservoir radius

θ = Contact angle between reservoir/aquifer

Fetkovich Method

The Fetkovich method is based on a simple material balance concept, and it is considered an approximation method compared to the Van-Everdingen and Hurst method. This method doesn't need tables, or dimensionless groups. It assumes "the flow of aquifer water into a hydrocarbon reservoir is modeled in precisely the same way as the flow of oil from a reservoir into a well". Thus an inflow flow equation combined with an average reservoir pressure change constitute the essential ingredients to arrive to the following equation:

$$\Delta W_{en} = \frac{W_{ei}}{P_i} [\bar{P}_{an-1} - \bar{P}_n] * [1.0 - e^{(-J_w P_i \Delta t_n / W_{ei})}] \quad (11)$$

Where

$$[\bar{P}_{an-1}] = P_i \left[1.0 - \frac{\sum_{j=1}^{j=n-1} \Delta W_{ej}}{W_{ei}} \right] = \text{Average aquifer pressure} \quad (12)$$

$$[\bar{P}_n] = \frac{P_{n-1} + P_n}{2} = \text{Interface pressure between aquifer/reservoir} \quad (13)$$

$$J_w = \frac{7.08 * 10^{-3} kh}{\mu \left[\ln \left(\frac{R_e}{R_o} \right) - \frac{3}{4} \right]} = \text{Aquifer productivity index} \quad (14)$$

$$W_{ei} = C_e * f * \pi * (R_e^2 - R_o^2) * h \phi * P_i = \text{Initial amount of encroachable water} \quad (15)$$

ΔW_{en} is the incremental water influx in the interval Δt_n .

Practical Aspects of Using Analytical Aquifers

There are several points of concern regarding how to represent an aquifer in a simulation study. Those are the Aquifer ‘TYPE’, ‘SOLUTION METHOD’, ‘AQUIFER PARAMETERS’ and the ‘LEAK SWITCH’. The following shows some user guidelines for each point.

a) Selecting Type of Aquifer

In IMEX, a user is able to select the type of aquifers to be connected to a specific reservoir. Those types are BOUNDARY or REGION or BOTTOM

The ‘TYPE’ should be selected according to the actual aquifer location, and on how the aquifer and reservoir are in contact. This could be determined from the geological description of the problem. If the geology did not provide enough information about the aquifer type, matching the reservoir performance using different types of aquifer is possible. An aquifer type and/or aquifer parameters could serve as a matching parameter when a field history indicates more energy is needed to maintain the reservoir pressure, or additional water has to be produced. More than one aquifer can be connected to a simulation model. In addition different “types” can be specified for each aquifer.

b) Selecting Method of Solution

An aquifer can be represented numerically or by either of the analytical methods described above. The three methods should provide similar results when using the same input parameters, provided the reservoir does not efflux fluids to the aquifer. Selection of one method over the others can be aided by using the following criteria:

1. Using Numerical Aquifers: This method is the most accurate method, since it provides an exact geometrical and probable geological description of the problem. This is helpful if the aquifer is allowed to contain either slight volumes of oil or some dissolved gases. The drawback of this method stems from the additional use of computer storage and CPU time. Thus, this is best used when the computer memory storage and computing time are not of concern. For large aquifers extent ($R_d > 10$), huge number of large blocks sizes would be required to correctly present the aquifer. This yields to significant decrease in the simulation speed.
2. Using Carter-Tracy Aquifers: This method is based on a rigorous numerical solution and it provides an accurate solution, however, it requires a different table entry to simulate a given aquifer extent (value of R_d). Thus changing the aquifer extent, require a change of the entry table. There are limited R_d values (1.5, 2, 2.5, 3.0, 3.5, 4.0, 4.5, 5, 5.5, 6, 7, 8, 9, 10, and infinite). Other values need the solution of equation (3) above.
3. Using Fetkovich Aquifer: This method is the simplest method of analytical aquifer presentation, and still provides reasonable answers. It requires the definition of the aquifer parameters, while it can handle any aquifer extent, except the infinite. For infinite aquifer extent, a value of $R_d > 100$ should be adequate.

Aquifer Parameters

Same aquifer parameters are used for both of Carter-Tracy and Fetkovich solutions.

For the current version (IMEX9800+) the following parameters are needed:

- Aquifer Thickness (H_AQ).
- Aquifer Porosity (POR_AQ).
- Aquifer Permeability (PERM_AQ).
- Effective Reservoir External Radius (R_o).
- Reservoir/Aquifer contact angle (radians) (F_AQ).
- Ratio between the external Aquifer radius to the Reservoir external radius (R_d).

The model is able to calculate default values for all of the above parameters for either Carter-Tracy or Fetkovich aquifers. However, it is highly recommended that a user enter all of the above parameters, since these parameters would play an important role in the reservoir performance. The model uses the reservoir data to calculate defaults of parameters not explicitly defined, the model's choice of defaults is consistent, but may not be intuitive. A list of model defaults is provided below. Accordingly, care must be exercised when the user allows the model defaults to simulate an aquifer.

If the CARTER-TRACY METHOD was selected, the user has to consider the following:

- a) For an infinite aquifer: The Model has this function as the default. The internal table terminates at $t_d=1000$. Beyond this value, a proper extrapolation is done automatically. Thus, the user need not enter a table.
- b) For a finite aquifer: The user has to enter the correct table for a pre-determined value of R_d (Table 2).
- c) The finite aquifer table the user inputs from Table (2), has to have the same infinite table value up to its starting time, while for times larger than the table last entry, the model linearly extrapolates using the last two entry points. Thus, the user has to enter the table correctly and in full.

Model Defaults

Assume the following:

H	= Effective Reservoir Thickness- Computed from Reservoir Geometry
H_AQ	= Aquifer Thickness- Computed Below
POR_AQ	= Aquifer Porosity
PERM_AQ	= Aquifer Permeability
AREA_Contact	= Connected Area Between Reservoir and Aquifer (obtained from model)
F_AQ	= Contact Angle between Reservoir and Aquifer
R_o	= Effective Reservoir radius
R_e	= Effective Aquifer External Diameter
R_d	= R_e/R_o

Parameter	Boundary	Region	Bottom
POR_AQ	Volumetrically Average Porosity for connected blocks	Volumetrically Average Porosity for connected blocks	Volumetrically Average Porosity for connected blocks
PERM_AQ	Volumetrically Average Permeability for connected blocks	Volumetrically Average Permeability for connected blocks	Volumetrically Average Permeability for connected blocks
H_AQ	H	H	$\sqrt{(AREA_Contact)}$
R_o	$AREA_Contact/(H_AQ*2\pi)$	$\sqrt{(AREA_Contact/\pi)}$	$\sqrt{(H_AQ/H)2\pi}$
F_AQ	1.0	1.0	$2*\text{Atan}(H_AQ/H)/2\pi$

Example:

Nx=Ny=Number of grids in x,y directions=11

Nz= Number of grids in Z direction = 12

Porosity = Constant = 0.3

Kx=Ky=Kz = x,y,z absolute permeability = 100 md

Dx=Dy= Grid thickness for x, y directions = 100 ft

Dz = Grid thickness in Z direction = 20 ft

Model calculate defaults:

Parameter	Boundary	Region	Bottom
POR_AQ	0.3	0.3	0.3
PERM_AQ	100	100	100
AREA	1056000 ft ²	AREA-R (depends on defined region)	1210000 ft ²
H_AQ	$Nz*Dz=240'$	$Nz*Dz=240'$	$\sqrt{(1210000)}=1100'$
R_o	$1056000/(240*2\pi) = 700.28'$	$\sqrt{(AREA/\pi)}$	$\sqrt{(1100*240/\pi)}=289.88'$
F_AQ	1.0	1.0	$2*\text{Atan}(1100/240)/2\pi=0.4316$

In some applications, when using the Fetkovich method, one could use different input parameters such as aquifer injectivity index, volume of water initially available for injection, and the initial reservoir pressure, in this case a user has to manually converts this data to the required parameters above. This could be furnished as follows:

Assume

W_i	= Volume of water available for Injection. (STB)
J_w	= Aquifer Injectivity Index (STB/D/PSI)
Type	= Boundary Aquifer
Contact Area	= Known from simulation

1. Determine aquifer thickness H_{AQ} (ft), and from the contact area determine the effective reservoir radius (ft) as follows:

$$R_o = \frac{AREA - Contact}{H_{AQ} * 2\pi}$$

2. Use the W_i (STB) in the equation below to determine the external reservoir radius (R_e - ft) – use contact angle θ as 1.0, and use any reasonable value for the aquifer porosity.

$$W_i = \theta * \frac{\pi [R_e^2 - R_o^2]}{5.615} * H_{AQ} * POR_{AQ}$$

W_i (ft³) is reported in the simulation output, which can be used either for checking or for computing R_e .

3. Use the Aquifer Injectivity Index (STB/D/psi) to determine the effective aquifer permeability (md), as follows:

$$J_w = \frac{7.08 * 10^{-3} * \theta * PERM_{AQ} * H_{AQ}}{\mu_w * \ln[R_e / R_o]}$$

The μ_w is the water or (aquifer fluid viscosity) in cp.

For further description of the above subject, the reader could consult references (1) to (6) below:

1. Van Everdingen, A.F., Hurst, W., "The Application of the Laplace Transformation to Flow Problems in Reservoirs", Petroleum Transactions, AIME, December 1949.
2. Carter, R.D., Tracy, G.W., "An Improved Method for Calculating Water Influx", Petroleum Transactions, AIME, vol. 219, 1960.
3. Van Everdingen, A.F., Timmerman, E.H., McMahon, J.J., "Application of the Material Balance Equation to Partial Water Drive Reservoir", Petroleum Transactions, AIME, vol. 198, 1953.
4. Fetkovich, M.J., "A Simplified Approach to Water Influx Calculations – Finite Aquifer Systems", J. Pet. Tech., July 1971, pp. 814-828.
5. Dake, L.P., "Fundamentals of Reservoir Engineering", Elsevier Scientific Publishing Company, 1978.
6. Craft, B.C., Hawkins, M.F., "Applied Petroleum Reservoir Engineering", Prentice-Hall Inc., 1959.

Table 1 - Radial Flow, Constant Terminal Pressure and Constant Terminal Rate Cases for Infinite Reservoirs

t	P(t)
1.0 (10) ⁻²	0.112
5.0 "	0.229
1.0 (10) ⁻¹	0.315
1.5 "	0.376
2.0 "	0.424
2.5 "	0.469
3.0 "	0.503
4.0 "	0.564
5.0 "	0.616
6.0 "	0.659
7.0 "	0.702
8.0 "	0.735
9.0 "	0.772
1.0 "	0.802
1.5 "	0.927
2.0 "	1.020
2.5 "	1.101
3.0 "	1.169
4.0 "	1.275
5.0 "	1.362
6.0 "	1.436
7.0 "	1.500
8.0 "	1.556
9.0 "	1.604
1.0 (10) ¹	1.651
1.5 "	1.829
2.0 "	1.960
2.5 "	2.067
3.0 "	2.147
4.0 "	2.282
5.0 "	2.388
6.0 "	2.476
7.0 "	2.550
8.0 "	2.615
9.0 "	2.672
1.0 (10) ²	2.723
1.5 "	2.921
2.0 "	3.064
2.5 "	3.173
3.0 "	3.263
4.0 "	3.406
5.0 "	3.516
6.0 "	3.608
7.0 "	3.684
8.0 "	3.750
9.0 "	3.809
1.0 (10) ³	3.860

Table 2 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs

R1 = 1.5	R = 2.0	R = 2.5	R = 3.0				
$\beta_1 = 6.3225$	$\beta_1 = 3.1965$	$\beta_1 = 2.1564$	$\beta_1 = 1.6358$				
$\beta_2 = 11.924$	$\beta_2 = 6.3118$	$\beta_2 = 4.2230$	$\beta_2 = 3.1787$				
t	P(t)	t	P(t)	t	P(t)	t	P(t)
6.0 (10) ⁻²	0.251	2.2 (10) ⁻¹	0.443	4.0 (10) ⁻¹	0.565	5.2(10) ⁻¹	0.627
8.0 "	0.288	2.4 "	0.459	4.2 "	0.576	5.4 "	0.636
1.0 (10) ⁻¹	0.322	2.6 "	0.476	4.4 "	0.587	5.6 "	0.645
1.2 "	0.355	2.8 "	0.492	4.6 "	0.598	6.0 "	0.662
1.4 "	0.387	3.0 "	0.507	4.8 "	0.608	6.5 "	0.683
1.6 "	0.420	3.2 "	0.522	5.0 "	0.618	7.0 "	0.703
1.8 "	0.452	3.4 "	0.536	5.2 "	0.628	7.5 "	0.721
2.0 "	0.484	3.6 "	0.551	5.4 "	0.638	8.0 "	0.740
2.2 "	0.516	3.8 "	0.565	5.6 "	0.647	8.5 "	0.758
2.4 "	0.548	4.0 "	0.579	5.8 "	0.657	9.0 "	0.776
2.6 "	0.580	4.2 "	0.593	6.0 "	0.666	9.5 "	0.791
2.8 "	0.612	4.4 "	0.607	6.5 "	0.688	1.0	0.806
3.0 "	0.644	4.6 "	0.621	7.0 "	0.710	1.2	0.865
3.5 "	0.724	4.8 "	0.634	7.5 "	0.731	1.4	0.920
4.0 "	0.804	5.0 "	0.648	8.0 "	0.752	1.6	0.973
4.5 "	0.884	6.0 "	0.715	8.5 "	0.772	2.0	1.076
5.0 "	0.964	7.0 "	0.782	9.0 "	0.792	3.0	1.228
5.5 "	1.044	8.0 "	0.849	9.5 "	0.812	4.0	1.578
6.0 "	1.124	9.0 "	0.915	1.0	0.832	5.0	1.828
	1.0	0.982	2.0	1.215			
	2.0	1.649	3.0	1.596			
	3.0	2.316	4.0	1.977			
	5.0	3.649	5.0	2.358			

Table 2 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs (Continued)

R = 3.5		R = 4.0		R = 4.5	
$\beta_1 = 1.3218$		$\beta_1 = 1.1120$		$\beta_1 = 0.9609$	
t	P(t)	t	P(t)	t	P(t)
1.0	0.802	1.5	0.927	2.0	1.023
1.1	0.830	1.6	0.948	2.1	1.040
1.2	0.857	1.7	0.968	2.2	1.056
1.3	0.882	1.8	0.988	2.3	1.072
1.4	0.906	1.9	1.007	2.4	1.087
1.5	0.929	2.0	1.025	2.5	1.102
1.6	0.951	2.2	1.059	2.6	1.116
1.7	0.973	2.4	1.092	2.7	1.130
1.8	0.994	2.6	1.123	2.8	1.144
1.9	1.014	2.8	1.154	2.9	1.158
2.0	1.034	3.0	1.184	3.0	1.171
2.25	1.083	3.5	1.255	3.2	1.197
2.50	1.130	4.0	1.324	3.4	1.222
2.75	1.176	4.5	1.392	3.6	1.246
3.0	1.221	5.0	1.460	3.8	1.269
4.0	1.401	5.5	1.527	4.0	1.292
5.0	1.579	6.0	1.594	4.5	1.349
6.0	1.757	6.5	1.660	5.0	1.403
		7.0	1.727	5.5	1.457
		8.0	1.861	6.0	1.510
		9.0	1.994	7.0	1.615
		10.0	2.127	8.0	1.719
				9.0	1.823
				10.0	1.927
				11.0	2.031
				12.0	2.135
				13.0	2.239
				14.0	2.343
				15.0	2.447

Table 2 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs (Continued)

R = 5.0		R = 6.0		R = 7.0	
t	P(t)	t	P(t)	t	P(t)
3.0	1.167	4.0	1.275	6.0	1.436
3.1	1.180	4.5	1.322	6.5	1.470
3.2	1.192	5.0	1.364	7.0	1.501
3.3	1.204	5.5	1.404	7.5	1.531
3.4	1.215	6.0	1.441	8.0	1.559
3.5	1.227	6.5	1.477	8.5	1.586
3.6	1.238	7.0	1.511	9.0	1.613
3.7	1.249	7.5	1.544	9.5	1.638
3.8	1.259	8.0	1.576	10.0	1.663
3.9	1.270	8.5	1.607	11.0	1.711
4.0	1.281	9.0	1.638	12.0	1.757
4.2	1.301	9.5	1.668	13.0	1.801
4.4	1.321	10.0	1.698	14.0	1.845
4.6	1.340	11.0	1.767	15.0	1.888
4.8	1.360	12.0	1.815	16.0	1.931
5.0	1.378	13.0	1.873	17.0	1.974
5.5	1.424	14.0	1.931	18.0	2.016
6.0	1.469	15.0	1.988	19.0	2.058
6.5	1.513	16.0	2.045	20.0	2.100
7.0	1.556	17.0	2.103	22.0	2.181
7.5	1.598	18.0	2.160	24.0	2.267
8.0	1.641	19.0	2.217	26.0	2.351
9.0	1.725	20.0	2.274	28.0	2.434
10.0	1.808	25.0	2.560	30.0	2.517
11.0	1.892	30.0	2.846		
12.0	1.975				
13.0	2.059				
14.0	2.142				
15.0	2.225				

Table 2 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs (Continued)

R = 8.0		R = 9.0		R = 10	
t	P(t)	t	P(t)	t	P(t)
8.0	1.556	10.0	1.651	12.0	1.732
8.5	1.582	10.5	1.673	12.5	1.750
9.0	1.607	11.0	1.693	13.0	1.768
9.5	1.631	11.5	1.713	13.5	1.784
10.0	1.653	12.0	1.732	14.0	1.801
10.5	1.675	12.5	1.750	14.5	1.817
11.0	1.697	13.0	1.768	15.0	1.832
11.5	1.717	13.5	1.786	15.5	1.847
12.0	1.737	14.0	1.803	16.0	1.862
12.5	1.757	14.5	1.819	17.0	1.890
13.0	1.776	15.0	1.835	18.0	1.917
13.5	1.795	15.5	1.851	19.0	1.943
14.0	1.813	16.0	1.867	20.0	1.968
14.5	1.831	17.0	1.897	22.0	2.017
15.0	1.849	18.0	1.926	24.0	2.063
17.0	1.919	19.0	1.955	26.0	2.108
19.0	1.986	20.0	1.983	28.0	2.151
21.0	2.051	22.0	2.037	30.0	2.194
23.0	2.116	24.0	2.090	32.0	2.236
25.0	2.180	26.0	2.142	34.0	2.278
30.0	2.340	28.0	2.193	36.0	2.319
35.0	2.499	30.0	2.244	38.0	2.360
40.0	2.658	34.0	2.345	40.0	2.401
45.0	2.817	38.0	2.446	50.0	2.604
		40.0	2.496	60.0	2.806
		45.0	2.621	70.0	3.008
		50.0	2.746		

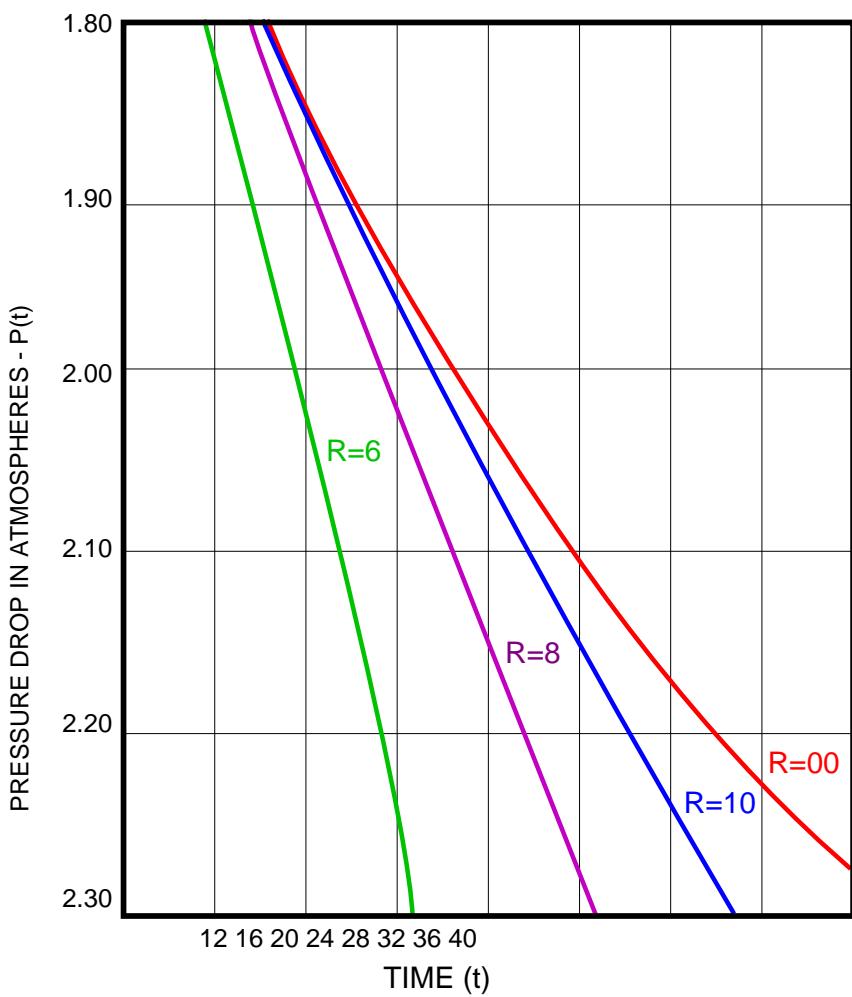


Figure G-1
Radial Flow, Constant Terminal Rate Case, Pressure Drop Versus Time, $P(t)$ Versus 1

Appendix H

Running IMEX Coupled to the GAP Surface Network Model

IMEX (2005.10 and later) can be linked to Petroleum Expert's GAP surface network model through the RESOLVE framework. RESOLVE is a tool written by P.E. 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 IMEX 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 IMEX, for example, wells can be sources or sinks depending on whether they are production or injection wells.

Before running the coupled model, the individual models have to be modeled satisfactorily in their separate applications. Thus an IMEX data set which is the case to be loaded into RESOLVE must be able to run without RESOLVE.

Once the case is loaded into RESOLVE, equivalent sources and sinks can be connected together graphically. For example, wells in an IMEX reservoir can be connected to their equivalent wells in GAP (P.E.'s surface network analysis package).

When a simulation is run, RESOLVE collects and passes data between all the coupled applications. The data passed in all cases is an IPR table (Inflow Performance Relationship - a table of node pressures against phase rates).

When a simulation is run, RESOLVE will perform the following steps:

1. Reload and initialize all cases
2. Broadcast to all clients (IMEX) to calculate inflows at all sources
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 clients (IMEX) that generated the inflow relation
6. Broadcast to all clients (IMEX) to perform a timestep of the requested length
7. Return to step 2

At each of the above steps, the command (to perform timesteps etc) is broadcast to all the client applications simultaneously; this allows the surface network to couple to multiple reservoirs models.

IMEX Timesteps and Dates / RESOLVE Timesteps and Dates

During a coupled run RESOLVE allows GAP to run using a timestep, whose size is determined by RESOLVE, to solve the Surface network (RESOLVE timestep).

After completing the timestep, RESOLVE passes well constraints for each coupled well back to IMEX along with the date (RESOLVE date) at the end of the RESOLVE timestep.

IMEX is now expected to solve the reservoir problem until the same RESOLVE date.

GAP waits until IMEX completes this solution. IMEX can take as many internal (IMEX timesteps) as required to reach the RESOLVE date.

When IMEX reaches the RESOLVE date, it passes IPR tables and other information back to GAP through RESOLVE.

Considerations and Guidelines

Connections between the reservoir and the surface network are made at well bottom-hole conditions. Thus GAP must handle flow in the wellbore.

Some or all IMEX wells can be linked to the GAP model through RESOLVE. IMEX generates IPR tables for those wells that are linked to the GAP model and in turn, gets instructions from RESOLVE on how to control those wells.

In general IMEX wells linked to GAP have their operating constraints and schedules controlled by GAP. IMEX wells not linked to GAP have their operating constraints and schedules controlled by IMEX.

IMEX is connected to GAP through RESOLVE on a per well basis, using IPR curves. VFP tables must be defined in GAP using a program like Prosper. In order to use the linked model, the GAP method for determining optimum rates must be set to VFP/IPR curve intersection.

Normally GAP (through RESOLVE) controls the timestep the coupled model takes, however IMEX can be made to provide input to RESOLVE about timestep selection. RESOLVE will then select the minimum of the GAP and IMEX timesteps.

If the RESOLVE timestep to the next RESOLVE date is too large, it is possible that an IMEX well's BHP may vary considerably while running to this date. In that case, a failsafe was implemented to limit a wells minimum pressure to 14.7 psi. The failsafe can be avoided by reducing the RESOLVE maximum timestep size.

The IMEX start date defined in the recurrent data set should be equal to or later than the GAP/RESOLVE start date. It cannot be earlier than the GAP/RESOLVE start date. On other hand, if the IMEX start date is later than the GAP/RESOLVE start date, wells in the linked model will be masked out and IMEX will not start until the start date is reached.

A linked model can restart from a non-linked run, e.g. a GAP linked model can run a prediction starting from the end of an IMEX only reservoir history match simulation.

Preparation of the IMEX Data Set

There are no GAP link specific keywords required for IMEX to couple with GAP. Although the following must be done:

1. Reservoir temperature (*TRES) must be defined in the IMEX Component Properties section.

2. The PVT of the reservoir and surface network should be the same. In other words, at reservoir temperature the surface network model PVT should match the PVT in IMEX.

But as listed below, some restrictions and differences in how well/recurrent data is interpreted are imposed. The following well data restrictions only apply to IMEX-GAP linked wells.

Recurrent Data

In recurrent data, any operation that changes the value of a well rate constraint is ignored. Rates are exclusively determined by GAP.

ON-TIME Fraction

At each RESOLVE date, the GAP down-time percentage is passed to IMEX as an on-time fraction. IMEX uses the received on-time fraction as if the same value was input through the IMEX 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 IMEX recurrent data as usual.

IMEX reports on-time fraction for all wells to RESOLVE, so that it is accessible via a RESOLVE script and readable in the RESOLVE reporting window.

Opening/Closing Wells

IMEX passes open / shut-in status for linked wells to RESOLVE at each RESOLVE date. After receiving this well status information from IMEX, GAP honors those wells IMEX shuts in and may add other shut-in wells based on its schedule or optimization procedure.

RESOLVE then runs a new GAP timestep and passes the updated well status back to IMEX. IMEX keeps the received well status for the IMEX timesteps required to reach the new RESOLVE date.

IMEX can attempt to open a well if the GAP schedule or the GAP network optimization overrides this, the well remains shut-in until the next RESOLVE date, where it will be checked again by GAP, and if the network schedule or optimization allows the well to open, it will open. This checking will occur at every RESOLVE date.

Thus either IMEX or GAP can shut-in a well, but both IMEX and GAP must agree that a well should be open before it actually opens.

From release 2010.10, IMEX is able to receive a ‘shut in with cross-flow’ command from RESOLVE, which closes the well above the formation. If a well in an IMEX data set is defined with a FULLY-MIXED cross-flow model (the default), the well will cross-flow below the formation when it receives the “shut in with cross-flow” command from RESOLVE. This improves well pressure continuity. Please check the RESOLVE manual for availability.

Monitors and Triggers

IMEX well monitors, group monitors and triggers are allowed to operate on wells or well layers only. Details are described below. The allowed operations are restricted to those which open or shut-in wells or well layers, or that modifies productivity / injectivity indices of the wells or well layers.

IMEX and GAP ONLY exchange information at each RESOLVE date, it is critical that the linked well and well layer status is not changed between RESOLVE dates.

From release 2011.10, IMEX passes the time to next recurrent date to Resolve. If this is handled by your RESOLVE Version, all IMEX recurrent keywords will be synchronized automatically by RESOLVE, as the next RESOLVE timestep will account for the next IMEX recurrent date. Please check the RESOLVE manual for availability.

For IMEX Monitor and Trigger options, which can trigger events without explicit keywords, the following considerations apply.

a) Monitor

For linked wells, monitors (*MONITOR) are checked at RESOLVE dates ONLY. For unlinked wells, monitors are checked at each IMEX timestep.

For well groups defined in IMEX, if a group includes linked wells, the applicable group monitor option is limited to the group monitor (*GCONM) and the group monitor will only be checked at RESOLVE dates. On the other hand, if a group does not include linked wells, group monitors are applied as usual. It should be underlined, however, that for coupled runs, group controls are not suitable to be applied in IMEX. They should be applied in the surface network model.

b) Trigger

Triggers (*TRIGGER) are checked at RESOLVE dates ONLY. Because of this behaviour, keywords within a trigger block that specifies the date may not be honored. These keywords should be moved out of the trigger so as to allow the dates to be synchronized by Resolve.

Well Constraint Definitions/Use/Restrictions

For coupled producers, a non-zero surface rate constraint should be given just to define the target stream phase for IMEX to use with the coupled well. For example,

*OPERATE *MAX *STO 500

indicates the target stream of the producer is oil, the rate itself is ignored.

The target stream phase is used to determine the IMEX constraint from the three phase rates received from GAP. Within the recurrent data, the keyword *TARGET with non-zero rate can be used to shift the target stream phase from one phase to another.

Similarly, non-zero STG and STW can be used to indicate a gas and a water producer (if applicable) respectively. If more than one of the STO, STG and STW constraints are defined, the first will be used.

For injectors, the injection stream to be used is given by keyword *INCOMP.

Bottom-hole pressure constraints, if applicable, should be defined in GAP. A BHP constraint given in IMEX is ignored for rate calculation purposes. However, for injectors, if the keyword:

OPERATE *MAX *BHP 1000

is given, the value of *BHP (1000) will be used to define as the upper limit for pressure in that wells IPR table which IMEX passes to GAP through RESOLVE. It will not be used as an operating constraint.

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 IMEX timestep and cause a rate mismatch between IMEX and GAP. Such conditional operations may be performed instead by the monitor keyword or within a trigger.

By rate mismatch we refer to the situation where the network model (GAP) and reservoir model (IMEX) are assuming different rates or well indices for the same well. This may include having a well open in one model and shut-in in another.

Injectors must be of the default mobility weighted type.

DATE and TIME

DATE and TIME keywords can normally be used within the IMEX data set. These can be used to obtain IMEX output when the WPRN/WSRF DATE output options are used.

The start date of IMEX cannot be earlier than the start date of the RESOLVE date schedule. The IMEX start date can be an IMEX Restart start date.

The stop date in the IMEX data set should be later than or equal to the stop date specified in the RESOLVE schedule.

Scripting

Using RESOLVE Scripts

A RESOLVE script is able to access a portion IMEX's simulation results, and can be written to do calculations using information from both IMEX and GAP. This allows RESOLVE to access IMEX variables which can be used by RESOLVE, as part of a scripted strategy, to control the network in GAP.

All of the wells, groups and sectors defined in IMEX pass results back to RESOLVE, where the well and group rates are average values between two synchronized dates. For accessible IMEX results and their scripting format, please see the IMEX Results Accessible from RESOLVE Script' in this appendix.

Accessible IMEX results are also reported in the RESOLVE Reporting window.

A script is normally designed to execute pre or post the 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 which generates scripts to do voidage replacement in a reservoir model.

IMEX 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 IMEX (as well groups) or in GAP.

Please refer the RESOLVE manual for instructions in how to use the wizard.

Editing Scripts

Generated scripts 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.

Script Naming Conventions

Care should be taken when naming instances in scripts due to RESOLVE's naming conventions. Scripting is designed to execute pre or post the solution of a specified named instance. If two IMEX instances are given the names 'IMEX' and 'IMEX-1', a script for instance 'IMEX-1' will also be executed for 'IMEX' as the name 'IMEX' is contained within 'IMEX-1'.

This can be avoided by giving the names 'IMEX-1' and 'IMEX-2' to the two IMEX instances.

Resolve Reporting

The Resolve reporting window contains IMEX result data at each synchronized date. Different from GAP reporting, the IMEX 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 "IMEX", there are well aggregate values. The same reporting entries can also be found under IMEX group names. The top group name, "FIELD" for example, includes all IMEX wells. The difference between well values reported under "FIELD" and "IMEX" is that the instance name "IMEX" includes the aggregate for only linked wells while "FIELD" includes all IMEX wells.

Other Considerations

Multiple Reservoirs

Other instances of IMEX can be set up in a RESOLVE model, allowing a surface network model to connect to multiple IMEX reservoir models. These IMEX models may have different start dates. One of the models should have a start time equal to the GAP/RESOLVE start date.

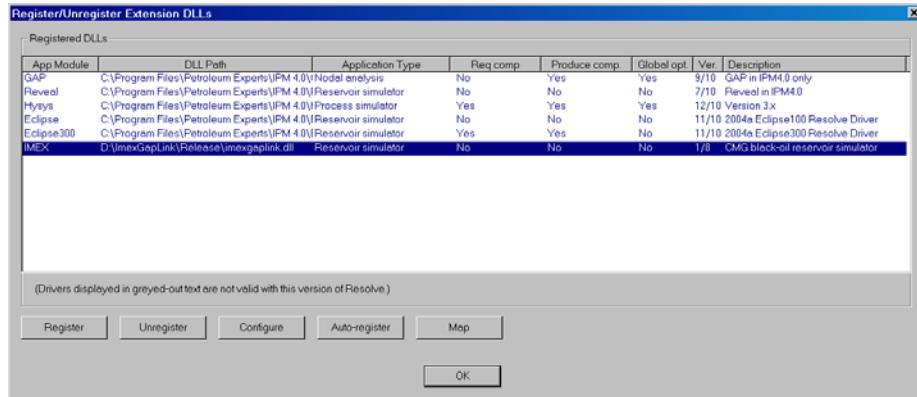
When multiple instances of IMEX are set up on the RESOLVE interface, each instance should be given a different name. Each instance may point to the same executable.

IMEX and RESOLVE communicate through ASCII files. Four ASCII files per IMEX instance are created and updated during the linked run. The files have prefix of the IMEX data set and suffixes related to their use within the linked model (LSResolve, LDResolve, LSImex, and LDImex). These files must not be altered during a linked model run.

Getting Started

- a) Register the imexgaplink.dll as a driver for RESOLVE

Start RESOLVE and then choose the 'Register Driver' from 'Drivers'. In the window similar to window below, click the 'Register' button to assign the location of imexgaplink.dll. Step a) only needs to be done initially or when you need to update drivers as RESOLVE remembers the location of registered Drivers.



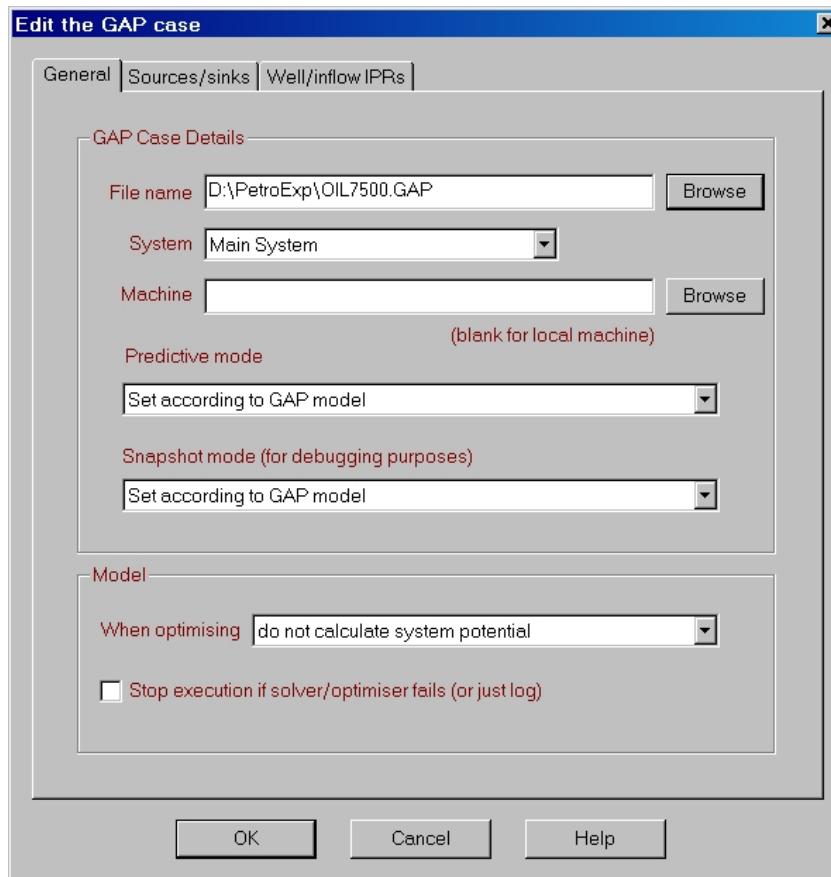
Click 'OK' to finish.

- Open a new RESOLVE Graphic View

Click the 'File' -> 'New' to start a new graphic view in RESOLVE.

- 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 IMEX model into RESOLVE

Similar to the steps used to add a GAP model, from ‘Edit System’ -> ‘Add Client Program’, choose ‘IMEX’ and add it onto the RESOLVE view. Edit the CMG Case setting by double clicking the IMEX icon.

e) IMEX Case Settings (see graphic below)

- Select a computer

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.

- Assigning the simulator and data set

In all cases, the user needs to give the location of both the IMEX executable and the IMEX 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.

- Entries for remote job submission

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 also editable. For example, it is necessary to edit this command if the user has a different login name on the server computer than that on the local computer. In this case it is necessary to alter *ssh -l username_on_client* to *ssh -l username_on_server*.

“Local path to work folder” is the local path that points to the folder/directory where the IMEX data set exists. Please see “Location of the work folder” in “Remote Simulation Job Submission via SSH” below for a more detailed description of the entry.

- Other options

The “Timestep selection for network solution” radio buttons provides the option for IMEX 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 IMEX for the next step.

Entry of “Additional command switches for the executable” can be used to input additional simulation options.

- Restart case

It is possible to give the restart file (.irf file) via command option. Such as,

```
-r case1.irf
```

However there is a little complication in the case of remote job submission. The ssh command needs the file name, case1.irf, be quoted and have back slash (\) be escaped. Hence it is recommended to give the .irf file within the data set using INDEX-IN keyword.

If the “-r” option has to be used anyway, the next two examples can be used as a guide.

For Unix server with restart file /cmg/datasets/case1.irf, the option should be:

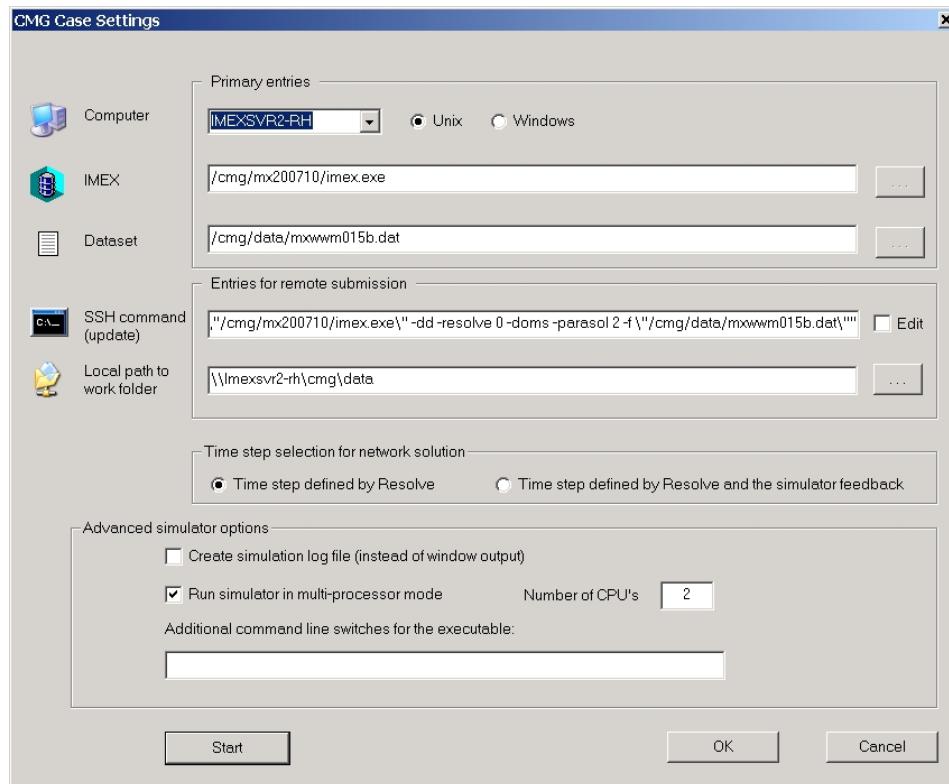
```
-r \"\cmg\datasets\case1.irf\"
```

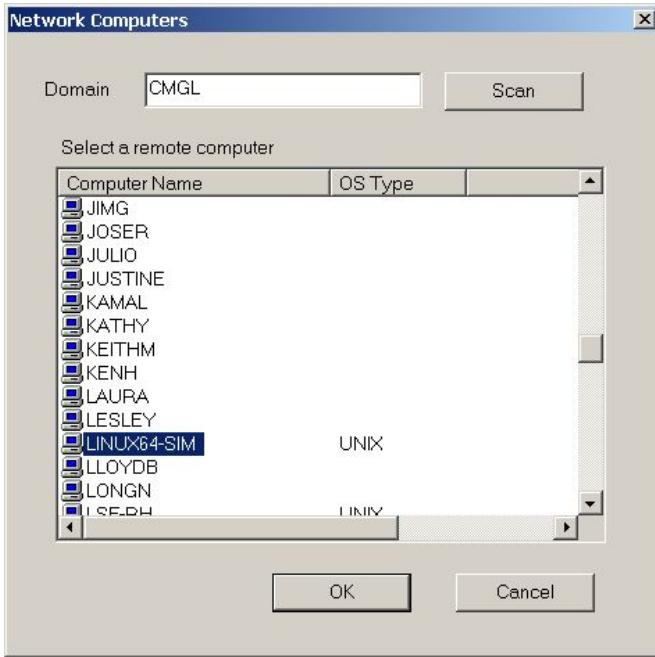
For Windows server with restart file \\cmg\\datasets\\case1.irf, the option should be:

```
-r \"\\\\cmg\\\\datasets\\\\case1.irf\"
```

- Use of Start and OK button

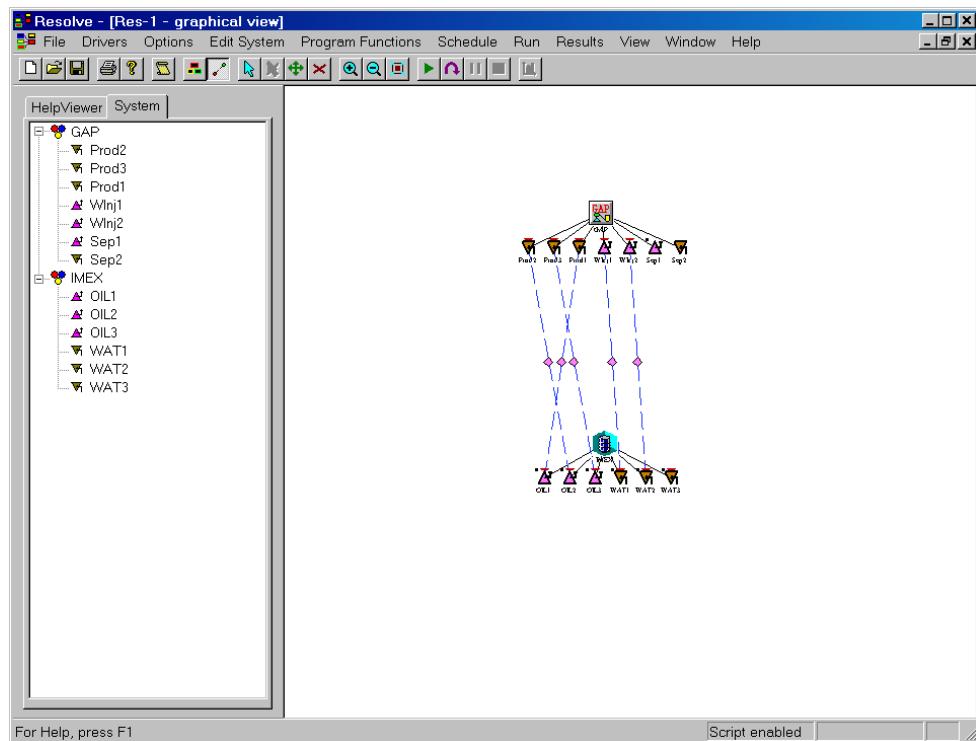
From release IMEX-2007-11, a new “Start” button is added to perform the functions of the original “OK” button. “Start” starts the model initialization. After the simulator finishes its initialization, the “Start” button converts to “Stop”, so that user has a chance to stop the simulation at its initialized stage. The remaining “OK” button is used to only save the current case settings, which does not execute the simulator. However, if the file name of the data set is changed, the “OK” button will become inactive. This ensures that, in order to accept the replaced data set, the user needs to start a simulation to update the well information.





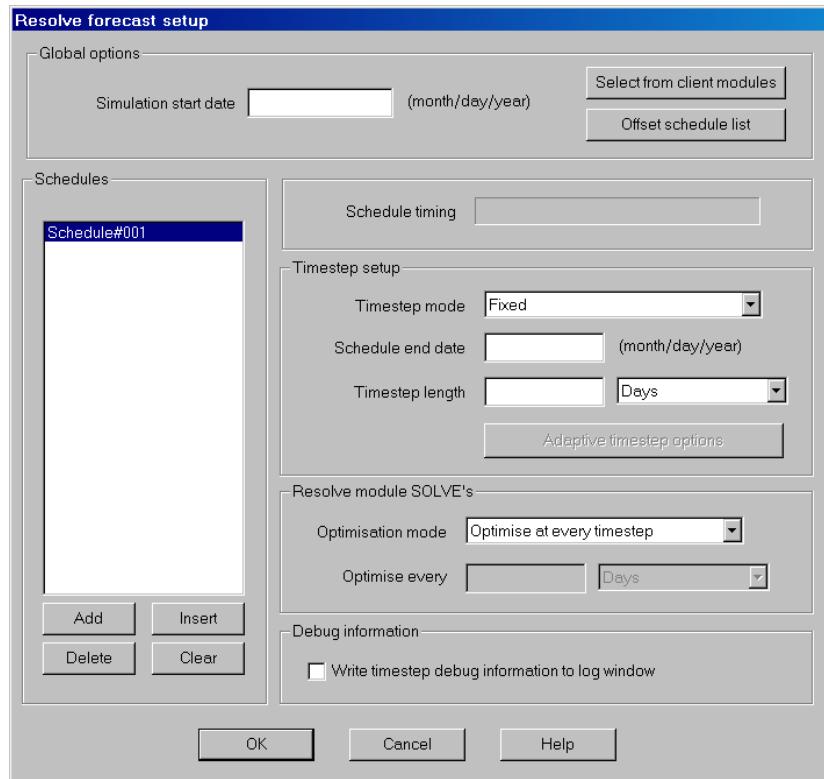
f) Link the wells between IMEX and GAP

Use the link function to connect the corresponding wells between IMEX 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 timestep is chosen and the option allowing IMEX timestep selection feedback is on, IMEX will still be able to reduce the timestep 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 tools bar. The user can also save the current settings to a RESOLVE file with the name suffix of .rsl. Loading the .rsl file provides a shortcut to starting a case without the duplication of the steps from b to f.

From release IMEX-2007-11, the saved .rsl file includes IMEX 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 only loading and then modifying an existing case without initializing the existing reservoir model.

About the IMEX log

There may be a date mismatch between the log of IMEX and RESOLVE /GAP of less than one day at most. This is due to differences in how IMEX and RESOLVE/GAP calculates date from time. The mismatch only affects the date displayed and does not affect the actual time used in the simulation. The mismatch will be addressed shortly.

Remote Simulation Job Submission via SSH

The IMEX-GAP link DLL (IMEX-2007-11 and later) is able to submit simulation jobs to network computers via Secure SHELL (SSH). Here the network refers to the intranet/LAN and simulator refers to IMEX-2007-11 or later. The required environments 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 data set exists. IMEX and 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 based computers need to install third party software to run 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, a Windows based program. There are quite a few software vendors provide SSH clients on 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 1: After the installation of CopSSH, the user may need to edit the “path” of the Windows environment variable to add the SSH bin directory, for example: C:\Program Files\copSSH\bin.

Note 2: The installation of CopSSH, will install both the server and client software on the users PC. The user may want to disable the CopSSH server service as only the client software is required. This service is called Openssh SSHD Service.

SSH Settings

There are various approaches that can be used to enable SSH communication. The following descriptions are intended to give an example of one approach which was used in our testing process. For technical details, user needs to refer to the SSH software manual.

The ultimate object of the SSH settings is to allow user to invoke remote simulation by issuing an one line “ssh” command at a local command console. For example, the following command line should directly start the IMEX with the data set, dataset1.dat,

```
ssh -l username_on_server simsvr d:\cmgexe\imex.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, for example id_rsa and id_rsapub, should be generated into the folder .ssh, which is created under the user's home directory of the local Windows. An empty “passphrase” is suggested when doing key pair generation. If the key pair is made with a passphrase, the passphrase will be asked for each time when SSH attempts to login.

Registering Public Key in the Server

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 / Edit/View settings...” window. Under the “Settings” tree view, click “Access Control/Windows accounts” and add the user and then import the user's public key.

When initially setting up the newly installed WinSSHD, the system administrator also needs to register the network domain name. That is, under the same “Settings” tree view, choose “Domain order” and click the “Add” 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:

```
ssh simsvr set
```

If the above command gives out the user environment variables in “simsvr”, SSH has been successfully set up. Here, it is assumed that in the above command user's login name is the same on both the server and the local computer.

If the above test is the first time that the user tries to link to the SSH server, SSH will indicate that it cannot establish “authenticity” and ask if the user wants to continue. Here, the user only needs to reply ‘yes’ so as to allow SSH to import the host key (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=/\$HOME/.kshrc, into the file (assuming the 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.

Above settings make the \$HOME/.kshrc be read when SSH does login.

Alternatively, user can also directly edit the ssh command line to make the shell startup file be read. Suppose the ssh command is the next line:

```
ssh simsvr \" /cmg/...
```

The command can be modified as:

```
ssh simsvr ". $HOME/.kshrc && \" /cmg/..."
```

The above command forces the \$HOME/.kshrc to be read before the simulation is executed. This also provides a workaround if the setting of the ENV=\$HOME/.kshrc does not work.

Another alternative method to let ssh read parameters CMG_HOME, LSFORCEHOST and LD_LIBRARY_PATH is to provide these parameters directly into the \$HOME/.ssh/environment file.

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,

```
case $(hostname) in
simsvr | simsvr1 ) LD_LIBRARY_PATH=/usr/cmrg/imex/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 work folder

It is recommended to locate the work folder on the local disk space of the machine on which the simulation runs. This method simplifies file access between IMEX and Resolve. The work folder can also be made on a disk controlled by a third party file server (by this we mean a disk not on either the computer running IMEX or the computer running RESOLVE).

It has been confirmed by tests that the file server can be either a Windows or Linux computer.

A “Local path to work folder” is required in the CMG case setting window, which tells Resolve how to access to the work folder. In the entry, the path should be addressed simply and directly.

Known Problems and Workaround

Samba Server

If IMEX is running in a UNIX computer and the work folder/directory is located on a drive on that computer or a third party UNIX file server, the UNIX of the folder should run Samba server letting Resolve be able to “see” the folder, it is recommended to stop the “opportunistic file locking” for those 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/*.LResolve/*.LSGem/*.LDImex/*.LDResolve/*.LDGem/
```

(E., Robert *et al*: Using Samba, Chapter 5). The user needs to restart the Samba demon to make sure the altered smb.conf is used.

Linux Simulation Run with Work Folder on a Windows Computer - Using Samba Client (smbfs)

Linux machine accesses the work folder via the smbfs file system. That is the share with the work folder is mounted in Linux with -t smbfs option. For example,

```
mount -t smbfs -o username=cmguser //win64/d /cmg/win64
```

Due to Windows file locking, IMEX and Resolve may lose communication. If this occurs, the user needs to check how the “Local path to work folder” is defined. The path should start with the server-name of the Linux platform. In the above example, suppose the Linux computer is named Linux_CPU, the path should be \\Linux_CPU\cmg\win64\.

IMEX Results Accessible from RESOLVE Script

IMEX variables that are accessible from RESOLVE scripts via DoGet() are described below. All specifications are case sensitive.

Use:

```
DoGet("specification")
```

Total Numbers

Well count

instance_name.nWell

Well layer count

instance_name.Well[{well_name}].nLayer

or

instance_name.Well[{#well_number}].nLayer

Group count

instance_name.nGroup

Sector count

instance_name.nSector

Example:

```
DoGet( "IMEX.nWell" )
DoGet( "IMEX.Well[{OIL1}].nLayer" )
DoGet( "IMEX.nGroup" )
```

Well Variables

Format of specification:

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 IMEX. Please refer the IMEX user manual.
The character # is placed before well_number to signal a number not a name is being entered.

Example:

```
DoGet( "IMEX.Well[{OIL1}].OilRatSC" )
DoGet( "IMEX.Well[{#1}].OilRatRC" )
```

Well Variable List

Variable name	Descriptions	Units/Remarks	
WelTyp	Well type; Indicates for producer or injector.	value =	1: producer -1: injector
WellLnk	Well link state. Indicates the well is linked to the GAP surface network or not. Please note at the moment of initial, all IMEX 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: Above well rates represent average rates.

Layer Variables

Format of specification:

instance_name.Layer[{well_name}][{layer_number}].layer_variable_name

or

instance_name.Layer[{#well_number}][{layer_number}].layer_variable_name

Note: *well_number* is the well number used in IMEX. Please refer the IMEX user manual. The character # is placed before *well_number* to signal a number not a name is being entered.

layer_number is the input sequence number of the layers given by the IMEX perforation definition. *layer_number* can also be a range of layers, expressed as *layer_num1 : layer_num2*, where *layer_num1* is less or equals to *layer_num2*.

IMEX perforations are defined via PERF or PERFV keywords within the IMEX data set.

Example:

```
DoGet( "IMEX.Layer[ {OIL1} ][{5}].WatRatSC" )
DoGet( "IMEX.Layer[ {#1} ][{5}].GasRatRC" )
```

Layer Variable List

Variable name	Descriptions	Units/Remarks		
LaySta	Layer status; Open, shut in or closed (not reopenable)	value =	1: Open 0: Shut in -1: Closed	
Depth	Block center depth of the layer	ft		
PrsWlb	Wellbore pressure at the depth of the layer	psig		
PrsLay	Layer block 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		

Note: Above layer rates represent average rates. A negative layer rate indicates the layer is backflowing with respect to its well type: for a producer this would be an injecting layer or for an injector this would be a producing layer.

Group Variables

Format of specification:

instance_name.Grup[*{group_name}*].*group_variable_name*

Note: If there is no user group defined in the IMEX data set, group name FIELD is used to represent all the wells defined in the IMEX data set. Otherwise, a user given top node name can be used.

Example:

```
DoGet( "IMEX.Grup[{Gather-1}].OilRatSCPrd" )
DoGet( "IMEX.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
WatRatSCIj	Water injection rate at stock-tank condition	STB/day
GasRatSCIj	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 of specification:

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 0 is used to represent all active blocks of the reservoir model. Please refer the IMEX user manual.

Note: *sector_number* is the sector number used in IMEX. Please refer to the IMEX user manual. The character # is placed before *sector_number* to signal a number not a name is being entered.

Example:

```
DoGet("IMEX.Sect[{\#0}].OilInPSC")
DoGet("IMEX.Sect[{\Entire Field}].OilInPSC")
```

Sector Variable List

Variable name	Descriptions	Units/Remarks
PrsTPV	Average pressure over the total pore volume	psig
PrsHPV	Average pressure over the hydrocarbon pore volume	psig
VolTPV	Total pore volume	RB
VolHPV	Hydrocarbon pore volume	RB
OilInPSC	Stock-tank oil in-place	STB
MbOInPSC	Stock-tank mobile 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

Example 1

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 FVFIInj
    if (Resolve.Timestep = 0) then
        FVFProd0 = 1.3
        FVFIInj0 = 1
    end if

    ' Estimate FVF values for producers and injectors
    Qres = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].BHFRatRCPrd")))
    QWsurf = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].WatRatSCPrd")))
    QOsurf = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].OilRatSCPrd")))

    if (Qres > 0 And QWsurf + QOsurf > 0) then
        FVFProd0 = Qres / (QOsurf + QWsurf)
    End If

    Qres = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].BHFRatRCInj")))
    QWsurf = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].WatRatSCIInj")))

    if (Qres > 0 And QWsurf > 0) then
        FVFIInj0 = Qres / QWsurf
    End If

    FVFRatio0 = FVFProd0 / FVFIInj0

    ' Get the cumulative voidages so far
    VoidProd0 = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].BHFCumRCPrd")))
    VoidInj0 = Abs(CDbl(DoGet("IMEX.Grup[{FIELD}].BHFCumRCInj")))

End If
```

Example 2

Set the field oil production constraint to 1/1000 of current field mobile oil in-place and also make the constraint be limited between 7.5 MBBL and 15 MBBL:

```
'Get IMEX estimated field mobile oil in-place'
Qprod = Abs(CDbl(DoGet("IMEX.Sect[\#0].MbOInPSC"))) * 0.001

If (Qprod > 15000) Then
    Qprod = 15000
elseif (Qprod < 7500) Then
    Qprod = 7500
End If

'Set the constraint in the production system
Call DoSet("GAP.MOD[\{PROD\}].SEP[\{Sep1\}].MaxQoil", Qprod)
Call LogMsg("max. Oil set to: " + cstr(Qprod))
```

Operation on Well Layers via RESOLVE Script

Since the IMEX release 2008.11, well layers are available to shut in, reopen or do workovers (change the well index of layer) via Resolve scripting (assuming your IMEXGAPLINK.DLL supports this feature). Layer operations are only available for those wells that are linked to the GAP model via Resolve.

Operations are specified via DoCmd() or DoSet(). In following descriptions, specifications are case sensitive.

Layer Open/Shut Operation

Use:

DoCmd("specification")

Format of specification:

instance_name.Layer[well_name][layer_number].layer_operation

or

instance_name.Layer[#well_number][layer_number].layer_operation

Note: *well_number* is the well number used in IMEX. Please refer the IMEX user manual. The character # is placed before *well_number* to signal a number not a name is being entered.

layer_number: same as described in accessing layer variables.

layer_operation can be Shut or Open.

Example:

```
DoCmd( "IMEX.Layer[{OIL1}][{5}].Shut" )
DoCmd( "IMEX.Layer[{#1}][{3:5}].Shut" )
```

Remarks:

Shutting the final opened layer of a well will shut the well at same time.

Note that using a script to open a layer of a well which was shut because all layers were shut previously in a Resolve script opens that well.

Note that using a script to open a layer of a well which was shut using AUTO layer status (in the IMEX *PERF definition) opens that well.

A well cannot be opened using a Resolve script if it was explicitly defined as *SHUTIN in the IMEX recurrent data.

Layer Workover via DoSet

Use:

DoSet("specification", value)

Format of specification:

instance_name.Layer[*{well_name}*][*{layer_number}*].PIMult
or
instance_name.Layer[*{#well_number}*][*{layer_number}*].PIMult

Note: *well_number* is the well number used in IMEX. Please refer the IMEX user manual. The character # is placed before *well_number* to signal a number not a name is being entered.

layer_number: same as described in accessing layer variables.

value is the multiplier to the current PI of the specified layer(s).

Example:

```
DoSet("IMEX.Layer[{OIL1}][{1:5}].PIMult", 2.0)
```

Layer operation commands should be placed in the PreSolve section of the script for the IMEX module in order to have an immediate effect (synchronized with the results that IMEX reports for that time).

For example, the following script (placed in the PreSolve section of the instance "IMEX-1") will shut layer 5, 6 and 7 of well no. 3 of the reservoir "IMEX-1", at the same moment when water cut becomes higher than 60%:

```
Sub Resolve_Presolve(ModuleList)

If (InStr(ModuleList, "IMEX-1") > 0) Then
    ...
    QW = Abs(CDbl(DoGet("IMEX-1.Well[{#3}].WatRatSCPrd")))
    QO = Abs(CDbl(DoGet("IMEX-1.Well[{#3}].OilRatSCPrd")))
    If (QW/(QO+QW+1.0E-12) > 0.6) Then
        DoCmd("IMEX-1.Layer[{#3}][{5:7}].Shut")
    Endif
    ...
End If

End Sub
```

Layer operations placed in any other portions of the script will not have an immediate effect. So if the above script is not placed within the PreSolve section or in the scripting section of a module other than "IMEX-1", the scripted layer operation will not take effect immediately but will take effect at the next GAP specified timestep after instance IMEX-1 reports the water cut is higher than 60%.

Pseudo-Solvent Production / Injection

Activate: The special link will be activated by the IMEX command-line option ‘-ssolvent’.

IPR: IMEX generates IPR with solvent rate in addition to the oil, gas and water rate. However the IPR written to RESOLVE is in the Black-Oil format, where for producers, the ‘gas’ rate is the sum of HC (Hydro-Carbon) gas rate and solvent rate. The total gas gravity is therefore the volume-weighted gravity of the total gas mixture at Bottom-Hole pressure. For injectors, the gas rate is either the rate of HC gas or solvent depending upon the type of the injector defined in IMEX with *INCOMP keyword.

Set-Point: Set-point received from Resolve includes bottom-hole-pressure (Pwf), oil, gas and water rates. Generally, IMEX determines the well constraint by choosing a rate from the set-point. The determination is based on the well type and the well’s primary (the first) constraint defined in IMEX dataset.

- Producer

With this special link, the total gas rate received from a set-point will be decomposed into HC gas rate and solvent rate based on the IPR of the well. Among the oil, gas, water and solvent rates, following logic is applied: (1) If the producer’s primary constraint is STO, STG or STS, the corresponding rate from the set-point will simply be selected; (2) If the producer’s primary constraint is not a Sock-Tank rate, for example the Bottom-Hole-Pressure, IMEX will check the mass rates of oil and total gas. If the mass rate of oil is greater than 1/10 of the mass rate of the total gas, the oil rate will be selected to make the STO constraint. Otherwise depending on the larger volume rate between HC gas and solvent, the STG or STS constraint will be made (*note).

- Injector

With this special link, the total gas from the set-point will be assigned to STC for HC gas or STS for solvent depending upon the injector-type defined in IMEX dataset.

*note: The decomposition of total gas is based on the assumption that the set-point is solved on the IPR that IMEX passed to Resolve. If IMEX detects the total gas rate of a set-point has deviated from the IPR, the decomposed gas rates will not be used. In this situation, if the constraint determination has to choose HC gas or solvent gas, the constraint will be switched to BHP control using the set-point Pwf.

Appendix I

Coupling IMEX to FORGAS

IMEX can be linked directly to SPT Group's FORGAS surface network model.

IMEX and FORGAS each have their own input and output files. IMEX and FORGAS 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 an IMEX data set must be able to run without the network model. All necessary sources and sinks (wells) should be defined appropriately.

IMEX passes an IPR table (Inflow Performance Relationship - a table of bottom-hole flowing pressures against phase rates) for each well to FORGAS.

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 bottomhole conditions. Thus the network model must handle flow in the wellbore.

Some or all IMEX wells can be linked to the network model through the controlling interface (FORGAS). IMEX generates IPR tables for those wells that are linked to the network model and in turn, gets instructions from the controlling interface (FORGAS) on how to control those wells.

In general, IMEX wells linked to the network model have their operating constraints and schedules described in the network model. On the other hand, wells within the IMEX model that are not linked to network model have their operations controlled by IMEX.

IMEX can also be made to control certain aspects of a linked well's operating condition. Specifically for linked wells:

- Any well monitor defined via the *MONITOR keyword is allowed.
- Keywords that control well and layer operations within a *TRIGGER are allowed.

Details are given in the ‘Recurrent data’ section of ‘Preparation of IMEX 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.

IMEX is connected to the network model on a per well basis, using IPR curves.

There should be no IMEX well control keywords for network model linked wells besides the initial well definition (see Preparation of IMEX Data Set).

Normally FORGAS controls the timestep the coupled model takes; however, IMEX can be made to provide input to FORGAS about timestep selection. FORGAS will then select the minimum of the network model and IMEX timesteps.

If the time to the next network model specified network solve date is too large, it is possible that an IMEX well’s bottom-hole pressure may vary considerably while timestepping to this date. Because IMEX 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 IMEX start date defined in the recurrent data set should be equal to the network model forecast start date. If the reservoir started production before the desired forecast start date, run IMEX alone to create a restart record on the desired forecast start date. Use that restart record for the runs using the combined models. 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 IMEX Data Set

There are no keywords required for IMEX to couple with the network model. But as listed below, some restrictions and changes apply.

The forecast start date of IMEX can not be earlier than the start date of the network model time schedule. The IMEX start date can be an IMEX restart start date.

Recurrent Data

The following well data restrictions only apply to IMEX-Network model linked wells.

Well Definition

All IMEX-Network model linked wells should be defined once at the start time of recurrent data. No further well keywords are necessary in the IMEX recurrent data after the start date. Wells which cycle between injectors and producers should be defined as two wells in both the network and IMEX models.

Well SHUT/OPEN Status

IMEX passes open/shut-in status for linked wells to the surface network at each surface network date. After receiving this well status information from IMEX, the surface network honors those wells which IMEX shuts 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 IMEX. IMEX retains the received well status for the IMEX timesteps required to reach the new network date.

IMEX 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 IMEX or the surface network can shut-in a well. However, both IMEX 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 IMEX constraint from the three phase rates received from surface network. If the first or primary constraint happens to be other than a rate constraint, IMEX 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 IMEX 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 IMEX 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 IMEX timestep and cause rate mismatch between IMEX 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

IMEX and the surface network ONLY exchange information at each network timestep. FORGAS automatically uses ALL of the IMEX dates.

Monitor:

For well groups defined in IMEX, 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.

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 IMEX 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 IMEX and FORGAS are synchronized. The user must place all well control keywords within triggers, because in linked runs trigger keywords are checked at only 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'  
treld > 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 the “*ON_ELAPSED ‘TIME’ treld > 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'  
treld > 0.0  
*SHUTIN 'PRODUCER-2'  
*END_TRIGGER
```

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.

IMEX 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 an IMEX run coupled with a 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 time step, the FORGAS down-time percentage is passed to IMEX as an on-time fraction. IMEX uses the received on-time fraction as if the same value was input through the IMEX recurrent data. Essentially the definition/handling of on-time fraction for coupled wells is moved to the FORGAS network.

For unlinked wells, on-time fraction should be given in IMEX recurrent data as usual.

IMEX passes back on-time fraction for all wells to FORGAS.

DATE and TIME

*DATE and *TIME keywords can normally be used within the IMEX data set. These can be used to obtain IMEX output when the *WPRN/*WSRF output options are used.

The forecast start date of IMEX must be the same as the forecast start date of the surface network date schedule. Use a restart record to adjust the IMEX forecast start date to correspond to that of the surface network model.

FORGAS will honor the IMEX stop date; the earliest stop date (between that specified in FORGAS and that specified in IMEX) will be used.

Multiple Reservoirs

The surface network model can connect to multiple IMEX reservoir models. FORGAS requires all IMEX files to have same start date.

IMEX and the network programs communicate through ASCII files. Four ASCII files per IMEX instance are created and updated during the linked run. The files are named using the root IMEX input data file name and the extension related to their use within the linked model (LSResolve, LDResolve, LSIMEX, and LDIMEX). These files must not be altered by the user during a linked model run.

Remote Simulation Job Submission via SSH

The IMEX-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:

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.

Both local and remote computers should be able to create/read/write files in the folder/directory where the simulation dataset exists. IMEX and FORGAS 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 a 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 FGI for the FORGAS link, a Windows based program. 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 IMEX with the dataset, dataset1.dat,

```
ssh -l username_on_server simsvr d:\cmgexe\IMEX.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 nonblank 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:

- In user’s \$HOME/.ssh directory, create a file with name environment. Add one line,
`ENV=.kshrc`, into the file (assuming Kshell is used).
- 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,

```
case $(hostname) in
simsvr | simsvr1 ) LD_LIBRARY_PATH=/usr/cmeg/imex/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 IMEX 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 FORGAS 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 of alternating text file communication between IMEX and FORGAS, there can be file sharing problems caused by Samba “opportunistic file locking”. The workaround is to switch off the file locking for all IMEX/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/*.LResolve/*.LSIMEX/*.LDImex/*.LDResolve/*.LDIMEX/
```

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, IMEX and 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 IMEX and FORGAS in addition to performing various other functionalities required for the running of the network model. The user provides the name of the IMEX input file as well as the name and location of the IMEX executable in the FORGAS data preparation program, FGI. In the FORGAS input data file, shared FORGAS/IMEX wells should belong to an “IMEX” reservoir. For each “IMEX” reservoir defined in FORGAS, the user selects the name of the IMEX input data file and 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.

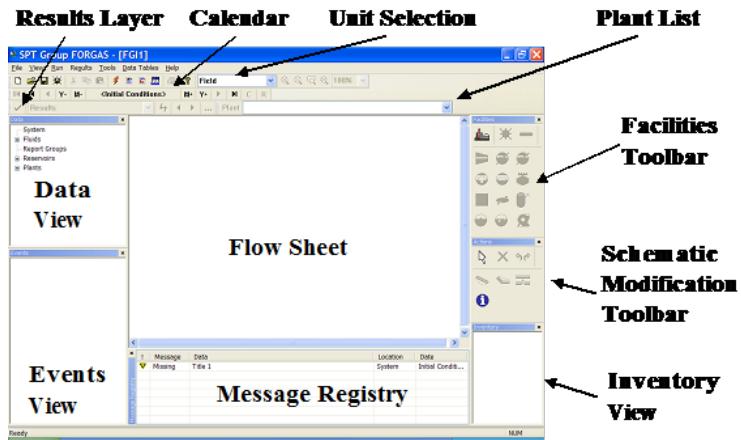
1. FORGAS will automatically start the specified IMEX executable using the file names of the specified IMEX input data files. The IMEX log file will appear within a DOS window, which the user can view as the run progresses.
2. FORGAS will wait while IMEX initializes its calculations and provides the inflow performance data to FORGAS for each well defined in IMEX.
3. IMEX 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. IMEX 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 IMEX. Thereafter, the recommended timestep from IMEX 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 IMEX, IMEX will take as many timesteps as required to move forward the required number of days. When there are multiple IMEX input data files coupled to FORGAS, FORGAS will look at the recommended timestep from each IMEX file, and will use the smallest of each to determine the next timestep to take. FORGAS will pass that timestep length to each IMEX file. Repeating FORGAS timesteps is not supported.

4. After the initialization phase, IMEX will wait until FORGAS has finished the calculations for the first timestep. For production and injection wells, FORGAS provides the oil, water and gas flow rates at the set point which are interpolated from the IMEX IPR tables. For each shared well, the well status (OPEN or SHUTIN) is passed by FORGAS to IMEX. The flowing bottom-hole temperature will also be provided to IMEX. 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.
5. Once FORGAS has finished its first timestep, it now waits for IMEX to use the data provided by FORGAS to calculate the conditions at the start of the next FORGAS timestep. IMEX now reads the data provided by FORGAS to determine which of its wells are shared with FORGAS. IMEX performs its timestep(s) to reach the specified FORGAS timestep. IMEX calculates the inflow performance of each well and passes that to FORGAS. The current date and the recommended length of the next timestep are passed to FORGAS. Then IMEX waits until FORGAS has retrieved and used the data to determine the well flow rates for the next timestep.
6. The forecast will continue until the end date is reached (namely the earliest date of those specified in the FORGAS and IMEX input data files) or there is an error encountered by FORGAS or IMEX.

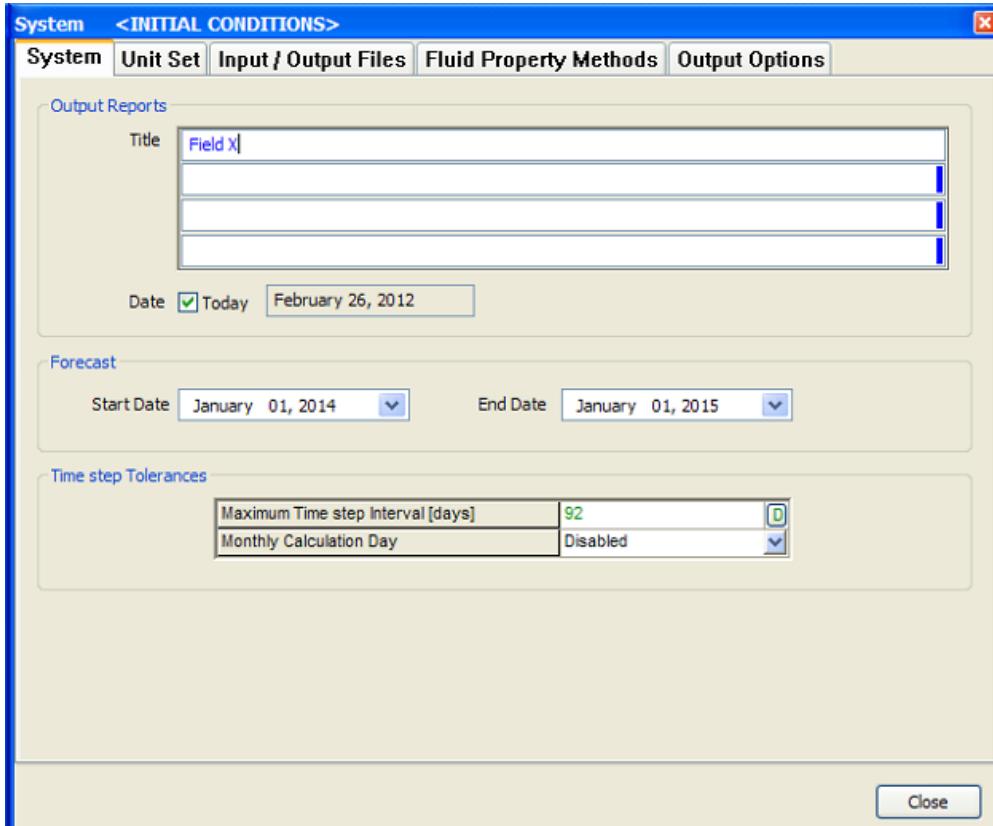
Getting Started Linking IMEX and FORGAS

Start FORGAS. Note the various areas of the screen, as described below.

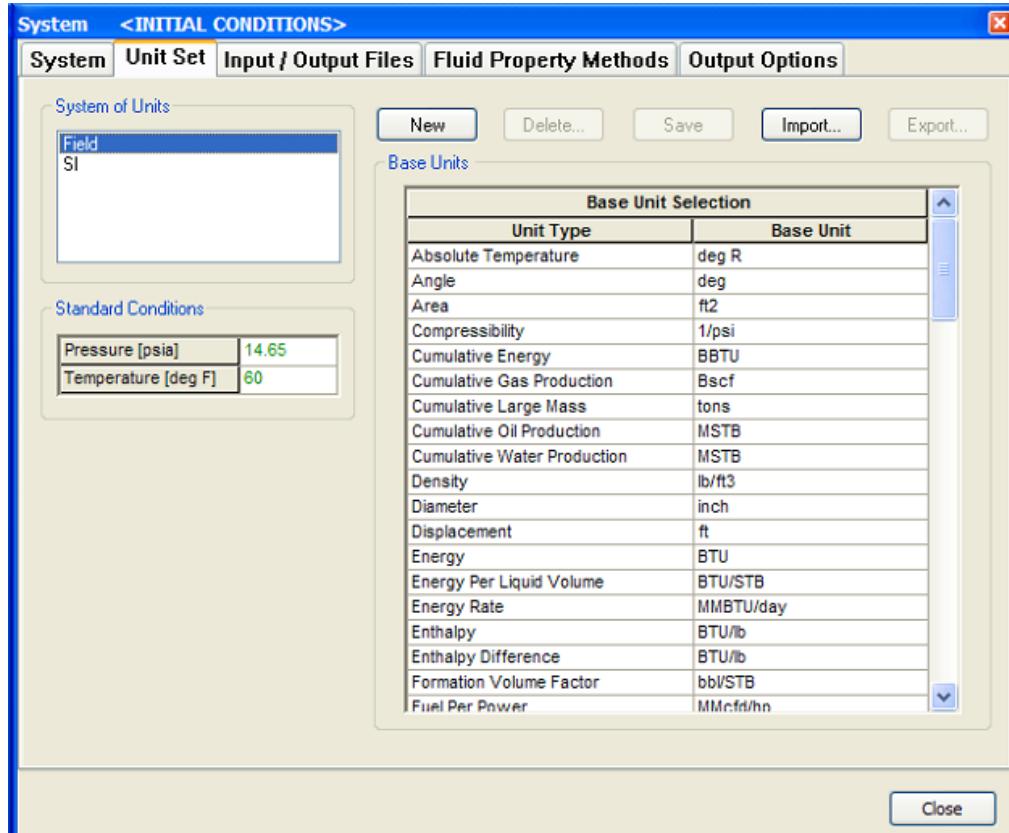


Select File => Save As and specify an appropriate file name. **FORGAS** will automatically append the extension “.fgw”.

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 **IMEX** must start their forecast on the same date. Use a restart record in **IMEX** to align the start dates. Enter the Forecast End Date. The **FORGAS/IMEX** run will end on the earliest of the **FORGAS** and **IMEX** end dates. The screen will appear similar to that shown below:



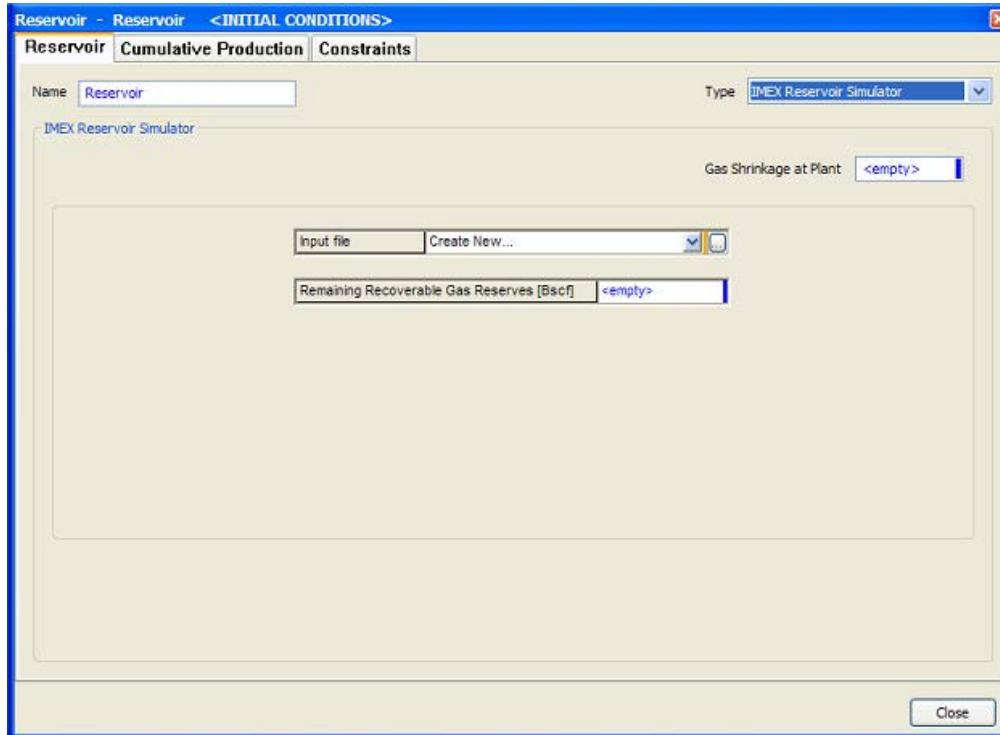
Click on the Unit Set tab and select the appropriate unit set, as shown below. It is not necessary to select the same unit set as **IMEX** uses, since the parameters passed between **IMEX** and **FORGAS** are always converted to imperial (field) units in their communication files.



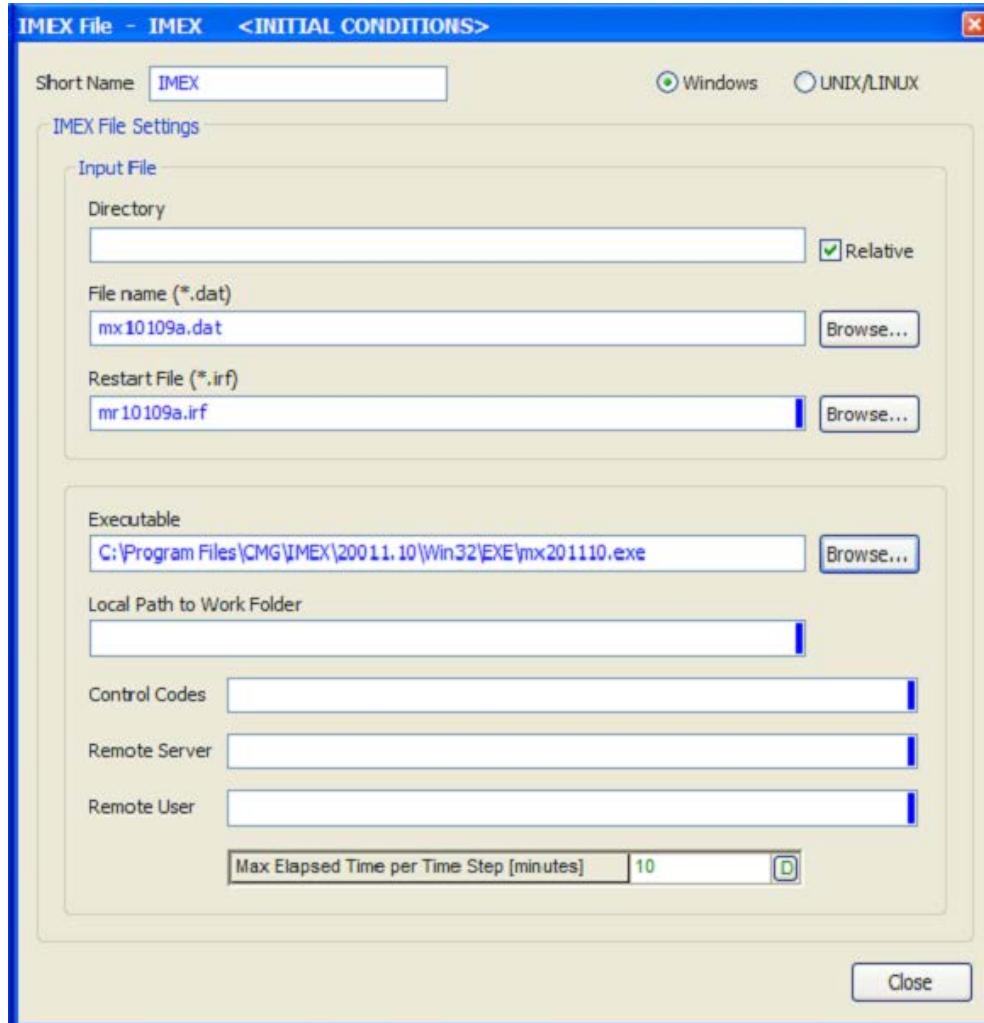
If desired, you can select the other tabs to select different options than the provided defaults for file creation, equation of state models and output tables. Close the System panel.

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.

Type in the appropriate name for the reservoir. For the reservoir Type, select IMEX 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.

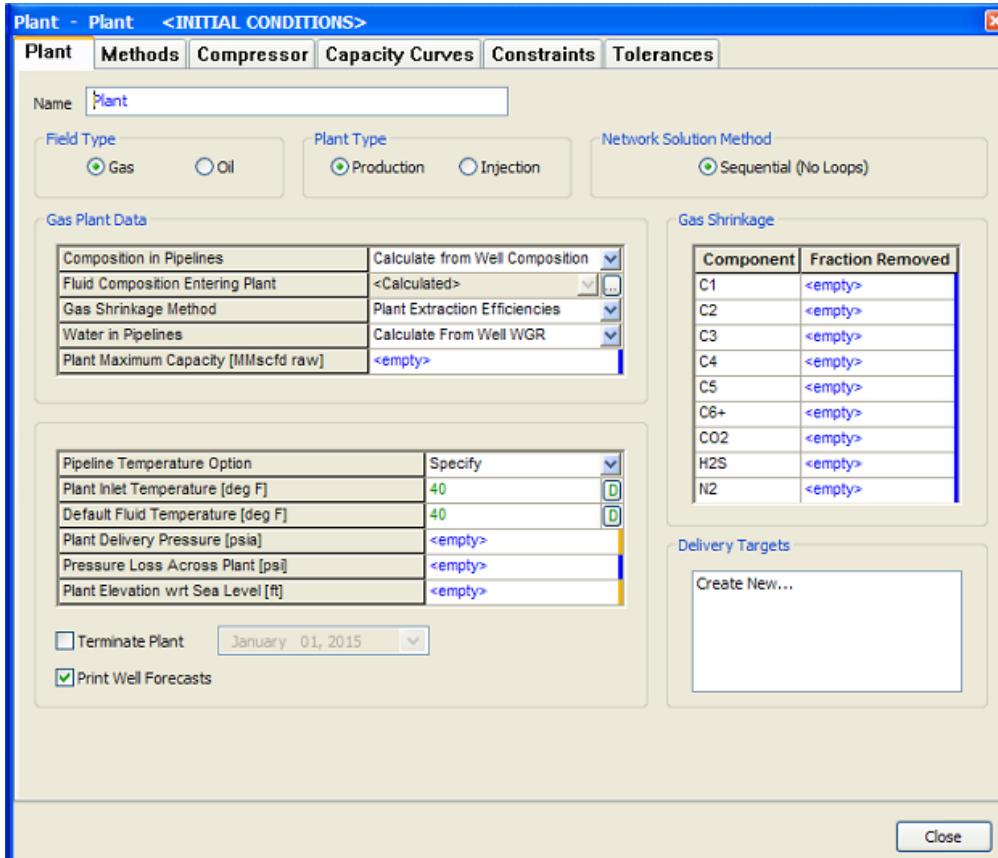


Click on the after Input file Create New... ... to specify the **IMEX** input data file name. On the panel which appears, select the **IMEX** input data file. If required to align the start dates with **FORGAS**, choose the **IMEX** restart .IRF file. Use the Browse button to find the location of the **IMEX** executable to be used. If running the **IMEX** data file on a different computer than being used by **FORGAS**, specify the appropriate Remote Server information (see above for further information). The panel should appear similar to that shown below:



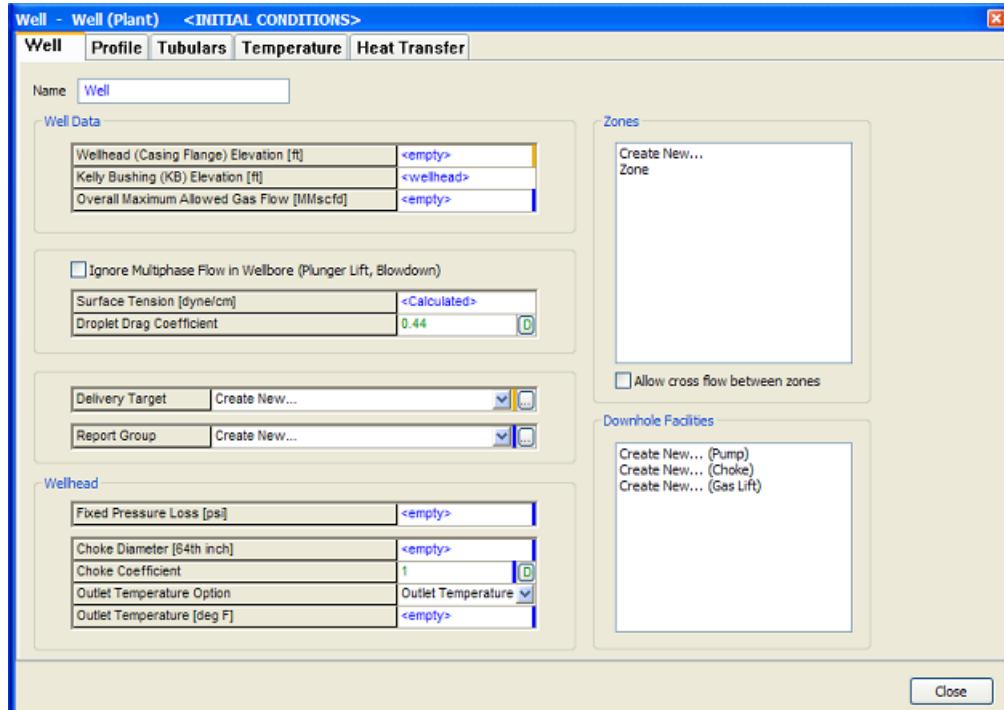
Select Close twice to close the reservoir data.

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).



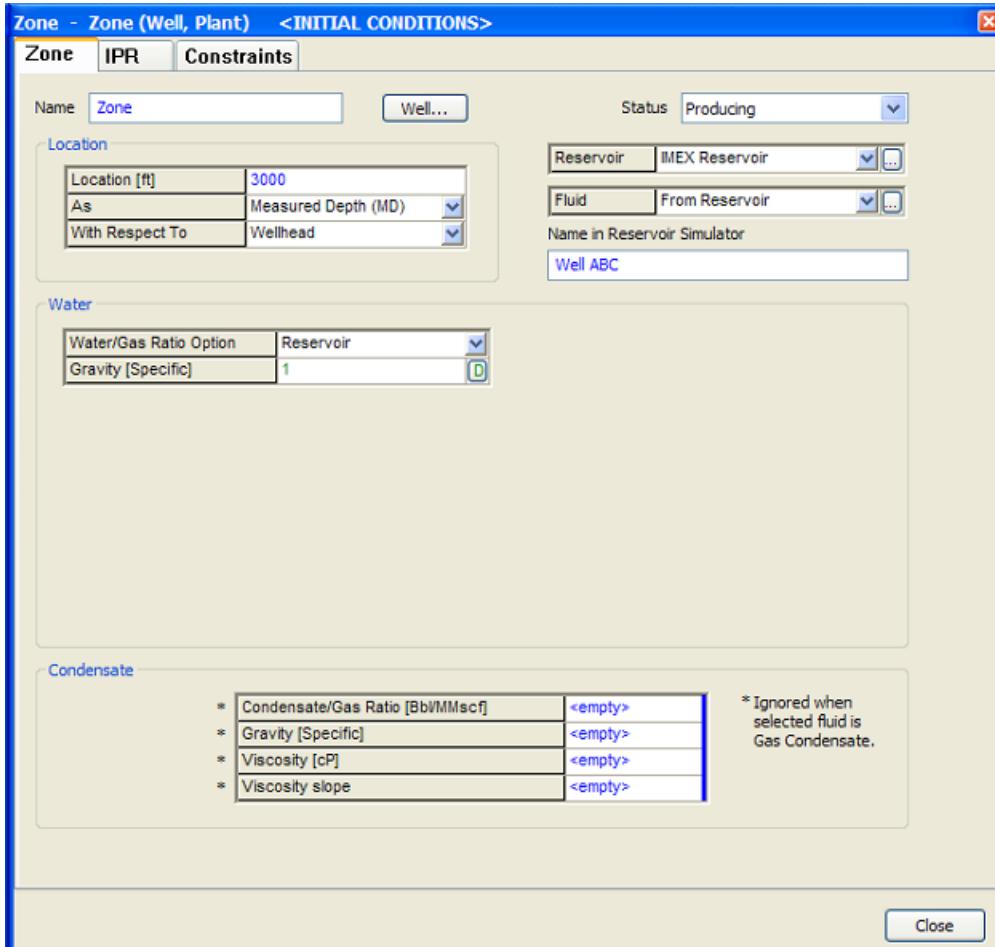
After specifying all of the plant data (select each tab), close the plant data.

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:

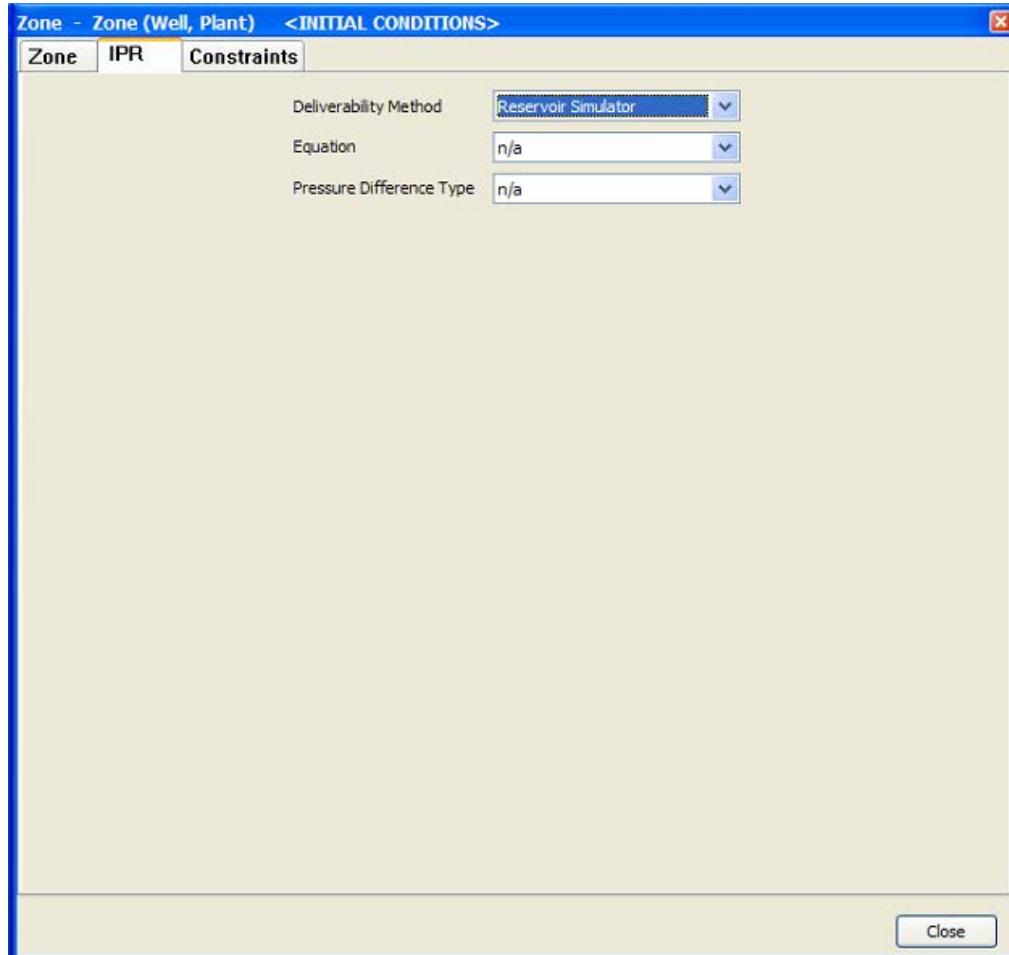


Zone is the term which refers to the perforations where the well connects to the reservoir. **FORGAS** automatically creates one Zone for a well. Double click on the word “Zone” in the Zones area. This will open the Zone data panel.

- Select the zone status. In both the IMEX and FORGAS input data files, a well must be given the status of “producer” for FORGAS/IMEX 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 IMEX. To use the amount of water predicted for this individual well by IMEX, select Reservoir for the Water/Gas Ratio Option.
- Fluid gravities will automatically be supplied by IMEX. You can either use these provided oil, water and gas gravities by leaving the Fluid selection as “From Reservoir” or you can specify fluid compositions directly in FORGAS. For gas systems, it is recommended that you specify the fluid compositions in FORGAS.
- Type in the well name in the **IMEX** input data file for the Name in Reservoir Simulator. It is important that this name exactly matches the well name in the **IMEX** input data file; otherwise the forecast will be terminated.
- The panel should appear similar to the following:



Select the IPR tab. IMEX will always provide the inflow performance data for each well. Select Reservoir Simulator for the Deliverability Method, as shown below:



Specify the rest of the required data for each well, each pipeline and each facility.

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 IMEX log file will appear in a separate DOS window. Note that an "f" alongside the IMEX time step number in the IMEX log file indicates completion of a FORGAS time step.

The results of the run will be available via IMEX's output and SR2 files and through the FORGAS Results menu (View Output Files, View Plots..., Export to Spreadsheet...).

For more details on using FORGAS coupled with IMEX, see the FORGAS Help => User Manual => FORGAS 10: Technical Reference => Reservoir Calculations => IMEX Reservoirs.

More details on using FORGAS can be found in the FORGAS Help => User Manual => FORGAS 10: Getting Started => Using the FORGAS Graphic Interface (FGI).

BOTAPI 534
BWI 563

Keyword Index

3

3PTSCALING 647

A

ADGCSTV 670
ADGMAXV 670
ADGPCRIT 670
ADS_WATERZONE 585
AIM 738
AIMSET 791
AIMWELL 792
ALL 237
ALTER 853
ALTERCP 875
ALTERFS 877
ALTER-HIST 862
API 704
APIGRAD 508
APIMINSO 721
APIT 705
APPOR-METHOD 1013
AQFUNC 431
AQLEAK 424
AQMETHOD 424
AQPROP 424
AQUIFER 424
AUTODRILL 820

B

BG 228
BGUST 539
BHPDEPTH 868
BHPGRAD 870
BHPHEADINIT 812
BINARY_DATA 249
BLOCKDFL 347
BLOCKGROUP 353
BLOCKS 347
BOT 530

C

CASEID 268
CCPOR 451
CHECKONLY 217, 269
CHECKKRB 757
CIRREVERS 460
CLUMPSETTING 1075
CO 561
COMMENT 221
CON 231
CONTACT_PBT_BEHAVIOR 726
CONVERGE 740
COORD 342
CORNERS 344
CORNER-TOL 421
COT 527
CPOR 389
CPRPOR 451
CRNCON_NODE 352
CROCKTAB 452
CROCKTABC 468
CROCKTABD 463
CROCKTABE 461
CROCKTABH 455
CROCKTABR 470
CROCKTABU 465
CROCKTYPE 449
CTYPE 478
CVO 562
CVW 565
CW 563
CYCPRT_END 996
CYCPRT_START 996

D

DAMP-PCOW-TROL 626
DATE 787
DATUMDEPTH 712
DEBUG 309
DENSITY 559
DEPLETION 458
DEPTH 328
DEPTH-TOP 334
DGOC 709
DI 322
DIFLIB 499
DIFRAC 385
DIM 258

DIP 336
DISPI 485
DISPJ 485
DISPK 485
DJ 324
DJFRAC 385
DK 326
DKFRAC 385
DPBDT 568
DPCONNECT 384
DPDWDT 570
DPLANES 766
DRILLQ 1009
DRILLT 1009
DRSDT 568
DRVDT 570
DTMAX 732, 790
DTMIN 733, 790
DTOP 330
DTWAUTO 789
DTWELL 789
DTYPE 769
DUALPERM 372
DUALPOR 371
DWGC 709
DWOC 709
DYNAGRID 1077
DYNAGRID-WLN-A 1077
DYNAGRID-WLN-V 1077

E

EGUST 539
END_TRIGGER 1026
EPSPC 618
EPSPCG 618
EQUALSI 230
EQUILIBRATE 818

F

FAULT 422
FAULTARRAY 447
FILENAME 252
FILENAMES 252
FLUX_ISECTOR 438
FLUX_SECTORNAMES 438
FORCE_MAXCHK 747
FRACTURE 226
FRACVOL 375
FRWIDTHCOR 659
FZ 238

G

GAPPOR 1011
GASZONE 718
GCIOFF 1024
GCION 1024
GCONCYCLE 994
GCONCYCLE_END 982
GCONCYCLE_START 982
GCONCYCR_END 992
GCONCYCR_START 992
GCONI 936
GCONIINCR 1050
GCONIMULT 1050
GCONM 979
GCONP 930
GCONPINCR 1048
GCONPMULT 1048
GCPOFF 1024
GCPON 1024
GEOMETRY 897
GFUELFrac 955
GFUELMAXR 957
GLCONTROL 892
GLIFT 889
GLOPT 894
GMKMAXR 947
GMKUPTO 945
GOC_PC 715
GOC_SW 719
GORINT 572
GPHYDDEP 1004
GPRODGROUP 949
GPTABLE 1002
GRAVITY 559
GRECYMAXR 953
GRID 313
GRIDPAR 768
GROUP 799
GROUPALQ 1006
GSALESFRAC 959
GSALESMAXR 961
GUIDEI 1021
GUIDEP 1021

H

HEADITER 810
HEAD-METHOD 808
HYSKRG 1, 624
HYSKRO 621
HYSKRW 621

I
IJK 232
INCOMP 833
INCOMPWL 855
INITIAL 678
INJECTOR 816
INT 243
INTER_REGION_TM 413
INTERRUPT 270
INUNIT 275
IRCONNECT 442
ISECTOR 436
ITERMAX 755
ITUBE1 1058
ITYPE 688
IVAR 234
IWELLBORE 824

J
JACDOMS 770
JACPAR 772
JFGMIN 644
JFUNC 662
JFW_SHIFT 654
JFWMAX 644
JVAR 235

K
KDIR 320
KILLOUGH_MOD_KRWCO 621
KRGCL 641
KRGRL 641
KROCRW 641
KROCW 641
KROGCG 641
KROGCRG 641
KROI 629
KRPERF 925
KRWEXTRAP 653
KRWIRO 641
KRWRO 641
KVAR 236

L
LAYERCLUMP 1068
LAYERGRAD 872
LAYERHFP 920
LAYERIJK 918
LAYERLUMP_ACTR 1076
LAYERRLUMP 1071

LAYERXYZ 914
LAYERXYZ-METHOD 914
LEASE 440
LIST 219

M
MANIFOLD 998
MATINV 775
MATRIX 225
MAXCHANGE 734
MAXCPU 731
MAXERROR 280
MAXSTEPS 730
MINC 379
MINCYC 744
MINSS 574
MOD 240
MODEL 489
MODELSHUT 818
MONITOR 846
MRC-RESET 887
MXCNRPT 845

N
NCUTS 748
NDARCYCOR 658
NETGROSS 404
NETPAY 402
NEWTONCYC 743
NFZ 238
NNODES 347
NODES 347
NODEX 347
NODEY 347
NODEZ 347
NOLIST 219
NOLISTLIM 220, 274
NONDARCY 655
NORM 734
NORTH 751
NREGIONS 687
NULL 386
NULL-PERF 814
NUMERICAL 729
NWHYFS 1066

O
OILTRAP 627
OMEGA_OST 556
OMEGASG 573
ON-TIME 1054

OPEN 820
OPERATE 835
OPERATE-HIST 860
OUTDIARY 306
OUTPRN 289
OUTSRF 299
OUTUNIT 278

P

PADSORP 576
PAYDEPTH 332
PB 689
PBS 699
PBT 690
PCGIMIN 644
PCGMAX 644
PCGMIN 644
PCON-MANIFOLD 1000
PCOW_SHIFT 654
PCSCALE_2PT 650
PCWIMIN 644
PCWMAX 644
PCWMIN 644
PDEGAA 758
PDEGAB 759
PDEGRBB 760
PDEW 694
PDEWT 695
PDRAIN-METHOD 796
PERF 900
PERFV 900
PERMI 400
PERMJ 400
PERMK 400
PHS_SEG_MODL 672
PINCHOUTARRAY 416
PINCHOUT-TOL 419
PIVOT 754
PLNRFRAC 367
PLNRFRAC_TEMPLATE 367
PMIX 578
PNTHRDS 767
POLYCONC 702
POR 388
PPATTERN 761
PPERM 583
PRECC 750
PREFCONC 582
PRES 684
PREST 685
PRIOR-FORM 1017
PRODUCER 816
PRPOR 389

PSPPLIT 308
PTHRESHI 660
PTHRESHJ 660
PTHRESHK 660
PTUBE1 1058
PTYPE 567
PVCUTOFF 418
PVISC 581
PVT 494
PVTAPI 505
PVTCOND 516
PVTG 513
PVTS 552
PVTVO 521
PWELLBORE 827

R

RANGE 355
RANGECHECK 279
REFDEPTH 707
REFINE 355
REFPRES 707
REFPST 551
REFPW 563
RELAX 736
RELTOL 749
REPORTING-GROUP 1052
RESDATE 281
RESTART 281
RESTART_SR2 283
RESTIME 281
REWIND 284
RG 227
RM-LAYERCLUMP 1073
RM-LAYERRLUMP 1074
ROCKDEN 670
ROCKFLUID 590
RPT 591
RST 692
RTYPE 632
RUN 786
RVF 697

S

SCLDMG 667
SCLDPS 666
SCLRMV-LAYER 883
SCLRMV-WELL 882
SCLTBL-LAYER 880
SCLTBL-WELL 879
SCONNECT 445
SD_REINF 668

SDEGREE 753
SEAWATFRC 703
SECTOR 433
SECTORARRAY 435
SECTORNAMES 436
SETPI 864
SGCON 633
SGCRIT 633
SGT 607
SHAPE 380
SHUTIN 820
SLCON 633
SLT 607
SMKMAXR 973
SMKUPTO 971
SO 700
SOIRG 633
SOIRW 633
SOLSTALL 774
SOLVER 756
SONEQ 724
SORDER 752
SORG 633
SORMAX 633
SORW 633
SPRODGROUP 975
SR2ASCII 272
SR2PREC 273
SRECYMAXR 977
SRFTNG 665
SRFTNW 665
STOP 1080
SUBDOMAIN 373
SVISC 584
SW 701
SWCON 633
SWCRIT 633
SWINIT 722
SWNEQ 724
SWT 594
SWTI 605

T

TARGET 858
TBLBAS_3PTKR 652
TIME 788
TITLE1 265
TITLE2 266
TITLE3 267
TRANLI 408
TRANLJ 408
TRANLK 408
TRANS_MULT_REGION 415

TRANSD 377
TRANSF 487
TRANSFER 382
TRANSI 405
TRANSJ 405
TRANSK 405
TRANSLATE 222
TRANSMF 410
TRANSMULT 411
TRES 493
TRIGGER 1026
TWELLBORE 832

U

USER_INPUT 679

V

VAMOD 393
VATYPE 393
VERTICAL 679
VGUST 543
VOLMOD 391
VOT 532
VOTAPI 546
VRI_GROUP 951
VWI 565

W

WATER_FIX 745
WCURRCN 857
WELGEO 911
WELL 803
WELLALQ 1008
WELLINIT 794
WIRANGE 909
WLISOPEN 822
WLISHTHUT 822
WMKMAXR 965
WMKUPTO 963
WOC_PC 715
WOC_SW 719
WORKOVER-DATA 884
WPRN 286
WPRODGROUP 967
WRECYMAXR 969
WRST 284
WSRF 297
WTINCR 1045
WTMULT 1045

X

XCORN 340
XDR 299
XFLOW-MODEL 806

Y

YCORN 340

Z

ZCORN 338
ZGUST 539