

---

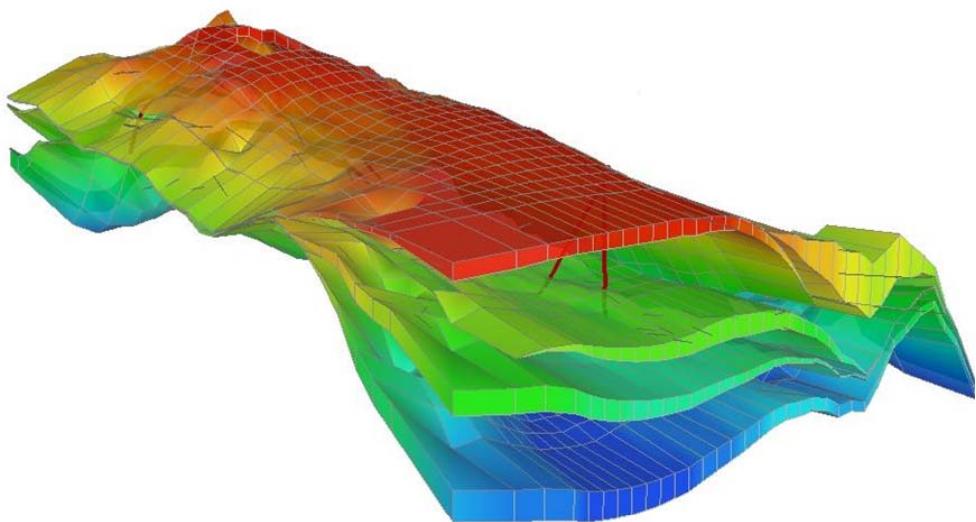
User's Guide

# STARS

***Advanced Process and Thermal  
Reservoir Simulator***

**Version 2011**

**By Computer Modelling Group Ltd.**



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 © 1987-2011 Computer Modelling Group Ltd.**

All rights reserved.

The license management portion of this program is based on:

Reprise License Manager (RLM) Copyright (C) 2006-2011, Reprise Software, Inc.

STARS uses Intel(R) Compilers.

STARS, 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.**

Office #150, 3553- 31 Street N.W.

Calgary, Alberta Canada T2L 2K7

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

# Preface

STARS is CMG's new generation advanced processes reservoir simulator which includes options such as chemical/polymer flooding, thermal applications, steam injection, horizontal wells, dual porosity/permeability, directional permeabilities, flexible grids, fireflood, and many more. STARS was developed to simulate steam flood, steam cycling, steam-with-additives, dry and wet combustion, along with many types of chemical additive processes, using a wide range of grid and porosity models in both field and laboratory scale.

This User's Guide details data entry for simulating the above processes. It requires some knowledge of reservoir engineering and some rudimentary exposure to reservoir simulation. This User's Guide provides a step-by-step procedure for preparation of input data for this program. A tutorial section is provided as well as a set of appendices.

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

**Computer Modelling Group Ltd.**

#150, 3553 – 31 Street N.W.

Calgary, Canada

T2L 2K7

**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

<b>Introduction</b>	<b>1</b>
Overview.....	1
Important Changes between STARS 2011.10 and 2010.10.....	2
Data Incompatibilities with Previous Versions of STARS .....	3
New Keywords Added to STARS 2011.10 .....	4
Enhancements to Existing Keywords .....	6
New and Changed Template Data Sets.....	7
Introduction to STARS .....	8
<b>Tutorial</b>	<b>11</b>
Introduction.....	11
Data Groups in the Keyword Input System .....	12
How to Read Keyword Syntax .....	13
How to Document Your Data Set .....	14
How to Do a Restart.....	15
Controlling Contents of the Output Print File.....	17
Controlling Contents of the Simulation Results File .....	18
Describing Your Grid System.....	19
Specifying Null Blocks .....	20
Describing Refined Grid .....	21
Using Dual Porosity/Dual Permeability.....	22
Problems with Small Time Steps or Long Execution Times .....	23
Defining Wells .....	25
Defining the Well Type .....	26
How to Shut In a Well and Reopen It .....	27
Operating and Monitoring Constraints .....	28
Specifying Well Indices .....	30
Horizontal Wells .....	32
Stopping a Simulation Run .....	33
Guidelines for Setting Up Well Data .....	34
Running Your Simulation .....	35
Improving Numerical Performance .....	37
Optimizing Memory Requirements .....	41
Well Management and Group Control.....	43
Parallel Processing.....	49

<b>Keyword Data Entry System</b>	<b>53</b>
Introduction to Keyword System.....	53
Comments (Optional).....	57
Blank Lines (Optional).....	58
Data Range Checking (Optional) .....	59
Include Files (Optional).....	60
Controlling Data File Listing (Optional) .....	61
Changing the Comment Indicator (Optional).....	62
Changing the Keywords by Using Translate Rules (Optional) .....	63
User Block Address.....	64
Input of Grid Property Arrays .....	66
Entering Matrix Grid Properties .....	68
Entering Fracture Grid Properties .....	69
Entering Refined Grid Properties .....	70
Entering Wellbore Grid Properties .....	71
Assigning Grid Properties to all Elements .....	72
Constant Value Arrays .....	73
Array Input In IJK Notation .....	74
Array Input of Values that Vary in the I Direction.....	76
Array Input of Values that Vary in the J Direction .....	77
Array Input of Values That Vary in the K Direction.....	78
Values that Vary for Most or All Grid Blocks .....	79
Values Stored in Binary Form.....	80
J and K Direction Data from I Direction .....	82
Modifying Array Data (Conditional).....	83
Interpolating Table Data (Optional) .....	86
<b>Input/Output Control</b>	<b>87</b>
Summary of Input/Output Control .....	87
Command-Line Arguments (Optional) .....	92
Input/Output File Names (Optional) .....	95
Dimension Over-Rides (Optional) .....	99
Scan Mode for Checking Errors (Optional) .....	102
Project Main Title (Optional) .....	103
Input/Output Data Units (Optional).....	104
Mass Basis Indicator (Optional).....	107
Maximum Number of Error Messages (Optional) .....	108
Starting Timestep or Time.....	109
Restart Record Writing (Optional) .....	110
Output Printing Frequency (Optional).....	112
Items in Output Print File (Optional) .....	114
SR2 Output Frequency (Optional) .....	120
Items in Simulation Results File (Optional).....	123

Grid Printout Orientation (Optional) .....	140
Matrix Solver Printout (Optional).....	142
Trap Control-C Interrupt (Optional) .....	143
Grid Array Data Echo Control (Optional) .....	144
Convert Velocities to Shear Rates (Optional).....	145
<b>Reservoir Description</b>	<b>147</b>
Summary of Reservoir Description Data .....	147
Grid Type.....	150
Convert Cartesian Grid to Corner Point (Optional) .....	154
Nine-Point Spatial Discretization (Optional).....	155
Block Dimensions for I Direction (Required) .....	157
Block Dimensions for J Direction (Required) .....	159
Block Dimensions for K Direction (Required) .....	161
Depth (Conditional) .....	163
Depth to the Tops of Grid Blocks (Conditional).....	165
Depths to Centre of Pay (Conditional).....	168
Depths to Top of Block (Conditional) .....	170
Grid Tilt Angles (Conditional).....	172
Corner Point Depths for Corner Point Grids (Conditional) .....	175
Lateral Corner Point Locations for Corner Point Grids (Conditional) .....	177
Line-Based Corner Point Locations for Corner Point Grids (Conditional) .....	179
Complete Corner Point Locations for Corner Point Grids (Conditional) .....	181
Corner Point Tolerance (Optional) .....	184
Local Refined Grid (Conditional).....	185
Block Geometry Modifiers (Optional).....	194
Null Block Indicator (Optional).....	202
Dual Porosity (Optional).....	203
Dual Permeability (Optional).....	204
Dual Porosity Subdomain Method (Optional) .....	205
SUBDOMAIN-DK Transmissibility Multiplier (Optional) .....	207
Dual Porosity MINC Method (Optional) .....	209
Shape Factor Calculation (Conditional).....	211
Fracture Spacing (Conditional).....	213
Fracture Definition (Conditional) .....	215
Discretized Wellbore (Conditional) .....	217
Porosity (Required).....	228
Permeabilities (Required) .....	230
Bulk Volume Modifiers (Optional) .....	232
Netpay (Optional) .....	234
Netgross (Optional).....	236
Transmissibility Multipliers (Optional) .....	237
Transmissibility Multipliers for Lower Indexed Block Faces (Optional).....	240
Transmissibility Multiplier for Matrix-Fracture Flow (Optional) .....	242

Pinch Out Array (Optional).....	243
Pinchout Tolerance (Optional) .....	245
Faults (Optional).....	247
Fault Array (Optional).....	249
Special Connections (Optional).....	251
Fault Transmissibilities (Optional).....	253
Aquifer Model .....	255
Pressure Influence Function (Conditional).....	263
Pore Volume Cut-Off Threshold (Optional) .....	264
Sectors (Optional).....	265
Sector Array (Optional).....	267
Sector Names and Locations (Optional).....	268

## **Other Reservoir Properties** 269

Summary of Other Reservoir Properties .....	269
Indicate End of Grid Definition (Required) .....	272
Rock Type .....	273
Formation Compressibility (Optional) .....	274
Reservoir Pore Volume Dilatation-Recompaction (Optional) .....	279
Reservoir Pore Volume Compaction Rebounding (Optional).....	284
Compaction-Rebounding With Variable cp and cT (Optional).....	287
Variable Permeability (Optional) .....	291
Rock Heat Capacity (Optional) .....	295
Thermal Conductivities (Optional).....	298
SUBDOMAIN-DK Heat Conduction Multiplier (Optional).....	304
Overburden Heat Loss (Optional) .....	306
Diagonal Transmissibility Multipliers (Optional) .....	308
Electrical Heating Sets (Optional).....	310
Electrical Heating Properties (Optional) .....	313
Water Phase Electrical Conductivity (Optional) .....	316

## **Component Properties** 319

Component Types and Names (Required) .....	319
K Value Correlations .....	323
K Value Tables .....	325
Molecular Weight (Required).....	329
Critical Properties (Required) .....	330
Reference Conditions .....	331
Fluid Enthalpies .....	335
Solid Phase Properties (Required).....	341
Liquid Phase Designation.....	344
Liquid Densities (Required) .....	346
Liquid Density Nonlinear Mixing .....	350

Gas Phase Density (Optional).....	353
Viscosity Type (Optional) .....	354
Gas Phase Viscosities .....	355
Liquid Viscosities (Required).....	359
Liquid Viscosity Nonlinear Mixing.....	365
Shear Effects Power Law.....	367
Shear Effects Table .....	371
Nonequilibrium Blockage.....	372
Mandatory Chemical Reaction Data.....	374
Thermal Chemical Reaction Data.....	377
Non-thermal Chemical Reaction Data .....	381
Generalized Reactions .....	385
Partial Equilibrium Reactions .....	387
Ice Modelling (Optional) .....	390

## **Rock-Fluid Data** 393

Summary of Rock-Fluid Data.....	393
Multiple Sets of Rock-Fluid Data.....	396
Interpolation of Relative Permeability and Capillary Pressure.....	398
Critical and Connate Saturations, Scale-Up Factors, and Normalization .....	401
Three-Phase Models .....	407
Wettability Options.....	409
Rock-Fluid Property Identifier (Required) .....	411
Rock Type Number for Rock-Fluid Data .....	412
CounterCurrent Rock Type Data .....	415
Interpolation Component .....	417
Interfacial Tension .....	418
Basic Foam Interpolation Parameters .....	420
Interpolation Set Number and Parameters .....	423
Water-Oil Relative Permeability Table .....	427
Liquid-Gas Relative Permeability Table .....	430
Hysteresis Parameters (Optional) .....	433
Relative Permeability Endpoints .....	454
Relative Permeability Temperature Dependence.....	457
Rock-Fluid Scaling for Each Block .....	458
Capillary Pressure Third-Phase Correction (Optional).....	461
Subdomain Reinfiltration (Optional) .....	469
Effective Molecular Diffusion Coefficients.....	471
Temperature and Viscosity Dependence of Diffusion.....	474
Mechanical Dispersivity .....	476
Total Dispersion Coefficients .....	478
Adsorbing Component Functions .....	480
Rock-Dependent Adsorption Data .....	483

<b>Initial Conditions</b>	<b>487</b>
Initial Conditions Identifier (Required).....	487
Initialization Regions (Optional).....	488
Vertical Equilibrium (Optional) .....	489
Initial Reservoir Pressure and Temperature .....	495
Initial Saturations .....	496
Initial Phase Mole Fractions.....	499
Initial Solid Concentration .....	504
Datum Depth Specification (Optional).....	505
Initial Conditions from Restart.....	507
Initial Conditions from IMEX Run .....	511
<b>Numerical Methods Control</b>	<b>515</b>
Summary of Numerical Methods Control .....	515
Numerical Methods Control Identifier (Optional) .....	517
Maximum Timestep Number (Optional).....	518
Maximum, Minimum Timestep Size (Optional).....	519
Model Formulation (Optional) .....	520
Numerical Set.....	521
Normal Variation in Variables per Timestep (Optional).....	523
Convergence Tolerances (Optional).....	525
Maximum Newtonian Cycles (Optional) .....	531
Under-Relaxation Option (Optional).....	532
Upstream Calculation Option (Optional) .....	533
Discretized Well - Reservoir Upstream Calculation Option (Optional).....	534
Small Rates Option (Optional) .....	535
Convergence Precision for Linear Solver (Optional).....	536
Orthogonalization (Optional) .....	537
Solver Equation Ordering (Optional) .....	538
Solver Factorization Degree (Optional).....	539
Pivot Stabilization (Optional).....	540
Maximum Iterations (Optional).....	541
Adaptive Implicit Flag (Optional).....	542
Pressure and Temperature Limits (Optional) .....	544
Maximum Number of Phase Switches per Timestep (Optional).....	546
Well Pre-Elimination Control (Optional) .....	547
Maximum Cuts Allowed (Optional).....	548
Number of Parallel Processing Threads (Optional).....	549
AIMSOL/PARASOL Switch (Optional).....	550
Number of PARASOL Classes for GMRES (Optional) .....	551
Red-Black Ordering Check for Parasol (Optional) .....	552
Factorization Degree within PARASOL Classes (Optional) .....	553

Factorization Degree between PARASOL Classes (Optional).....	554
PARASOL Class Partitioning Pattern (Optional).....	555
Parallel Jacobian-Building Planes per Domain (Optional) .....	560
Parallel Jacobian-Building Domain Numbers (Optional) .....	561

## **Geomechanics** **563**

Summary of Geomechanical Model .....	563
Geomechanical Model Identifier (Optional).....	572
3D Finite Element .....	573
Independent Geomechanics Grid.....	574
Independent Geomechanics Graphics.....	581
Plane Strain Option.....	584
Deformation Rock Type .....	585
Plastic Model Formation Properties.....	586
Temperature-dependent Properties .....	588
Porosity-Dependent Properties .....	590
Solid Component Properties .....	592
Yield Criterion .....	596
Cap Model.....	598
Cap Model 1.....	599
Nonlinear Constitutive Model.....	603
Nonlinear Elastic Constitutive Model 1.....	604
Nonlinear Elastic Constitutive Model 2.....	606
Creep Model .....	613
Creep Model 1, 2 .....	615
Pseudo Dilation Model .....	619
Generalized Plasticity Model .....	622
Single Surface Failure Model .....	628
Modified Cam Clay Model .....	634
Thermal Expansion Coefficient .....	639
Matrix Permeability Option .....	640
Barton-Bandis Fracture Permeability .....	643
Fracture Direction .....	647
Dilation Relative Permeabilities .....	648
Other Dilation Properties .....	650
Well Radius.....	652
Stiffness Matrix Calculation Option .....	653
Deformation Solution Control .....	654
Geomechanics AIMSOL Control .....	657
Dimension Over-Rides (Optional) .....	660
Initial Stress Distribution (2D).....	661
Initial Stress Distribution (3D).....	664
Geomechanical Reference Block .....	669

Prescribed Boundary Conditions (2D) .....	670
Prescribed Boundary Conditions (3D) .....	676
Point Loads (2D) .....	680
Point Loads (3D) .....	684
Distributed Edge Loads (2D) .....	686
Distributed Surface Loads (3D) .....	690
Gravity Loads (2D) .....	695
Gravity Loads (3D) .....	697
Fixed Null Block .....	699
Fixed Cap Rock .....	702
Geomechanics Domain.....	703
Pressure Boundary Domain .....	705
Coupling Options .....	708
Geomechanical Coupling Factor .....	711
Pressure Tolerance Multiplier .....	712
Coupling Update Times.....	713
Porosity Calibration.....	715
Iterative Coupling to Fluid Flow .....	717
Boundary Stress Unloading.....	719

## **Well and Recurrent Data** 721

Summary of Well and Recurrent Data .....	721
Well and Recurrent Data Identifier (Required) .....	730
Simulation Reference Times .....	731
Simulation Pause .....	734
Simulation Reference Times .....	736
Group Identification (Optional).....	737
Well Identification (Required) .....	741
Define Reporting Group (Optional) .....	744
Well Head Method (Optional).....	746
Perforations in Inactive Blocks (Optional).....	749
Well Backflow Model (Optional).....	751
Set Frequency of Initialization of Bottom-Hole Pressure (Optional).....	753
Shut in Wells above Formation (Optional) .....	755
Well Type Definition (Required) .....	757
Shut and Reopen a List of Wells (Optional) .....	759
Wellbore Pressure Drop and Heatloss (Optional) .....	761
Injection Stream Attributes .....	769
Composition of Injected Phases .....	771
Well Operating Constraints (Required) .....	774
Maximum Number of Continue-Repeats (Optional).....	781
Well Monitoring Constraints (Optional) .....	782
Well Element Geometry (Conditional) .....	787

Location of Well Completions (Conditional) .....	789
Location of Vertical Well Completions (Conditional) .....	799
Geometric Data for Deviated Well Completions (Conditional) .....	800
Simplified Geometric Data for Deviated Well Completions (Conditional).....	803
Limited Entry Perforations (Optional) .....	806
Pressure Gradients for Calculation of Pressure Differences between Completions (Conditional).....	809
User Specified Reference Depth for Well BHP (Optional) .....	812
User Specified Pressure Gradient For Reference Depth for Well BHP (Optional) .....	814
Alter Primary Well Operating Constraint Value (Optional).....	816
Alter Well Constraint Value (Optional).....	818
Resetting Well Operating Constraint after Value Change (Optional) .....	820
Cyclic Steam Stimulation Groups.....	822
Automatic Switching between Steam Cycles .....	823
Gas Lift Option .....	826
Other Well Attributes.....	828
Group Production Constraints (Optional) .....	830
Group Injection Constraints (Optional) .....	835
Recycled Water Component Mask for Group Water Recycling (Optional) .....	843
Make-up Water Composition for Group Water Recycling (Optional) .....	844
Water Make-up Target for Group Water Recycling (Optional).....	845
Monitored Group Constraints (Optional).....	846
Defining Group Production or Injection as Going through a Manifold (Optional).....	850
Pressure-Constraint Translation for Manifolds (Optional) .....	852
Specification of Hydraulics Tables for Calculating Pressure Difference between Manifold and Surface (Optional).....	854
Manifold Depth for Calculating Pressure Difference between Manifold and Surface (Optional) .....	856
Group Artificial Lift Quantity Value (Optional).....	858
Well Artificial Lift Quantity Value (Optional) .....	860
Priority List for Automatic Drilling of Wells (Optional).....	862
Group Apportionment Options (Optional).....	864
Apportionment Method for Meeting Group Targets (Optional) .....	866
Priority Formulae for Apportionment (Conditional) .....	870
Guide Rates for Groups or Wells.....	874
Flag for Accompanying Groups or Wells Not Under Group Control (Optional) .....	877
Well/Group On-time Fraction (Optional) .....	879
Hydraulic Pressure Table (Conditional) .....	883
Allow a Set of Keywords to be Processed When a Specified Condition (Trigger) is Satisfied (Optional) .....	890
Alter Well Constraint Value via a Multiplier (Optional) .....	913
Alter Well Constraint Value via an Increment (Optional) .....	917

Group Production Constraints Multiplier (Optional) .....	921
Group Injection Constraints Multipliers (Optional) .....	925
Allow a Cycling Group to be Defined (Optional) .....	929
Allow Cycling Group Data to be Revised (Optional) .....	940
Allow Cycling Group Data to be Revised (Optional) .....	942
Select Cycling Group Cycle Part to Start and End Cycling Group Control (Optional) .....	944
Constant and Convective Heat Transfer Model.....	946
Adiabatic Heat Transfer Control .....	951
Slaved Heater Control .....	953
Heater Well .....	955
Wellbore Block Transmissibility Multipliers (Optional) .....	960
Pressure Dependent Transmissibility Multipliers .....	961
Automatic Rock-Fluid Switching.....	963
Reset Adaptive Implicit.....	965
Dynamic Grid Amalgamation Control (Optional) .....	966
Discretized Wellbore in Recurrent Data (Conditional) .....	975
Electrical Heating Boundaries (Conditional).....	977
<b>Tables</b>	<b>981</b>
Table 1: Ordering of Components.....	981
Table 2: K-Value Coefficients for Selected Components .....	982
Table 3: Critical Properties for Selected Components .....	983
Table 4: Liquid Viscosity Coefficients for Selected Components .....	984
Table 5: Gas Heat Capacity Coefficients for Selected Components .....	985
Table 6: Vaporization Enthalpy for Selected Components .....	987
Table 7: Selected Unit Conversions .....	988
<b>Appendix A: Well Model Details</b>	<b>989</b>
Overview .....	989
A.1 Radial Inflow Well Model.....	990
A.2 Well Indices.....	992
A.3 Anisotropic Permeability.....	994
A.4 Backflow.....	995
A.5 Surface Flash .....	997
A.6 Calculation of Geometrical Factor CC .....	998
A.7 Notes on Discretized Wellbore Model Usage .....	1000
<b>Appendix B: Data Sets</b>	<b>1005</b>
B.1 Summary of Test Bed Data Sets .....	1005
B.2 Template Sample Data Sets .....	1007

## **Appendix C: Advanced Processes** 1015

Overview.....	1015
C.1 Hot Water Flooding Process .....	1016
C.2 Steam Flooding Process .....	1017
C.3 Steam Cycling Process .....	1018
C.4 Fire Flood Process.....	1019
C.5 Additives Overview.....	1020
C.6 Gas, Water and Oil Phase Tracers.....	1021
C.7 Gas Additives .....	1022
C.8 Water-Rock Chemical Interactions .....	1023
C.9 Polymers and Gels.....	1024
C.10 Surfactant and Caustic.....	1025
C.11 Fines and Emulsions.....	1026
C.12 Oil Additives and Partitioning Inversion.....	1027
C.13 Foam.....	1028

## **Appendix D: Fluid and Rock Properties** 1035

Overview.....	1035
D.1 Components and Phases.....	1036
D.2 Component Design Concepts.....	1040
D.3 Fluid Phase Equilibrium.....	1042
D.4 Fluid Densities .....	1047
D.5 Viscosity .....	1050
D.6 Rock Fluid Properties .....	1057
D.7 Component Adsorption and Blockage .....	1062
D.8 A Simple Foam Model.....	1065
D.9 Phase Enthalpies .....	1068
D.10 Thermal Conductivity .....	1069
D.11 Overburden Heat Loss .....	1071
D.12 Thermal Aquifer.....	1073
D.13 Chemical Reactions.....	1076
D.14 Basic Concepts for Nonequilibrium Mass Transfer.....	1080
D.15 Stable Emulsion Flow and In Situ Generation Concepts .....	1081
D.16 A Lamella Density Model of Foam .....	1084
D.17 Oil Banking Theory .....	1085
D.18 Converting Black-Oil PVT to STARS .....	1088
D.19 Other Aquifer Models .....	1094
D.20 Velocity-Dependent Viscosity .....	1107

<b>Appendix E: Grid Design</b>	<b>1111</b>
Overview .....	1111
E.1 Nonuniform Formation Properties .....	1112
E.2 Resolution of Process Phenomena .....	1113
E.3 Variable Depth and Thickness .....	1115
E.4 Grid Orientation .....	1116
E.5 Symmetry Elements .....	1118
E.6 Local Grid Refinement .....	1119
E.7 Hybrid Grid.....	1120
E.8 Naturally Fractured Reservoirs .....	1122
<b>Appendix F: Equations</b>	<b>1137</b>
Overview .....	1137
F.1 Overview .....	1138
F.2 Conservation Equations .....	1139
F.3 Phase Equilibrium Relationships .....	1144
F.4 Well Equations .....	1145
F.5 Summary of Conservation Equations.....	1147
F.6 Solution of Nonlinear Equations – Newton’s Method .....	1148
F.7 Solution of Linear Equations – General Sparse Solver.....	1151
F.8 Treatment of Solid Components .....	1152
F.9 Adaptive-Implicit Method.....	1155
F.10 Use of Constraint Equations in the Sxy Formulation.....	1157
<b>Appendix G: Electrical Heating</b>	<b>1161</b>
Overview .....	1161
G.1 Brief Description of Theory .....	1162
G.2 Mathematical Model Used by STARS .....	1164
G.3 Reports and Plots .....	1167
G.4 Templates .....	1169
G.5 Input Data .....	1171
G.6 References .....	1172
<b>Keyword Index</b>	<b>1173</b>

# Introduction

---

## Overview

This introduction is divided into the following sections:

- Important Changes Between STARS 2011.10 and 2010.10
- Data Incompatibilities with Previous Versions of STARS
- New Keywords Added to STARS 2011.10
- Enhancements to Existing Keywords
- New and Changed Template Data Sets
- Introduction to STARS

---

## Important Changes between STARS 2011.10 and 2010.10

### Flexible Wellbore

These options have been added to the Flexible Wellbore feature:

1. Add a *flow control device*, based on either orifice flow or fluid friction, to flow between either tubing/annulus or annulus/reservoir.
2. Add a *packer* to prevent fluid flow axially along a wellbore stream.
3. Add an *instrumentation tubing* string to record temperature without having to add a fluid flow well.
4. *Switch mode* (injection/production) of a Flexible Wellbore stream while keeping the current conditions, that is, without re-initializing the stream.
5. *Change grid* in recurrent data, e.g., with \*DYNAGRID.

### Geomechanics

The Cam Clay material model is now available with Geomechanics.

### SR2 File

These changes were made to the SR2 file system to improve efficiency and capacity. Results and STARS can read both the new and previous SR2 file formats.

1. Increase binary record index from 32 bits to 64 bits, effectively removing the limitation of number of blocks with \*GRID \*CORNERS (previously 22 million).
2. Reorganize well performance records to reduce fragmentation, reducing the time required for Results to load SR2 files with a large number of wells and/or completion layers.
3. Eliminate long lists of consecutive integers in the IRF file.
4. Eliminate redundant corner point coordinates for (1) most grids using \*GRID \*CORNER and (2) all grids generated by \*WSRF \*GRIDDEFORM. This saves space in the binary MRF file and reduces significantly the resources needed by Results to load the file.

---

## Data Incompatibilities with Previous Versions of STARS

*The following **mandatory** data changes must be done to an existing STARS data set in order for it to work correctly with version 2011.*

1. The documented default of \*PINCHOUT-TOL is now honoured. Data sets without \*PINCHOUT-TOL but with cell thicknesses less than the documented default (e.g., micro models) may have run in previous versions. Now, such a data set requires that you override the \*PINCHOUT-TOL default (e.g., with value 0). This was in the 2010 release but not the 2010 manual.
2. The behavior of \*VERTICAL \*DEPTH\_AVE has changed for cases where \*REFDEPTH is in an overburden region containing no fluid-bearing cells. You may need to change your data in order to reproduce the previous behavior. See **Insufficient Pressure and Overburden and Zero-Porosity Cells** in the EXPLANATION section of keyword \*VERTICAL.
3. The behavior of \*PVCUTOFF has changed for thermal runs. A cell's type is changed to "zero porosity" if its pore volume falls below \*PVCUTOFF and it has a small non-zero porosity or net-to-gross ratio. Previously that cell's type was changed to null, which generally is not appropriate in a thermal context. Manual application of keyword \*NULL can reproduce the previous resulting grid, but this is not recommended and is not practical for large grids. See template STTST21.
4. A well completion in a hybrid grid (\*REFINE \*HYBRID) cannot be deviated. If such a completion appears in a deviation list of \*LAYERXYZ or \*LAYERIJK, it should be marked \*UNDEVIATED. Such a completion that appears with deviation data will be set internally to undeviated. Previously the deviation data was used, incorrectly.

---

## New Keywords Added to STARS 2011.10

1. New \*OUTSRF \*GRID subkeyword \*SCLASS allows you to view the Parasol class number. Also, \*OUTPRN \*ITER \*TSS triggers a Parasol classes report.
2. New \*OUTSRF \*GRID subkeywords \*STRESINVF, \*STRESINVS, \*VOIDRATIO, \*TVERDPLGEO, \*TSUBSIDGEO, \*THEXPCOEF, \*THCOMPR, \*BULKVOL and \*GEORTYPE are available with \*GEOMECH.
3. New special history \*PRODSTEAMR allows you to plot well steam production as cold-water equivalent liquid. See template STTST07. This was in the 2010 release but not the 2010 manual.
4. New keyword \*TORTIKE\_VG allows you to use the steam viscosity correlation of Tortike et al. See template STFLU039.
5. New \*BLOCKAGE subkeyword \*WG allows you to specify blockage in only the water and gas phases.
6. New keyword \*INIT\_FROM\_IMEX allows you to get initial conditions from an IMEX SR2 file set. See template STSMO058.
7. New keywords \*COHESHARD and \*FRANGSOFT are added to the Generalized Plasticity Model \*GENPLAST. See template STGEO057.
8. New keyword \*MCCMODEL lets you use the Cam Clay geomechanical material model. See template STGEO058.
9. New \*PRINTGEO flag “13” reports on the mapping between the host fluid-flow grid and the separate geomechanics grid (\*GEOGRID).
10. New subkeywords \*STO and \*STF are available with \*GCONI. See templates STWWM054 and STWWM055. This was in the 2010 release but not the 2010 manual.
11. New \*GCONI subkeyword \*VREFP allows you to specify a reference volume fraction injection target, for \*GAS or \*WATER. See template STWWM067.
12. New \*DYNAGRID subkeywords \*SATWAT, \*SATOIL and \*SATGAS allow you to specify dynamic gridding saturation-difference parameters for each phase separately.
13. New Flexible Wellbore keyword \*REPLACE allows you to change the mode (injection or production) of a stream without losing the current conditions. See template STWWM070.
14. New Flexible Wellbore subkeyword \*PACKER allows you to specify a fluid flow barrier. See templates STWWM068 and STWWM069.
15. New Flexible Wellbore subkeyword \*TUBING\_INST allows you to specify a tubing that reports temperature but has no fluid flow. See templates STWWM071 and STWWM072.
16. New Flexible Wellbore subkeyword \*MAX\_NUSSELT lets you control the limit of Nusselt number in the radial heat transfer calculation.

17. Flexible Wellbore is now available with options that change the grid during the run, e.g., \*DYNAGRID.
18. New Flexible Wellbore keyword \*FCD-ORIF allows you to specify a flow control device based on orifice flow between tubing and annulus or between annulus and reservoir. See templates STWWM068 and STWWM069.
19. New Flexible Wellbore keyword \*FCD-FRIC allows you to specify a flow control device based on flow friction between tubing and annulus or between annulus and reservoir.

---

## Enhancements to Existing Keywords

1. The manual entries for \*DI, \*DJ, \*DK and \*PVCUTOFF were improved.
2. The plotting limit of 22 million cells available with \*GRID \*CORNER has been removed. This limit was due to the 32-bit indices previously used in the SR2 files.
3. The calculation of interblock flow due to diffusion and dispersion uses directly the most up-to-date block values of those properties. Previously, interblock diffusion and dispersion transmissibilities were calculated at the start of the run and used unchanged throughout the run. This was in the 2010 release but not the 2010 manual.
4. Rock-fluid option \*LININTERP is no longer constrained by static dimensioning.
5. \*SOLVER \*PARASOL operation has been verified for a combustion case with 7 million cells and 7 flow equations, run in parallel on up to 32 threads. This case required about 105 Gb of RAM.
6. The \*PPATTERN algorithms ensure that all the members of a discretized wellbore fall within the same Parasol class, thereby enhancing stability of multi-threaded runs. See **Discretized Wellbore** and **Reporting Class List and Distribution** in the EXPLANATION of \*PPATTERN. This was in the 2010 release but not the 2010 manual.
7. Hydraulic Pressure Table \*PTUBE1 now works in a thermal run. Previously, \*PTUBE1 was confined to \*ISOTHERMAL runs.
8. The manual entry for \*OPERATE now reports that constraint types \*BHO, \*BHG and \*BHW may be used for injection.
9. Heater well \*HTWELL now works with \*DYNAGRID.
10. New \*MONITOR action \*AUTOLAYER allows a producing well's shut-in layer to be checked periodically for conditions that allow for reopening. See template STWWM047. This was in the 2010 release but not the 2010 manual.
11. A Flexible Wellbore continues to calculate heat transfer and loss when all its streams are closed.
12. RESULTS Graph reports Flexible Wellbore stream type (e.g., tubing, annulus).

---

## New and Changed Template Data Sets

These files can be found in CMG release area .../cmg/stars/2011.vv/tpl where vv is the particular version number.

### Fluid Types (directory /flu)

stflu039.dat Test/Illustrate \*TORTIKE\_VG Steam Viscosity

### Geomechanics (directory /geo)

stgeo056.dat Infinite Mohr-Coulomb Medium, Dilation Angle, Over/Underburden

stgeo057.dat Generalized Plastic Model \*GENPLAST with \*COHESHARD and  
\*FRANGSOFT

stgeo058.dat Skempton and Mandel-Cryer Effects with Modified Cam Clay Model

### Grid Options (directory /gro)

stgro052.dat Diagonal 25x13 Grid for 1/8 9-Spot Pattern

stgro053.dat Parallel 18x18 Grid for 1/8 9-Spot Pattern

### Simulator Options (directory /smo)

stsмо058.dat Illustrate/Verify \*INIT\_FROM\_IMEX

### Wells and Well Management (directory /wwm)

stwwm047.dat Miscible Flood with Nested Group Control and Autodrill

stwwm067.dat Miscible Flood with Reference Volume Replacement

stwwm068.dat Flexible Wellbore with \*PACKER and \*FCD-ORIF for Annulus

stwwm069.dat Flexible Wellbore with \*PACKER and \*FCD-ORIF for Tubing

stwwm070.dat Flexible Wellbore with \*REPLACE

stwwm071.dat SAGD with Flexible Wellbore and Instrumentation Tubing

stwwm072.dat Steam Cycling with Flexible Wellbore and Instrumentation Tubing

---

## **Introduction to STARS**

### **INTRODUCTION**

STARS is a three-phase multi-component thermal and steam additive simulator. 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.

Some of the novel features of STARS are:

### **DISPERSED COMPONENT INCLUDING FOAM**

The concept of dispersed components - stabilized dispersions (droplets, bubbles, and lamellae) of one phase in another, which can be treated as components in the carrying phase at the scale of reservoir simulation - provides a unifying point of view in the modelling of polymers, gels, fines, emulsions, and foam. This concept can be coupled with the flexible component property input package capabilities (including adsorption, blockage, nonlinear viscosity, dispersion, and nonequilibrium mass transfer) to allow the user to design appropriate simulation models of complex phenomena via input data choices alone.

In particular, two general approaches to the modelling of foam flow are available. The first, a mechanistic model, allows direct simulation of foam creation, propagation, and coalescence effects such as can be observed in detailed laboratory core experiments. The second approach is more empirical and appears more appropriate for foam scoping studies and field pilot history matching. The first approach can be used to justify aspects of the empirical model.

### **NATURALLY FRACTURED RESERVOIRS**

The flow in naturally fractured reservoirs can be simulated by using four different models - dual porosity (DP), dual permeability (DK), multiple interacting continua (MINC), or vertical refinement (VR) - depending on the process or mechanisms to be studied.

The basic approach idealizes the fractured reservoir as consisting of two parts: fracture and matrix. The fractures, having small storativities, are the primary conduits of fluid flow, whereas the rock matrices have low fluid conductivities but larger storativities. The various simulation models differ in the details of matrix-matrix and matrix-fracture flow descriptions and are discussed in greater detail in the STARS Technical Manual.

### **ADAPTIVE IMPLICIT FORMULATION**

STARS can be run in fully implicit and adaptive implicit modes. In many cases only a small number of grid blocks need to be solved fully implicitly, since 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. STARS can select these blocks dynamically, based on specified thresholds or on matrix switching criteria.

## **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**

STARS uses a state-of-the-art solution package AIMSOL based on incomplete Gaussian Elimination as a preconditioning step to GMRES acceleration. AIMSOL has been developed especially for adaptive implicit Jacobian matrices. For more information see the AIMSOL Technical Manual.

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

## **LOCAL CARTESIAN**

Two facilities for local grid refinement are available. These options can be used to study near-well effects in field scale simulation. Static fractures can also be efficiently modelled with this technique. With either method, the user specifies a region of the reservoir that is to be subdivided. All interblock connections and transmissibilities are calculated automatically. All extra terms are handled correctly by the matrix solution routine.

## **FLEXIBLE GRID SYSTEM**

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

## **AQUIFER MODELS**

Aquifers are modelled by either adding boundary cells that contain only water or by the use of a semi-analytical aquifer model.

The former 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 latter method is 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. When reservoir fluid invades the aquifer a combination of both methods is required.

## **INPUT/OUTPUT UNITS**

SI, field, or laboratory units can be specified.

## **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, including grid design.

## **DISCRETIZED WELLBORE**

The advent and growing acceptance of horizontal well technology has raised many new questions that need to be addressed with reservoir simulation models. In particular, the impact of long wellbore transients, viscous pressure drop and multiphase flow patterns in creating non-uniform injectivities and productivities along the wellbore are of concern. STARS provides an efficient and consistent method for handling these questions by discretizing wellbore flow and solving the resulting coupled wellbore/reservoir flow problem simultaneously. Appropriate multiphase flow correlations are used to adjust wellbore flow patterns in an explicit fashion at the end of each timestep. The circulating option makes available a concentric tubing/annulus pair of flowing streams that are tightly coupled by heat transfer. The discretized wellbore also models phase segregation between wellbore sections which can be very important in non-horizontal configurations. Also, heat conduction in a discretized wellbore continues even though the fluid flow has stopped, which can be significant in start-up and shut-in phases of a project.

## **GEOMECHANICAL MODEL**

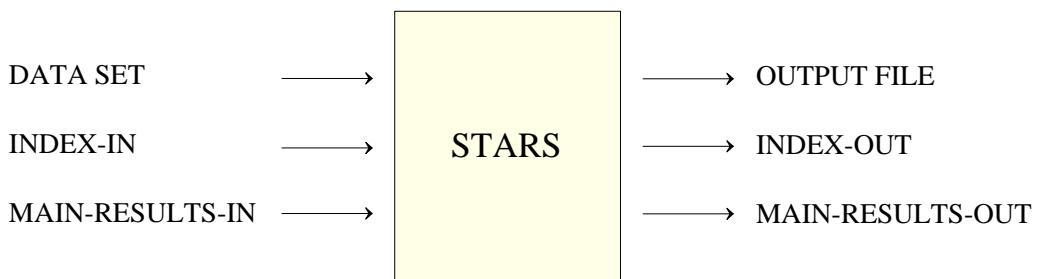
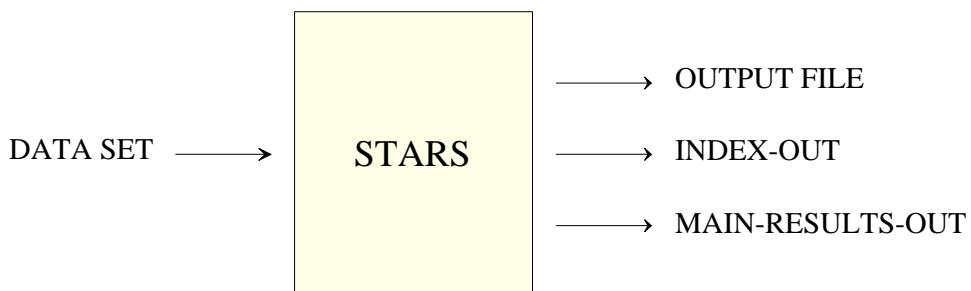
Several production practices depend critically on the fact that the producing formation responds dynamically to changes in applied stresses. These include plastic deformation, shear dilatancy, and compaction drive in cyclic injection/production strategies, injection induced fracturing, as well as near-well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

# Tutorial

## Introduction

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

STARS uses the data set that you create initially and then creates three other files. Each STARS run creates a text output file, an SR2 index file (IRF), and a SR2 main file (MRF):



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 nine different data groups in the keyword input system.
- b) The groups must follow a certain input order:
  - Input/Output Control
  - Reservoir Description
  - Other Reservoir Properties
  - Component Properties
  - Rock-fluid Data
  - Initial Conditions
  - Numerical Methods Control
  - Geomechanical Model
  - Well and Recurrent Data
- c) The keywords belonging to each group cannot appear in other groups, unless it is specifically indicated. Usually, this happens with 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 Read Keyword Syntax

Each keyword has a syntax, that is, the exact characters, options and ordering that the keyword processor will accept. Once you know the few syntax rules, you will be able to interpret the form of each keyword used in this manual.

Items contained in round brackets are optional, that is, you enter the item or not. Do not put these round brackets in your data. For example, the syntax for the \*RANGECHECK keyword is

\*RANGECHECK ( \*ON | \*OFF )

which means that the following forms are acceptable:

\*RANGECHECK

\*RANGECHECK \*ON

\*RANGECHECK \*OFF

The vertical bar means 'or', and separates items in a list of choices.

Braces {} denote any number of a list of items. For example, {well\_name} denotes an arbitrary list of quoted well names.

A list of items in braces on a line below a keyword denotes a table of arbitrary length. For example, the water-oil relative permeability table:

\*SWT

{ S<sub>w</sub> k<sub>rw</sub> k<sub>row</sub> (P<sub>cow</sub>) }

means that any number of rows of S<sub>w</sub>, k<sub>rw</sub> and k<sub>row</sub> (and optionally P<sub>cow</sub>) can be entered, subject to sufficient dimensioning.

A property that is to be assigned to grid blocks will be denoted with ARRAY. For porosity,

ARRAY: \*POR

This indicates that the \*POR keyword must be used with a grid-array-reading option. An alternative syntax that you may see for a grid array is

\*POR {grid}

Some keywords require one number for each component. These will be shown as

\*KEYWORD {ncomp}

\*KEYWORD {numy}

\*KEYWORD {numx}

where ncomp, numy and numx are specified in the \*MODEL keyword which defines the component list.

A colon denotes a range. It is used most frequently for block I-J-K addresses. In the I direction,

i1(:i2)

denotes a single number i1 or the range i1:i2. Of course, the range must fit the context; here, i1 and i2 must lie in [1,ni], and i1 must not be greater than i2 (ni = blocks in I direction).

---

## 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 quotes.

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

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

Example:

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

---

## How to Do a Restart

### WHAT IS A RESTART FILE?

A restart file contains information that allows the simulation to continue from another run.

### 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 a restart file 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 required only if you plan on restarting from your current run.

To do a restart run:

- a) In the first run use keyword \*WRST in the Input/Output Control section and/or in the Recurrent Data section. \*WRST indicates the frequency of writing to the restart record. Run this first data file. The resulting files will include an IRF file, and MRF file and possibly an RRF file.
- b) Copy the first data file (or just the main data file if you are using the \*INCLUDE option) to another filename, preferable with a similar name (e.g., case1a.dat, case1b.dat). Do not change any of the original non-recurrent data (with the exceptions noted below). Add keyword \*RESTART to the Input/Output Control section of your data set. If you do not wish to be prompted interactively for the input restart IRF filename, specify it with \*FILENAMES \*INDEX-IN.
- c) Make desired changes to the recurrent data, but only for times after the restarting time. Increase the maximum number of timesteps, if necessary, or leave out \*MAXSTEPS altogether.
- d) Run the second run, supplying the first run's IRF filename if you are prompted for it.

Example:

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

## WHAT CAN BE CHANGED AT A RESTART

It is safest to change at a restart only those data found in the recurrent data section. The following data, which affect only inter-block flow or source/sink terms, may be changed at a restart with caution.

- a) Chemical reactions and partial equilibrium reactions.
- b) Rock-fluid data, but not adsorption. A preferred method is defining multiple rock types with \*RPT and assigning them with \*KRTYPE in recurrent data.
- c) Viscosities.
- d) Absolute permeability, but only if it does not affect the porosity (e.g., dilation).

Changing component properties or reservoir characteristics manually at a restart is not recommended, because it over-rides the consistency built into the simulator and can produce results that cannot be reproduced later.

At no time should data that affects material in place (e.g., densities, K values, block sizes, and porosity) be changed at a restart. Doing so results in material balance errors that cannot be resolved during the first timestep of the restart run. Over-riding material balance checks to “get through” this problem is not recommended or supported.

Special history definitions cannot be changed at a restart.

The \*TFORM option and \*ISOTHERMAL setting may not be changed at a restart. This implies that a thermal run may not be restarted from an isothermal run.

---

## Controlling Contents of the Output Print File

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

- a) \*WPRN and
- b) \*OUTPRN.

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

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

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

Example:

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

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

\*OUTPRN limits what 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 \*ALL. This is the default. To print out a one line summary for each well use \*OUTPRN \*WELL \*BRIEF.

---

## **Controlling Contents of the Simulation Results File**

To control the contents of the Simulation Results Files (SR2), use \*OUTSRF.

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

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

\*OUTSRF limits what well data, grid data, and reservoir 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.

---

## Describing Your Grid System

To describe your grid system, you need:

- a) \*GRID,
- b) \*DI,
- c) \*DJ,
- d) \*DK,

Optional keywords are

- e) \*DEPTH, \*DTOP and \*DIP.

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 3 choices: regular Cartesian, variable depth/variable thickness, and radial-angular cylindrical 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
```

The first describes a regular Cartesian grid that is 10 x 10 x 6. The second describes a variable depth variable thickness grid that is also 10 x 10 x 6. Lastly, the third example describes a radial- angular cylindrical system for a coning study. It is 10 x 1 x 15.

The keywords \*DI, \*DJ, and \*DK are required keywords. 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. Note that your data starts with the bottommost layer when using \*KDIR \*UP.

---

## Specifying Null Blocks

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

- a) \*NULL and
- b) \*VAMOD.

Both must appear in the Reservoir Description section.

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.

Note that \*NULL is the preferred method for specifying null blocks. In STARS a block with zero porosity does not become a null block like it does for an isothermal simulator. In STARS a zero-porosity block stays active in order to handling thermal conduction, even though it has zero pore volume.

---

## Describing Refined Grid

To describe the location of refined grid, use \*REFINE. \*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.

For example, you want to split block (1,1,3) in a 10 x 10 x 3 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 1 1 3 into 2 3 2
```

You are allowed to split up a fundamental block into a maximum of 4 refined blocks in each direction. 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.

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 attributed to the refined grid blocks.

---

## Using Dual Porosity/Dual Permeability

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

- a) \*DUALPOR
- b) \*MINC,
- c) \*SUBDOMAIN,
- d) \*DUALPERM,
- e) \*DIFRAC,
- f) \*DJFRAC, and
- g) \*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.

### DUAL POROSITY/DUAL PERMEABILITY CASE

In the case of a dual porosity/dual permeability model, the input of porosity values requires input 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.

---

## Problems with Small Timesteps or Long Execution Times

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

\*OUTPRN \*ITER \*NEWTON

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

Convergence failure may result due to:

- a) Inner 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 time-step cuts and are caused by maximum changes, in the primary variables, which exceed the specified nominal changes by a factor more than two per 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 timestep cuts occur, then you can try the following remedies:

4. Check the rock and PVT curves for non-linearities. The curves should be smooth.
5. Check that grid and other properties are properly specified.
6. Check the well constraint specification. It is good practice to always specify a maximum bottomhole pressure for each injector and a minimum bottomhole pressure for each producer.
7. Increase the specified number of Newton's cycle using the keyword \*NEWTONCYC if non-convergence 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 non-linearities (4) or reducing the timestep size (1) are better solutions.

8. Set some regions of the reservoir or the entire reservoir to fully implicit. The default switching criterion, \*AIM \*STAB checks for switching from explicit to implicit only if the grid block is a neighbour of an implicit block. Thus if there are regions of the reservoir where there are dramatic changes taking place and these regions are not adjacent to wells, then set the problem regions to implicit.

Examples of such situations include:

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

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.

See also “Improving Numerical Performance” later in this Tutorial section.

---

## Defining Wells

Wells are defined using the following keywords. Be aware that the order of the keywords must be strictly adhered to:

**\*WELL**

(Required)

**\*PRODUCER**

(Required keywords which must follow well completion keywords.)

-or-

**\*INJECTOR**

-or-

**\*SHUTIN**

-or-

**\*OPEN**

**\*INCOMP**

(Required if you are injecting oil or gas phase. Keyword follows **\*INJECTOR**.)

**\*OPERATE**

(At least one operating constraint is required.)

**\*MONITOR**

(Monitoring constraints are optional.)

**\*GEOMETRY**

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

**\*PERF**

(At least one of these three or a combination thereof, is required.)

-or-

**\*PERFV**

-or-

**\*PERFRG**

These keywords must all reside in the Well Data section of your data set.

It is possible to define a well, name it, and specify its group affiliation with a **\*WELL** keyword at one time, specify its completions with **\*PERF** at a later time, and finally define the well's type with **\*PRODUCER** or **\*INJECTOR** at a still later time and have the well become active.

---

## Defining the Well Type

There are four well types. They are:

- a) \*PRODUCER,
- b) \*INJECTOR,
- c) \*SHUTIN, and
- d) \*OPEN.

Each of these keywords must appear in the Well Data section, and \*PRODUCER or \*INJECTOR must be defined before a well can be put into operation.

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

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

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

You may open a shut-in well any time after the well's type has been defined with \*PRODUCER or \*INJECTOR.

Example:

One cycle of steam stimulation. Define both wells, then open and shut as needed.

```
time 0 ** Cycle No. 1 - Injection

    ** INJECTOR: Constant pressure steam injection
    well 1 'Injector 1'
    injector mobweight 1
    operate bhp 1000
    tinjw 450 qual .7
    perf 1 ** i j k wi
        1 1 1 88

    ** PRODUCER: Constant liquid rate type
    well 2 'Producer 1'
    producer 2
    operate stl 1000
    perf 2 ** i j k wi
        1 1 1 88

    shut in 2 ** Shut in producer

time 10 ** Cycle No. 1 - Soak

    shut in 1 ** Shut in injector

time 17 ** Cycle No. 1 - Production

    open 2 ** Turn on producer

time 40 stop
```

---

## Operating and Monitoring Constraints

OPERATE and \*MONITOR indicate the constraints on a given well. At least one operating constraint is required and the monitoring constraints are optional.

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

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

If more than one operating constraint is violated, then the 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),
- 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,
- 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
```

---

## Specifying Well Indices

To input well indices, these keywords are used:

- a) \*GEOMETRY,
- b) \*PERF, or
- c) \*PERFV, or
- d) \*PERFRG.

These keywords must reside in the Well 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, \*PERFV, and \*PERFRG 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 precedes \*PERF, \*PERFV, and \*PERFRG.

\*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  
    1 1 2:4 1.24  
-or-  
  
*WELL 1 '12-09-18-56'  
**           rad   geofac   wfrac   skin  
*GEOMETRY *K   .375   .2488   1.0   0.0  
** The well completion keyword must follow  
** the geometry keyword pertains to well 1.  
*PERF *GEO 1  
** if   jf   kf   ff  
    1   1   2:4   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  
    2:4   1.56
```

If you are using refined grid blocks and wells are located within the vicinity, then \*PERFRG must be used. \*GEO is again required if \*GEOMETRY is used. \*PERFRG 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
*PERFRG 1
** if jf kf ir jr kr wi
  1    1    3    2    2    1:2   1.75
```

---

## Horizontal Wells

Horizontal wells can be simulated in two different ways.

### METHOD 1:

The first method involves modelling the well as a line source (injector) or sink (producer). This method neglects wellbore frictional pressure drop and liquid holdup effects.

When using the source-sink method, you should be aware that if the field you are trying to model has any known backflow problems, this method will give erroneous results. Small amounts of backflow in general are not important. If you notice a change, even a small one, areally, in the permeability, the difference may cause backflow.

The keywords used to define a horizontal source/sink are:

- a) \*WELL
- b) \*INJECTOR or \*PRODUCER
- c) \*OPERATE
- d) \*GEOMETRY
- e) \*PERF \*GEO

\*GEOMETRY and \*PERF result in the output of well productivities. Run this data set and observe the resulting production rates. If these values are not what you wish, enter your own values via \*PERF without \*GEO.

### METHOD 2:

The second method for modelling horizontal wells is to use the discretized wellbore model.

This novel method dynamically handles wellbore hydraulics and can also be used for vertical producers. This method is ideally suited where frictional pressure drop or liquid holdup effects are important. The keyword for invoking this option is \*WELLBORE. Since the method models the wellbore as a second porosity in the well block, the corresponding compressibility, rock type and relative permeability tables must also be assigned.

---

## **Stopping a Simulation Run**

Normally simulation stops after timestepping reaches the last time or date specified in the data file. Use keyword \*STOP to terminate the simulation run at a date/time before the last one.

Example:

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

---

## Guidelines for Setting Up Well Data

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

1. Either \*TIME or \*DATE is required.
2. Define a value for \*DTWELL, the first timestep size used immediately after the well is defined.
3. Identify all new wells using \*WELL.
4. Indicate the well locations, geometries, or the well indices using \*GEOMETRY and any of the well completion keywords (\*PERF, \*PERFV, or \*PERFRG). This may be done at a \*TIME later than that at which the \*WELL statement defined the well.

Each set of well definitions consists of :

- 5a. Define the type of 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.
- 5b. Define the operating or monitoring constraints for that well.

Steps 1 through 5 MUST appear in any data set. Step 4 may follow step 5 but only if the PERF lines are entered at the same \*TIME as the well type information.

6. Use \*SHUTIN only after steps 1 through 5 have been followed. When \*WELL is entered, the well status is initialized as shut in. \*SHUTIN may be entered for a well any time after the \*WELL information is entered.
7. Use \*OPEN to reopen a previously shut-in well. Use \*OPEN only after the well's type has been defined with \*PRODUCER or \*INJECTOR.
8. 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 the following steps:

9. Define new wells and use steps 1,3,4 and 5 before adjusting the parameters of existing wells.
10. You may alter the primary operating constraint of any well with \*ALTER, once the well's type has been specified with \*PRODUCER or \*INJECTOR. Use with \*TIME or \*DATE.
11. You may adjust the Input/Output controls and the transmissibility multipliers as required.
12. The keywords \*DTWELL and \*DTMAX may also appear in subsequent well changes.

---

## Running Your Simulation

This section discusses methods for running your simulation.

### Overview

STARS requires that the user supply an input data filename, which itself may control all other input and output filenames. That data filename is supplied either by a command-line argument or by interactive prompting. STARS writes information to the various output files, but also writes useful diary information to the standard output device, for example, the screen. The method for specifying the data filename, as well as capturing the diary output, depends upon which of the running modes detailed below is used: Launcher, script or raw command.

### CMG Technology Launcher

The CMG Technology Launcher is a graphical interface to the suite of CMG software including STARS, for Windows platforms. You can drag-and-drop an input data file to the STARS program, causing it to run in a new window. The Launcher handles the passing of the data filename to STARS. However, for a restart run you must supply the name of the input restart IRF, either with keyword \*FILENAME \*INDEX-IN in the data or as response to an interactive prompt. The diary output is directed to the newly created window, which remains after the simulation has finished, and optionally to a file.

### Script

The script method of running STARS is useful when a series of data files are to be run sequentially and Launcher is not available. When using a script, it is recommended that all required filenames (input data and possibly input restart) be specified via command line arguments and/or \*FILENAME keywords so that no prompting is required. The following script found in the STARS release area directory `.../cmg/stars/yyyy.vv/tpl` (where yyyy is the year and vv is that year's version number) can be used "as is" or customized for specific tasks. Any script will contain the raw command described below. If you wish to access an executable corresponding to a particular Launcher icon, the Modify Icon dialog contains the pathname to the executable.

Windows: The CMD batch file `runall.bat` accepts an application name such as "st200610" and runs it with all the data files found in the directory, putting each diary output into a file whose name is the data file appended with ".log". It requires the batch file `runall1.bat` to work. The desired STARS executable file and its associated DLL's must be copied to the directory in which the simulations will be run, or the exe pathname can be changed in the script. Typical usage is

```
runall st200010
```

### Raw Command

In both UNIX and Windows CMD the raw command to run STARS looks like

```
st2006vv.exe -f datafile -log
```

where vv is the particular version number. The executable filename is whatever has been copied or linked, and full pathnames may be used as well. The input data filename can be supplied either by command line argument or interactive prompt. The input restart filename can be supplied either by command line argument, keyword \*FILENAME \*INDEX-IN in the data or interactive prompt. All the allowed command-line arguments are described at the beginning of the Input/Output Control section.

The diary is written to the standard output device, which can be allowed to scroll onto the screen or, more usefully, redirected to a file using “-log”. UNIX platforms can use “&” to run it in background, and “nohup” to keep it running after the user logs off.

### **Running Priority on W2k/WinNT**

Of the priority levels available on W2k/WinNT the default priority used to run a simulation is Normal/Medium, which can significantly reduce the responsiveness of other tasks such as editing large files and RESULTS viewing. Experience has shown that reducing the simulation running priority to Low restores the response time of other tasks while having little affect on the simulation run time.

The following can be done for the process running the simulation, or for the manually created command window before the script or raw command is issued (child processes inherit the lower priority).

Go into Task Manager (right click on task bar), go to the Processes tab, and right click on the process of interest (e.g., "st200010.exe" or "CMD.EXE"). From the menu choose Set Priority and choose Low. You can also make the priority visible in the Processes tab by selecting View>Select Columns ... and check the Base Priority box. This new column is displayed in future invocations of Task Manager, until you disable it.

---

## Improving Numerical Performance

This section discusses methods for diagnosing poor numerical performance of a simulation, along with suggestions for improvement.

### How To Read Diary Output

Besides the usual simulation results written to the output files, a summary of each simulation timestep, called the diary or log, is written to the screen (or a file if it has been redirected). The following is an example diary output.

```
----Timestep----  -----Time-----  -----Production-----
      C          Oil    Gas   Water   GOR   Wat.
  Size   U          ft3/d   bbl/d   bbl/d /bbl   Cut %
No.   days   IT   T   days   yy/mm/dd
-----  --  -  -----  -----  -----  -----  -----  -----
 1   .5000   4   .5000  1980/01/02  7.555     3.280    30.27

--Injection--  Mat  ---Maximum Changes---
Gas    Water  Bal   Pres   Sat   Temp
      Err
ft3/d   bbl/d   %     psi     w/o/g   deg F
-----  -----  -----  -----  -----  -----
 403.6     0    196.6   0.0086w   4.940
```

The Timestep section has four columns: timestep number, timestep size in days, the number of Newton iterations required to solve the non-linear timestep problem and the number of times the timestep failed to converge (cuts). The Time section has the time and date of the timestep. The Production section shows total Oil, Gas and Water production rates, along with GOR and Water Cut. The Injection section shows total Gas and Water injection rates (these phases may be different depending on what is being injected). Then, the Material Balance Error is shown in percent. Lastly, Maximum Changes of pressure, saturation (with phase indicator) and temperature are shown.

### Timestep Size

The timestep size can be due to (a) maximum changes from the previous timestep compared to \*NORM values, (b) maximum timestep size from \*DTMAX, or (c) smaller timesteps due to frequent convergence failures. Check the following if timestep sizes are smaller than expected.

- If at least one of the Maximum Changes is near its \*NORM value, then the timestep size is appropriate, and the only way to increase it is to increase the \*NORM values. Note that there are \*NORM values for phase compositions that are not shown in the diary (but can be shown in the output file). Composition changes rarely control the timestep size for more than a few isolated timesteps.
- Small timestep sizes may be due to frequent cuts. Each cut reduces the timestep size by a factor (1/2 the first try, another 1/3 the next try, etc.). Therefore, cuts every 1 to 3 timesteps may be reversing the increase in timestep size gained by low maximum changes. In this case, the cause of the cuts needs to be investigated.

- Even if maximum changes are small and there are no cuts, it takes a number of steps to increase the timestep size from small values. The formula for timestep size based on maximum changes is described in the explanation for keyword \*NORM, and contains damping which limits the increase in timestep size to a factor of 2.3. Therefore, small values specified by \*DTWELL should be used only when necessary.
- Use \*DTMAX only when necessary. Using \*DTMAX to reduce timestep cuts or other poor numerical performance is not recommended, since it merely masks the real problem which itself may be fixable.

### **Material Balance Error**

The percent material balance error is, for a timestep, the maximum value over all the components and energy. This maximum reported in the diary gives only an overview, and a more detailed report can be found with the timestep summary in the text output file.

Normally, material balance error increases smoothly as the run progresses, ending with an acceptably small value. With default \*CONVERGE values, a typical error at early timesteps is 1e-6%, and final values of .01% to 1% indicate that convergence is under control. If the final material balance error is very small, increasing \*CONVERGE values may reduce the number of Newton iterations while letting the error increase to a still acceptable level.

Large final material balance errors (>5%) may be due to these causes.

- Large convergence tolerances can lead to excessive material balance error. The default \*CONVERGE values are recommended as a starting point.
- Material balance error can be due to insufficient accuracy of iterative matrix solution. The key parameter is \*PRECC, the ratio by which the mean equation residual must be reduced from its initial value before the solution is accepted. Normally \*PRECC should not be increased much above the default value, since a loose matrix tolerance translates directly into high material balance error.
- Material balance error can be due to persistent “failed” matrix solutions. STARS proceeds with the current solution anyway, possibly with high material balance error.

### **Matrix Solver Failure**

The matrix solver has several convergence criteria, which when violated cause an immediate return with the current solution and a “failed” flag. Occasional failures are acceptable, but consistent or continuous matrix solution failures must be dealt with. The following are points to check when persistent matrix solutions failures occur.

- The parameter “mtfail” printed at the end of the text output file is the total number of matrix solver failures for that run. Also, a message is issued to the diary when more than half of a timestep’s Newton iterations experience a matrix solver failure. More than one matrix solver failure for every 5 to 10 Newton iterations likely is too many.
- Keyword \*ITERMAX controls the maximum number of inner matrix solver iterations allowed. The matrix solution “fails” if the residual is not reduced by ratio \*PRECC within \*ITERMAX iterations. If increasing \*ITERMAX does not reduce matrix solution failures, further measures are required. Generally, larger grids need larger \*ITERMAX.

- Small values of \*NORTH (<20) may be constraining the iterative solver. Generally, larger grids need larger \*NORTH.
- Matrix solver parameter \*SDEGREE determines how much fill is used during the iterative matrix solution process. The default is 1, which requires the least CPU and storage. When increasing \*ITERMAX and \*NORTH does not decrease the matrix residual sufficiently, the last resort is increasing \*SDEGREE which should be done by increments of 1. Higher degree increases storage and CPU per iterative significantly, but hopefully converges in fewer iterations. Higher degree may require manual setting of some solver dimension parameters via keyword \*DIM.
- Sometimes the matrix order \*SORDER has an effect on the rate of residual reduction in the matrix solution. There are no concrete rules regarding \*SORDER, except that the default is generally robust and most efficient.

All these matrix solver parameters can be changed at a restart. One efficient technique is to run just one timestep from a restart record, trying variations of the parameters described above to find which is most effective

### How To Read Newton Iteration Details

The following is an example of detailed Newton iteration output triggered by the \*NEWTON option of \*OUTPRN \*ITER. This is from Test Bed #9.

CYC	dpmx	block	dsmx	block	dtmx	block
1	1.00E+02	8,1,1	1.00E-02	16,1,1	4.72E+00	16,1,1
2	1.95E+02	16,1,1	8.56E-03	16,1,1	4.95E+00	16,1,1
3	1.97E+02	16,1,1	8.64E-03	16,1,1	4.94E+00	16,1,1
4	1.97E+02	16,1,1	8.64E-03	16,1,1	4.94E+00	16,1,1

CYC	dxmx	block	urpm	iconv	nitr
1	-1.00E-02	16,1,1	1.00	20	1
2	-8.56E-03	16,1,1	1.00	12	1
3	-8.64E-03	16,1,1	1.00	2	2
4	-8.64E-03	16,1,1	1.00	0	2

On the far left is the Newton iteration, or cycle, number. Next there are five sets of maximum values and block addresses, for pressure, saturation, temperature, gas mole fraction and oil mole fraction, respectively. Each value is the maximum over the entire grid of the change in that variable, and the associate block is the one corresponding to that change. Normal convergence shows as large changes in the first few iterations, and then the changes become constant. The above example shows this clearly, but it may not be apparent when each iteration's maximum change occurs in a different block. The change between iterations is not allowed to exceed the corresponding \*NORM (e.g., pressure \*NORM of 100 in the above example). If any maximum change over the timestep exceeds 3 times the corresponding \*NORM value, then the timestep is cut.

Column URPM is the under-relaxation parameter, and shows that this run has no under-relaxation. See keyword \*UNRELAX.

Column ICONV is the number of unconverged primary iterating variables or equations. The condition  $\text{ICONV} = 0$  is a necessary condition for convergence of the timestep. Generally, ICONV decreases steadily, but occasional small increases is normal. Divergence is indicated by it increasing consistently along with increases in maximum changes. Oscillation is shown by a series of iterations with a constant ICONV value or repeated pattern of values.

Sometimes convergence is held up by only a few variables, in which case ICONV is very small but constant. Use the \*UNCONV option of \*OUTPRN \*ITER to see details of unconverged variables or equations. When the last several iterations all have  $\text{ICONV} = 0$ , some other convergence criteria requires additional Newton iterations, the most common being material balance error too high.  $\text{ICONV} = -200$  indicates that convergence continues because a well equation has not converged.  $\text{ICONV} = -10$  indicates that convergence continues because the minimum allowed pressure has been encountered.

The last column NITR is the number of inner iterations taken by the matrix solver. NITR exceeds \*ITERMAX only when the matrix solution has “failed”. When NITR is very low the \*DEGREE can be decreased if it is above 1. When NITR is consistently near a high \*ITERMAX then it is possible that higher \*NORTH or \*DEGREE may be more efficient.

---

## Optimizing Memory Requirements

This section discusses methods for finding key dimension values that minimize the process (virtual) memory required by STARS for a given block and component set. This method is useful also when maximizing the number of grid blocks that can be used within a given amount of memory, such as the limit associated with 32-bit processors or physical memory installed on a computer.

### Important Concepts

Effective dimensioning of an entity (e.g., array) involves two quantities: the allocated size (maximum dimension) MD and the maximum N actually required. The best situation occurs when  $N = MD$ , called tight dimensioning. When  $MD > N$  the entity is over-sized and space is wasted; when  $MD < N$  then the entity is too small, a fatal condition when the array is used.

The key to understanding dimensioning issues for an array is to realize that the sequence of steps taken by STARS is

- a) Determine MD based on N (if possible), default settings or use of keyword \*DIM,
- b) Allocate array with dimension MD, and
- c) Determine N if not obtained in (a).

In the case of tightly dimensioned arrays, N is known in step (a) so MD is given the value N. However, for some large solver arrays, N is not known until after the array is allocated. Keyword \*DIM allows the user to specify MD directly.

### 32-bit Process Space Limit

A computer based on a 32-bit architecture has a hard limit in how much space can be addressed (and hence allocated) in a single process such as a STARS run. For most such operating systems this limit is 2 Gb. However, the limit is 3 Gb for the following Windows operating systems:

- Windows .NET server family
- Windows XP Professional Edition
- Windows 2000 Datacenter Server
- Windows 2000 Advanced Server

To access this option add switch /3GB to boot.ini on the target machine. The STARS Win32 executable is already enabled for 3 Gb.

This limit shows up as a failure to allocate an array and the dimensioning report indicates that the allocated total for all arrays is below the limit. This means that the total of all array storage for the current set of MD values is too large. In this case you must reduce the MD's so that the total space is under the limit and you get a successful run with \*CHECKONLY. There are two cases:

1. If the limit is reached before the numerical data is read, then you must decrease the number of blocks or components (but drastically since no solver arrays are allocated yet).
2. If the limit is reached after the numerical data is read, then reducing \*SDEGREE or changing to an RCM \*SORDER may help. Reduction of blocks or components still has the most effect.

## **Insufficient Dimensions**

During an array's step (c), above, it may happen that  $N > MD$ , in which case you will get an error message for insufficient dimensioning. This should happen for only some solver arrays and very rarely for connection-length arrays in the Grid Module. Use \*DIM with the appropriate sub-keyword to replace the default value of MD with a sufficient value. If this MD increase puts you over the 2-Gb limit, you must change something else (e.g., reduce blocks, \*SDEGREE, etc.) before proceeding.

Use \*CHECKONLY (NOT ‘run one timestep only’) to test for sufficient dimensioning, since a solver array’s N may change as the number of active wells changes with time. Except for rare cases, data that runs successfully with \*CHECKONLY will not experience a condition of insufficient dimensioning when run without \*CHECKONLY.

## **Minimizing Dimensions**

Keyword \*OUTSOLVR causes STARS to report detailed dimensioning information when determining N for the solver arrays associated with the fluid flow equations. This report is issued at the end of echoing each segment of recurrent data, so there may be many such reports. \*OUTSOLVR is most useful when used with \*CHECKONLY, since a solver array’s N may change as the number of active wells changes with time. Also, \*OUTSOLVR is useful only if there are no allocation failures and no error messages from insufficient dimensioning. The corresponding keyword for geomechanics is \*SITERPG.

## **Procedure to optimize storage allocation**

1. Run the data set with default dimensioning and keywords \*CHECKONLY and \*OUTSOLVR \*ON in the Input/Output Control data section. If you are using \*GEOMECH with AIMSOL then use \*SITERPG in the Geomechanics data, but with \*MAXSTEPS 1 instead of \*CHECKONLY.
2. Deal with any array space under-allocation messages relating to the grid loading as described above.
3. Once the grid array allocation is sufficient, check the .out file for the keyword \*OUTSOLVR and make sure that in the simulator’s “summary of input data” the \*OUTSOLVR \*ON combination is active.
4. Deal with any under-allocation of solver array space messages as described above.
5. When a complete \*CHECKONLY run is successful, examine the .out file (and the .geo file for geomechanics) for the heading “Solver Array Dimensions” and determine the maximum amount of storage required for each array variable listed. To do this you will need to examine the Solver Array Dimension tables reported after ALL recurrent data segments, since the storage may change either way as the run proceeds. The maximum amount may not be given at either the beginning or the end of the run.
6. Set the values of the array variables to the maximum required using DIM statements. Note that not all the array variables identified can be set explicitly as some are calculated from others. Note also that, if you end up with *very* tight dimensioning, any change affecting the grid or well completions may result in insufficient array space being allocated.

---

## Well Management and Group Control

### Specifying the 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 apportionment keyword \*APPORMETHOD. 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.

The group control hierarchy is optional. If a group control hierarchy is used, then not all wells must be attached explicitly to a group. Those wells that are not attached to a group by the user are attached automatically to the internally-generated group 'Default-Group'.

A statement must appear to indicate which groups are connected to the 'FIELD'. The following is an example of valid data input for a case with group control. If any of the \*WELL keywords had been encountered before the \*GROUP keyword, then a warning message would be generated but the well would be attached to the group as directed and simulation would continue. This allows group structures to be defined late in a run, for example at the beginning of the prediction stage after a history has been simulated. Note that in this example, 'Field' is the only group name to appear after \*ATTACHTO but not directly after \*GROUP.

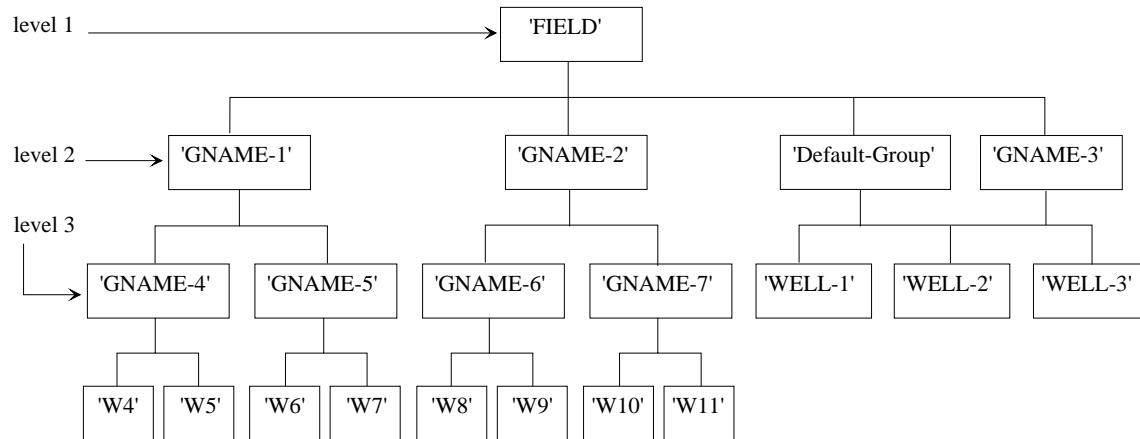
```
*GROUP 'GRP-1'      'GRP-2'      *ATTACHTO 'Field'  
  *WELL 1 'PR-15,10'    *ATTACHTO 'GRP-1'  
  *WELL 2 'INJ-5,11'    *ATTACHTO 'GRP-1'  
  *WELL 3 'PR-03,03'    *ATTACHTO 'GRP-1'  
  *WELL 4 'PR-10,13'    *ATTACHTO 'GRP-1'  
  *WELL 5 'INJH2O'      *ATTACHTO 'GRP-2'  
  *WELL 6 'INJ-6'       *ATTACHTO 'GRP-1'
```

Groups must be defined before any group operating or monitoring constraints are specified. The following example shows a correct sequence.

```
*GROUP 'GRP-1'      'GRP-2'      *ATTACHTO 'Field'  
  *GCONP      'GRP-1'  
    *TARGET    *STO  274.0  
    *MAX       *GOR  400.0  *SHUTMOW  
  *GCONI      'GRP-1'  
    * TARGET   *STW  500.  
  *GCONI      'GRP-2'  
    * TARGET   *STW  350.
```

A maximum of three levels is allowed 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. Wells may be connected to second-level groups (but then groups cannot be), but only wells may be connected to third-level groups.

An example is shown below:



The highest level group is the 'FIELD'. The highest level is not optional. If \*GROUP data lines appear and either no top-level group is specified, for example,

```
*GROUP 'G1' *ATTACHTO 'G2'
*GROUP 'G2' *ATTACHTO 'G1'
```

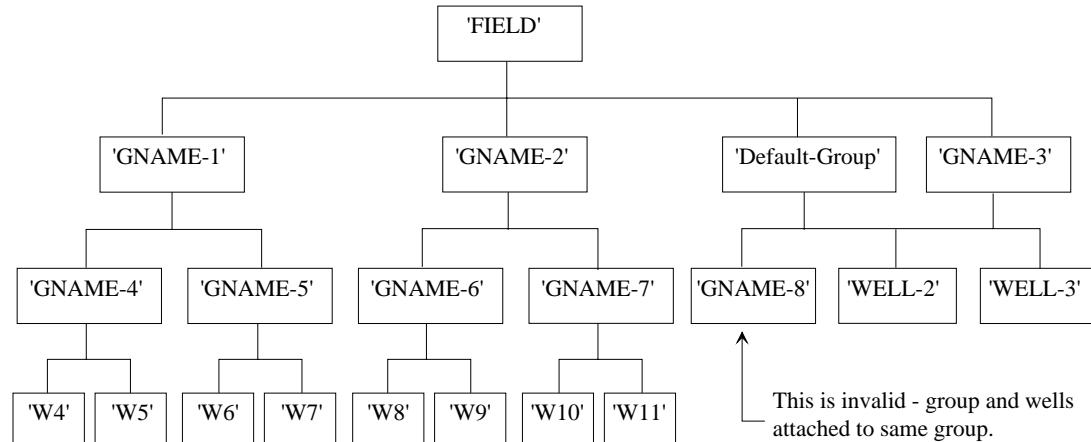
or more than one top-level group is specified, for example,

```
*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.

An example of an invalid well-management hierarchy is given below:



## Production Control

Production controls are entered using keyword \*GCONP 'group\_name'. Target oil, gas, or water production rates can be specified for the centre using \*TARGET \*STO, \*TARGET \*STG, or \*TARGET \*STW.

The target rate is apportioned among contributing producers using one of the available apportionment methods specified by keyword \*APPOR-METHOD. All producers must have a minimum BHP. If none is specified, a minimum BHP of 101.325 kPa is assigned.

Consider the following example of two producers connected to a group with an STO target constraint, assuming SI units:

```
*GCONP  'GRP-1'
      *TARGET  *STO    274.0
      *MAX      *GOR    400.0  *SHUTMOW
  *WELL 1  'PR-15,10'      *ATTACHTO 'GRP-1'
  *WELL 2  'PR-03,03'      *ATTACHTO 'GRP-1'
*PRODUCER 1
  *OPERATE  *MAX  *STO  6.0E+03
  *OPERATE  *MIN  *WHP  5.0E+02
*PRODUCER 2
  *OPERATE  *MAX  *STO  6.0E+02
  *OPERATE  *MIN  *BHP  1.0d+03
```

The following steps are taken to estimate the instantaneous production potential (IPP) of wells 1 and 2 which will serve as the basis for apportioning the group target of 274.0 between the two wells. Well 1 does not have minimum BHP constraint specified, therefore a value of 101.325 kPa (14.696 psia) is assumed. The minimum WHP is converted to a minimum BHP. The larger of the two values is used to compute a maximum oil rate based on the productivity index of well 1 at the specific time in the simulation. The oil rate thus computed is capped to the maximum rate specified for the well, or:

$$\text{IPP (well 1)} = \min [\text{qoil}\{\max(\text{bhp}=101.325, \text{bhp}@\text{whp}=500)\}, 6000.0]$$

A similar procedure is used to compute IPP for well 2

$$\text{IPP (well 2)} = \min [\text{qoil}(\text{bhp}=1000.0), 600.0]$$

## Injection Control

Injection controls are entered using keyword \*GCONI 'group\_name'. A target injection can be specified for the centre using \*TARGET \*STG and \*TARGET \*STW for solvent (gas) and water injection rates respectively, or \*VREP \*GAS and \*VREP \*WATER for gas and water voidage replacement fractions respectively. The target rate is apportioned among all injectors using one of the available apportionment methods specified by keyword \*APPOR-METHOD. All injectors must have a maximum BHP. If none is specified, a maximum BHP of 1,000,000 kPa (147,000 psia) is assumed.

## Individual Well Constraints

Each well can be subjected to its own rate and pressure constraints. If the rate allocated by the centre violates the well's own constraint, the well's constraint will be used. For example, if the gas injection rate of a well allocated by the group exceeds the well's maximum gas rate, the well's maximum gas rate will be used. If a well would violate its min. BHP limit while producing the allocated rate, the well will then produce at its minimum BHP. In these

situations, the rates for the other wells will be readjusted to compensate for the differences. If all wells are under their own constraints, then the group target rate will not be maintained.

Thus, the group production target rate should be less than the sum of the max. production rates from all producers in the group; and the target injection rate should be less than the sum of the max. allowable injection rates from all group injectors, if the group target is to be met.

### Introducing Group Control After Start of Simulation

Group control can be instituted after the simulation has started, at a well change time or upon restart. The simplest example consists in introducing a field target after the simulation has run for some time. Assume that a run has been started with no group structure referred to, i.e. with wells defined using \*WELL lines with no \*ATTACHTO subkeywords defining parent groups. To establish a group structure, it suffices (at a well change time) to introduce the single line

```
*GROUP 'Default-Group' *ATTACHTO 'Field'
```

The name 'Default-Group' must appear exactly as shown in the above line, since it is an internally set name. The top-level group is given the name 'Field' in this example but the user is completely free to choose this name (up to a maximum length of 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

```
*GCOMP 'Field' *TARGET *STO 400.0
```

Here it is important that 'Field' match exactly the character string after \*ATTACHTO in the \*GROUP line; if 'Campo' had been used above it would have to be used here.

### Data Input

The following are all the keywords related to the group well control:

#### Group Specification

```
*WELL 'well_name' (*ATTACHTO 'group_name')
```

#### Production Control

```
*GCOMP 'group_name_1' 'group_name_2' ... 'group_name_n'  
      (*MAX)      (*STO)      value (*STOP)  
      (*TARGET)    (*STG)      (*CONT)  
      (*STW)       (*SHUTALL)  
      (*STL)       (*SHUTMOWS)  
      (*BHF)       (*SHUTMOW)  
      (*SHUTMOL)  
      (*SHUTMOLDOWN)  
      (*SHUTMOLUP)  
      (*RECYCLE)   (*GAS)      recyc_frac  
      (*WATER)  
      (*VREP)  
      (*PMAINT)    (*PMSECT)   'sector_name'  
      (*PMTARG)    p_targ  
      (*PMCOEF)    c1 c2 c3  
*APPOR-METHOD *PROD 'group_names'  
      (*IP | *GUIDE | *INGUIDE | *PRIOR)
```

```

*GUIDEP      (*STO) ('group_names' | 'well_names') guide_rates
(*STG)
(*STW)
(*STL)
*PRIOR-FORM *PROD 'group_names'
(*PRIOR-RATE   (*MRC | *BHP (bhp_val)))
(*PRIOR-CTRL   freq trc_min trc_max)
(*PRIOR-NUMER  A0 A1 ... Anph)
(*PRIOR-DENOM  B0 B1 ... Bnph)
*GCPOFF ('group_names' | 'well_names')
*GAPPOR 'group_names' *AUTODRILL (*ON)
(*OFF)
*GCONN 'group_name_1' 'group_name_2' ... 'group_name_n'
(*GOR)           value (*STOP)
(*WCUT)          (*SHUTALL)
(*WGR)           (*SHUTMOWS)
(*MAXGAS)         (*SHUTMOW)
(*MAXSTW)         (*SHUTMOL)
(*SHUTMOLDOWN)
(*SHUTMOLUP)
(*MINOIL)         value (*STOP)
(*MINGAS)         (*SHUTALL)
(*MINBHF)

```

### Group Injection Control

```

*GCONI 'group_name_1' 'group_name_2' ... 'group_name_n'
(*MAX)           value (*STOP)
(*TARGET)        (*STG)          (*CONT)
(*STW)
(*BHG)
(*BHW)
(*RECYCLE)       (*GAS)          recyc_frac
(*WATER)
(*VREP)          (*GAS)          vrep_frac
(*WATER)
(*GMKUP)
(*WMKUP)
(*PMAINT)        (*GAS)          (*PMSECT)    'sector_name'
(*WATER)          (*PMTARG)     p_targ
(*PMCOEF)        c1 c2 c3
*APPOR-METHOD (*GASI | *WATI) 'group_names'
(*IP | *GUIDE | *INGUIDE | *PRIOR)
*GUIDEI ('group_names' | 'well_names') guide_rates
(*STG)
(*STW)

```

```

*PRIOR-FORM (*GASI | *WATI) 'group_names'
(*PRIOR-RATE      (*MRC | *BHP (bhp_val)))
(*PRIOR-CTRL      freq   trc_min   trc_max)
(*PRIOR-NUMER     A0 A1 ... Anph)
(*PRIOR-DENOM     B0 B1 ... Bnph)
*GCIOFF (*GAS) ('group_names' | 'well_names')
(*WATER)
*GAPPOR 'group_names' *AUTODRILL (*ON)
(*OFF)

```

## Limitations

The following limitations currently apply to the well management and group control module.

1. A maximum of three group levels is allowed.
2. The topmost group level (the field) cannot have wells attached to it, but only other groups.
3. Groups to which wells are attached cannot have other groups attached to them.
4. 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.
5. Group controlled injection fluids are limited to gas (solvent) and water only. Oil injection is not supported.
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 or close layers. If the well is perforated horizontally, the behavior of this option may be unpredictable.

---

## Parallel Processing

Parallel processing allows STARS to run a given data set in significantly less clock time. This tutorial (1) describes parallel processing in some detail and (2) discusses the associated issues of timing, tuning and practical speed-ups.

### Types of Parallelism

Current computer hardware supports two general types of parallelism based on the configuration of memory. A **shared-memory** machine shares all the available memory amongst all the CPUs via a fast high-capacity bus. This configuration gives the advantage of high inter-CPU communication speed.

A **distributed-memory** system, often referred to as a “cluster”, consists of a number of nodes that communicate via a high-speed network; each node has one or several CPUs and its own memory. The speed of a distributed-memory configuration usually is limited by the network hardware, but there is no real limit on the number of machines which may be networked together.

STARS currently supports the shared-memory paradigm. However, work is also in progress for supporting distributed-memory systems.

### OpenMP

OpenMP is the “open specification for multi-processing”, a coding standard developed through the collaboration of industry, government and academia. It is used to specify shared-memory parallelism in programs that are run on systems configured for Symmetric Multiprocessing (SMP).

Parallel processing in STARS is achieved by placing throughout the program code OpenMP directives that instruct the compiler to generate multi-threaded binary code. When STARS is executed in parallel, the program starts as a single “master” thread. Upon reaching a parallel code region (generally a loop), the master thread spawns a number of “slave” threads, which form a “team”. Each thread in the team executes the parallel code region simultaneously using its own private portion of the overall data. When the team is finished, all threads are synchronized, control returns to the master thread, and the program continues.

### Jacobian Domains and the Parallel Solver

Building of the Jacobian matrix in parallel is achieved through domain decomposition, where the entire grid is partitioned into subdomains and each thread generates in parallel the Jacobian matrix entries for a different subdomain. Jacobian domain decomposition is controlled using keywords **\*DPLANES**, **\*DTYPE** or command-line argument **-doms**. Other miscellaneous tasks are done in parallel when Jacobian domain decomposition is enabled. For each task performed when Jacobian domain decomposition enabled, the numerical result is the same no matter how the grid is partitioned into subdomains.

CMG’s parallel solver PARASOL is used to solve the linear system of equations in parallel. Similar in concept to domain decomposition, the reservoir is first partitioned into disjoint sets of blocks known as solver classes which are further organized into levels. Reservoir partitioning is controlled using keyword **\*PPATTERN** or command-line argument **-parasol**. For a given system of linear equations the numerical result from PARASOL can vary with the number of threads. See **Tuning Solver Performance** below.

## Number of Threads

STARS has been tested in parallel on all of the supported platforms. As of 2005 STARS has been run with up to eight threads but STARS contains no built-in limit for the number of threads. Parallelization of more code regions will improve speed-up for more processors so that unparallelized code does not dominate run times.

## Licensing

Licensing for the parallel-processing feature must be enabled in order to run STARS in parallel. The number of processors for which STARS is licensed, and the number of available processors, will determine the number of threads which can be practically used.

## Parallel Processing Keywords and Command-line Options

The following keywords are used to specify parallel processing in STARS:

- \*SOLVER \*PARASOL – use CMG’s parallel iterative solver.
- \*PNTHRDS – set the number of threads to be used.
- \*PPATTERN – define the Parasol class partitioning pattern.
- \*DPLANES – specify the number of planes per Jacobian domain.
- \*DTYPE – explicitly set the domain numbers for individual blocks.
- \*CHECKRB – control red-black ordering for Parasol.
- \*PDEGAA – set the factorization degree within Parasol classes.
- \*PDEGAB – set the factorization degree between Parasol classes.
- \*PNPROSL – choose the number and scaling of GMRES vector operation classes.

The following command-line options can also be used to specify parallel processing in STARS as an alternative to keywords (e.g., used by Launcher):

**-doms**, which is equivalent to:

\*DPLANES (note that -doms overrides both \*DTYPE and \*DPLANES)

**-parasol *n***, which is equivalent to all of the following keywords together:

- \*SOLVER \*PARASOL
- \*PPATTERN \*AUTOPSLAB *n*
- \*PNPROSL *n*
- \*PNTHRDS *m*, where *m* is the smaller of *n* and the number of logical CPUs available. The parameter *n* is used to specify the number of threads to be used, and in general should not exceed the number of logical CPUs available. Note that if *n* is omitted, it will be defaulted to 2.

Please refer to the Numerical Methods Control section for a detailed explanation of these keywords and suggestions for their use.

## Timing and Speedup

A STARS data set which has been suitably tuned to run in parallel (using \*SOLVER \*PARASOL) will have a significantly lower elapsed time than the same data run without parallel (using the default \*SOLVER \*AIMSOL).

Speedup is defined as the ratio of elapsed times for lower and higher numbers of threads. For example, if a particular data set runs in 45 minutes in serial but 30 minutes in parallel, the speedup is 45/30 or 1.5. Theoretical maximum speedup can be represented by Amdahl's Law:

$$\text{speedup} = 1 / (s + p/n)$$

where

$p$  is the fraction of CPU time executing parallel code,

$s$  is the fraction of CPU time executing serial code ( $1-p$ ), and

$n$  is the number of CPUs used.

The value of  $s$  includes code in which parallel directives are not currently used (for example, data input and results output), as well as the overhead associated with OpenMP (for example, creating and administering threads).

In cases where the number of Newton or solver iterations varies significantly, speedups should be calculated on a per-Newton iteration basis.

### Tuning Solver Performance

For a given system of linear equations the numerical result from PARASOL can vary with the number of threads. In fact, the result from PARASOL with one thread (non-parallel) can differ from the result from AIMSOL. These result differences are due to incomplete convergence of the iterative solution for different kinds of approximations in the solution of the equations. These result differences usually are small and show up as slight differences in numbers of Newton iterations, matrix failures or material balance. Sometimes the difference between results can be significant, often indicating that solver or Newton iteration convergence criteria in the Numerical Methods Control section should be adjusted. See “Improving Numerical Performance” in the Tutorial section.

It is not uncommon for a data set that has good solver performance for few threads to experience significantly worse performance with more threads. The most common cause is insufficient solver convergence, usually indicated by large numbers of solver failures, timestep convergence failures or large material balance errors.

This is illustrated by a real case. STARS template sthrw007 was run on an IBM p550 using 1, 2 and 8 threads, with the following run results:

No. of Threads	Newton Iterations	Cuts	Matrix Failures	Elapsed (sec)	Speedup
1	1575	2	0	1126	1.00
2	1581	2	2	635	1.77
untuned 8	1618	3	398	360	3.13
tuned 8	1575	2	0	357	3.15

For 1 and 2 threads, the numerical performance and production are close, as expected. However, for 8 threads, the untuned result is quite different. The large number of matrix solver failures (398) indicates trouble with solver convergence. Inspection of the reservoir performance shows significant differences from the other runs, an unacceptable result.

To reduce the matrix failures, both \*ITERMAX and \*NORTH each were increased from default values of 30 to 50. The tuned 8-thread result closely matches the other thread cases in both numerical and reservoir performance.

### CPU Breakdown

When command-line option **-cputime** is used, detailed statistics of CPU and elapsed times are written at the end of the log file. These statistics can be used to determine where most of the time is being spent running a particular data set. For the example sthrw007 used above, these statistics show that the majority of the processor time is spent in two areas of the code: Jacobian Building (JBuild) and matrix Solver.

No. of Threads	Task	CPU (sec)	% of Total CPU	Clock (sec)	% of Total Clock
1	JBuild	604.94	53.79	605.45	53.78
1	Solver	301.15	26.78	301.35	26.77
8	JBuild	632.62	48.96	160.03	44.87
8	Solver	431.01	33.36	108.75	30.49

For the single-thread run, the CPU and Clock (elapsed) times are nearly identical, as all the work is being done by one thread. For the 8-thread run, the CPU times are much larger than the Clock times since the same amount of work as in the single-thread run is now being spread over 8 threads. The result is a decrease in Clock time from 605 to 160 s for JBuild, and from 301 to 108 s for Solver. The combined speedup for these two portions of the code is 3.37.

Note that the CPU times for the 8-thread run (632 s for JBuild and 431 s for Solver) are larger than those of the single-thread run (604 s for JBuild and 301 s for Solver), due to the overhead of OpenMP as well as the use of a different linear solver (\*PARASOL instead of \*AIMSOL).

# 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, \*TRANSLATE and \*RANGECHECK.

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**

A primary keyword should appear on a new line and may be followed by its data and subkeywords on the same line and/or subsequent lines. A keyword is primary when it appears in the upper right-hand corner of the corresponding manual page, for example, \*COMPNAME. A sub-keyword that has the same name as a primary keyword in another data section is not primary itself. For example, \*WELL is a primary keyword in the Recurrent Data section but may appear as a secondary keyword of, and on the same line as, \*OUTPRN in the I/O Control section.

Each row of a table must appear on a new line, since this defines the columns.

STARS allows primary keywords to be located on the same line for backward compatibility, but the practice is not recommended. Builder does not support this practice, and STARS may disallow it in a future version.

## **CASE**

Keywords and alphanumerical strings may be in upper case, lower case, or any combination. Filenames must conform to the requirements of the operating system being used, for example, upper case for IBM mainframe systems.

## **LINE LENGTH**

Only the first 512 characters in a line are processed, and any character after that is ignored.

## **DELIMITERS**

Keywords, numbers, and character strings must be separated from each other by blanks, commas, or new-line characters. Consecutive commas, with nothing except blanks between them should not occur in the data file.

## **CHARACTER STRINGS**

Character strings ALWAYS must be enclosed in either a pair of single quotes (e.g. '5-35-48-W5') or double quotes (e.g. "5-35-48-W5"). When inserting either type of quote in the string, enclose the string in the other quote type, e.g., 'This is the "right" way.' or "Land's End". When a string's maximum length is specified, characters after that maximum will be ignored.

## **TRANSLATION**

You can use your own keyword for any main keyword if you define the translation rule using \*TRANSLATE.

## **NUMBERS**

Numbers are input in free format. Real numbers do not require decimal points.

Exponentiation is indicated by 'E', 'e', 'D' or 'd'. Numbers must not contain embedded blanks.

If an integer is expected, and a number with a decimal fraction is read in, an error message will be issued, and the program will stop.

The following are examples of valid real numbers:

```
25.040  
-3  
1.23E+02  
0.02D-4  
34.e02  
+2.3  
+.3  
-.3
```

The following are NOT valid real numbers:

```
34. E 02 <-- blanks in number  
- 34.E02 <-- blank in number  
34.E.2 <-- decimal in exponent
```

Sequences of numbers may be separated either by commas or by blank spaces.

## **REPEAT COUNT**

There is a simple way to input multiple sequential occurrences of a number. Suppose you have five numbers in order:

```
.23 .23 .23 .41 .27
```

There are two ways to input these numbers. One is to write them as they appear directly above.

However a shortcut measure is to write them using the multiple occurrence indicator ("\*").

Since the first three numbers in sequence are the same you can write the numbers this way:

```
3*.23 .41 .27
```

Note that there MUST NOT be a space either before or after the "\*".

## **INTEGER RANGE**

In any instance where a sequence of INTEGER values is required, a colon must be used to indicate a range of values from one integer to another integer. Blanks cannot be present between either integer and the colon. For example:

```
1 2 3 4 6 10 11 12  
and  
1:4 6 10:12
```

are two equivalent ways of giving the same sequence of INTEGERS. Note that this method of input will not work if real numbers are expected.

## TABLES

The keyword documentation sometimes indicates that a table of data must be entered. All the required data items (columns) are listed in order. Always enter the data in the order shown.

For each row, a value is expected for each mandatory column. In addition, you may elect to enter values for an optional column (shown enclosed in round brackets in the table syntax); if so, you must enter a value for each row.

An example of such an event includes the water-oil relative permeability tables (\*SWT keyword). Pcow is optional, and need not be entered, but in this case the user has capillary pressure data.

*SWT			
**Sw	krw	krow	(Pcow)
0.2	0.0	1.0	45.0
0.2899	0.022	0.6769	19.03
0.3778	0.018	0.4153	10.07
0.4667	0.061	0.2178	4.09
0.5556	0.143	0.0835	1.80
0.6782	0.289	0.0123	.50
0.7561	0.450	0.0	.10
0.8325	0.780	0.0	.0
0.9222	1.000	0.0	.0
1.0000	1.000	0.0	.0

If the capillary pressure is not used (pcow = 0), then the table would be entered as

*SW		
**Sw	krw	krow
0.2	0.0	1.0
0.2899	0.022	0.6769
0.3778	0.018	0.4153
0.4667	0.061	0.2178
0.5556	0.143	0.0835
0.6782	0.289	0.0123
0.7561	0.450	0.0
0.8325	0.780	0.0
0.9222	1.000	0.0
1.0000	1.000	0.0

Tables from different sources may be merged automatically by using the \*INT table entry option.

## 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.

---

## 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:

If \*RANGECHECK is absent in a data section, then \*RANGECHECK \*ON is assumed.

\*RANGECHECK starts as \*ON at the beginning of each data section as well as each segment of recurrent data (data between adjacent \*TIME or \*DATE lines).

\*RANGECHECK without \*ON or \*OFF implies \*ON.

### CONDITIONS:

This keyword may appear anywhere in the data file, as many times as needed.

### EXPLANATION:

Most input data is examined to determine if it is within an expected range of values.

Specifying \*RANGECHECK \*OFF will disable non-critical data range checking until \*RANGECHECK \*ON or the end of the data section is encountered.

\*RANGECHECK \*OFF also will suppress the printing of all "warning" messages. Error messages always will be printed.

Example: To override the pressure \*DNORM range:

```
*RANGECHECK *OFF
*NORM *PRESS 500
*RANGECHECK *ON
```

It is strongly recommended that range checking be kept enabled for as much of the data file as possible.

---

## 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 '*pathname*'

### 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.

The *pathname* must be enclosed by quotes.

The *pathname* must be a valid path name in the file system.

### EXPLANATION:

When \*INCLUDE is encountered, secondary input file *pathname* is opened and data is read from it. When the end of the secondary file is reached, that file is closed and data reading continues in the primary (or original) input file.

When *pathname* is an absolute file path name, it is used directly. An absolute path name starts with a directory delimiter, and on Windows there may be a leading drive (letter and colon).

When *pathname* is a relative file path name, it is relative to the directory containing the main data file. In this case *pathname* is converted internally to the corresponding absolute or relative path name.

The two system types have different directory delimiters - slash for Unix and backslash for Windows. Each delimiter in *pathname* that is not valid for the current system type is converted internally to the delimiter that is valid. This allows you to use a data file containing *pathname* with one delimiter type on both system types.

In all cases a line in the log/diary reports the internal path name used to open the file.

### Example:

```
** Main data file is "/unix1/user/proj3/data/case23.dat".
** Include files are in "data" sub-directory "includes".
*INCLUDE 'includes/grid_data.txt'
** Path name is adjusted to
**      "/unix1/user/proj3/data/includes/grid_data.txt".
```

---

## Controlling Data File Listing (Optional)

\*LIST, \*NOLIST

### PURPOSE:

\*LIST specifies listing the input data file, from this point forward, to the output print file.

\*NOLIST specifies not listing the input data file to the output print file, starting immediately after the current line.

### FORMAT:

\*LIST

\*NOLIST

### DEFAULTS:

Optional keywords. Default: \*LIST

### CONDITIONS:

\*LIST or \*NOLIST may appear at any point in the data file, but must be on a line by itself.

### EXPLANATION:

By default, the entire data file is listed to the output print file prior to the start of the simulation run, with the exception of a limit of 20 echoed lines for each grid-array keyword. If a \*NOLIST keyword is inserted in the data file, the data file is not listed from the point of the \*NOLIST keyword until a \*LIST keyword or the end of data file is reached. Keyword \*NOLISTLIM disables the limiting of grid-array keyword data.

---

## **Changing the Comment Indicator (Optional)**

**\*COMMENT**

### **PURPOSE:**

\*COMMENT changes the two character sequence that denotes the beginning of a comment.

### **FORMAT:**

\*COMMENT 'ab'

### **DEFINITION:**

ab

A two-character string denoting the start of a comment. The string 'ab' must be enclosed in quotes.

### **DEFAULTS:**

Optional keyword. Default: \*COMMENT '\*\*'

### **CONDITIONS:**

\*COMMENT may appear at any point in the data file, but must be on a line by itself. All subsequent comments following the appearance of the \*COMMENT keyword must be preceded by the two-character sequence 'ab'.

### **EXPLANATION:**

By default, comments in the data file are denoted by the character string '\*\*'. This may be changed by using the \*COMMENT keyword.

Example:

```
*COMMENT '--'  
*TRANSLATE 'KX' 'PERMI' -- This is a translate  
-- rule
```

From this point on in the data file all comments should begin with '--'. In the above example the two lines beginning with '--' are comments.

---

## **Changing the Keywords by Using Translate Rules (Optional)**

**\*TRANSLATE**

### **PURPOSE:**

\*TRANSLATE changes or translates your own favorite keyword into a CMG simulator recognizable keyword.

### **FORMAT:**

\*TRANSLATE 'your\_keyword' 'CMG\_keyword'

### **DEFINITION:**

**your\_keyword**

A single-word keyword that you want the simulator to recognize. The allowed characters are those in the character set specified in subroutine RDLINE in the simulator source code; no blanks, commas or asterisks are allowed. You may add any character that your computer operating system supports to this character set. Enclose the string in single quotes.

**CMG\_keyword**

The CMG simulator keyword (WITHOUT asterisk) that you want to replace. This must be a valid keyword recognized by the simulator. This must be enclosed in single quotes.

### **DEFAULTS:**

Optional keyword. Default: Use the internal simulator keywords.

### **CONDITIONS:**

\*TRANSLATE may appear at any point in the data file, but must be on a line by itself. Subsequently, a simulator keyword may be referred to by using either 'your\_keyword' (defined by a \*TRANSLATE keyword) definition or the internal simulator keyword 'CMG\_keyword'.

### **EXPLANATION:**

If you need to redefine a keyword because you want to make the keyword more meaningful to yourself, or simply for convenience, the \*TRANSLATE keyword will accomplish this task.

Example:

```
*TRANSLATE 'KX' 'PERMI'
```

This translate rule translates the \*KX or KX keyword such that the simulator recognizes this to mean \*PERMI. Subsequent to this keyword \*KX, KX, \*PERMI, or PERMI may be used to refer to the \*PERMI keyword.

A keyword may have more than one translate rule,

Example:

```
*TRANSLATE 'KX' 'PERMI'  
*TRANSLATE 'x_permeability' 'PERMI'  
*TRANSLATE 'permx' 'PERMI'
```

## User Block Address

\*UBA

### PURPOSE:

Specify address for any block in the grid.

### EXPLANATION:

The User Block Address (UBA) is a natural extension of the familiar I-J-K notation for addressing blocks, but allows the user to refer to any block in the advanced grids available in CMG simulators. UBA is based on the idea that, even for the most complex grids, the relationships between grids and blocks can be represented as a tree diagram with the fundamental grid at the base node. The UBA starts by giving the fundamental grid block in I-J-K notation, e.g., "3,6,2" or "3 6 2". Delimiting blanks or commas must separate the block indices.

### Refine Grids

To refer to a block at a finer grid level, the fundamental I-J-K is followed by a slash (/) and the next level of refined grid block in I-J-K notation, and so on, to the block of interest.

Delimiting blanks or commas must surround a slash. Multiple refinement levels are allowed.

For example, if fundamental block (3,6,2) contains a 3x3x1 refined grid, then the UBA for one of the fine blocks is "3,6,2 / 2,3,1". UBA "3 6 2" still refers to the fundamental "parent" block.

### Naturally Fractured Grids

In the naturally fractured grid options, a fundamental "spatial" block is split into two "storage" blocks: fracture and (optionally refined) matrix. Indicate which of the two with UBA qualifiers MT or FR after the indices, e.g., "10,14,3 FR" and "10,14,3 MT". If the block is fractured and neither FR nor MT is present, FR is assumed.

The MINC and Subdomain options involve refinement of the matrix portion of the "spatial" block that is referenced with the refined-grid notation. For example, MINC block 2 (second from the inside) located in fundamental block "10,14,3" is "10,14,3 / 2,1,1 MT".

### Discretized Wellbore Grids

Since wellbore blocks are refined grids, they may be addressed as such. With few exceptions UBA qualifiers WB and TU may be used to replace the trailing " / n,1,1". This alternative syntax is unique and interchangeable. These UBA qualifiers allow for better readability as well as backward compatibility of existing data. A wellbore qualifier may be used together with the full refined grid syntax, but it must correspond to the refined block in question (i.e., annulus or tubing).

There are three different cases for using these qualifiers. In a non-circulating wellbore block, "WB" means " / 1,1,1". In a circulating wellbore block, "TU" means " / 1,1,1" (the tubing) and "WB" means " / 2,1,1" (the annulus).

The DW-in-Hybrid option locates a wellbore grid in the centre block of a hybrid grid. If the UBA before the qualifier denotes a block that contains a hybrid grid that itself contains a wellbore grid, then the qualifier indicates the corresponding wellbore block " / 1,1,1 / n,1,1 ".

## UBA Ranges

Some keywords support the entry of ranges of blocks in the UBA format, for example, \*PERF. A UBA range consists of the UBA described above, except that at least one integer grid index is replaced by an integer range (e.g., “3 6 1:4 / 2 2 1:3 MT”). Remember that in an integer range the first number must not exceed the second number.

For some keywords the order of processing the individual blocks in the range is significant. The following shows the sequence in which individual blocks are processed, with each point in order of priority.

1. In each integer range, the values are processed from the first (lower) number to the second (larger) number.
2. In each level, the I indices are processed first, then the J indices, then the K indices. This is commonly known as “natural” order.
3. For multiple grid levels, the lowest (i.e., finest, rightmost) grid level is processed before the next higher grid level.

Example: The UBA range “1,3:4,5 / 6,7:8,9:10” is processed in the following order:

```
1 3 5 / 6 7 9  
1 3 5 / 6 8 9  
1 3 5 / 6 7 10  
1 3 5 / 6 8 10  
1 4 5 / 6 7 9  
1 4 5 / 6 8 9  
1 4 5 / 6 7 10  
1 4 5 / 6 8 10
```

## UBA for Output

User block addresses appear in many places in the simulator output. For output purposes only, a descriptive 2-letter abbreviation is appended to the UBA of each block that is not a normal block in single porosity system.

NL	null block
PN	pinch out block
ZP	zero porosity block
FR	fracture block in dual porosity system
MT	matrix block in dual porosity or dual permeability
Mi	MINC block, where i = 1 as the innermost block
Si	Subdomain block, where i = 1 as the topmost block
WB	discretized wellbore (annulus) block
TU	discretized tubing block

Where an output field does not have sufficient length for the entire address, the presence of additional refinement levels will be indicated by "+", e.g., “23,13,12+ WB”.

---

## 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* (*comp\_name*) (*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

*comp\_name*

Character string in quotes, that must be a component name specified via  
\*COMPNAME. This is required for component-dependent grid arrays.

*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

\*RG *uba\_range*

\*WELLBORE *uba\_range*

\*ANNULUS *uba\_range*

\*TUBING *uba\_range*

\*ALLELEM

If no *array\_qualifier* is present then \*ALLELEM is assumed, except in chapter “Reservoir Description” where \*MATRIX is assumed. These *array\_qualifier* keywords are described separately.

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.

### *read\_option*

The read options are

- \*CON
- \*IVAR
- \*JVAR
- \*KVAR
- \*ALL
- \*IJK
- \*BINARY\_DATA
- \*EQUALSI

These *read\_option* keywords are described separately.

All these *read\_options* except \*IJK ensure definition of each block in the grid. \*IJK 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 *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.

## **EXPLANATION:**

A grid block is divided into at most two parts: matrix and fracture. All other grid, wellbore and natural fracture options are treated as local refined grids.

For example, a discretized wellbore is treated as a refined grid contained in another grid block. Let the coarse block address be i j k. The address of the (possibly naturally fractured) formation surrounding the discretized wellbore is i j k. The address of the wellbore is i j k / 1 1 since it is the first block in the 'refined' grid. If the wellbore has two streams, the tubing address is i j k / 1 1 1 and the annulus is i j k / 2 1 1. An array assignment to block i j k will be inherited to the refined grid it contains by default.

Another example is the MINC natural fracture option. The MINC option divides a block into a fracture and several distinct matrix blocks; these matrix blocks are treated as a refined grid of the matrix portion of the coarse or 'parent' block. Array assignment to block i j k will be inherited by default to the refined (MINC) grid it contains. For parent block i j k, the innermost matrix block has address i j k / 1 1 1, the next block is 1 1 1 / 2 1 1 and so on.

---

## 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:

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 *uba\_range*

### DEFINITIONS:

*uba\_range*

A User Block Address (UBA) range or single UBA, without UBA qualifiers MT, FR, WB and TU.

### CONDITIONS:

When \*RG *uba\_range* is used together with \*EQUALSI or \*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:

Refined grids are initially defined using the \*REFINE keyword in the RESERVOIR DESCRIPTION section.

By default, all refined grid blocks are assigned the values that are 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 fundamental block (1,1,1) contains a 3x2x1 refined grid. Use the following to enter a different porosity value of each of the refined grid blocks, with some later modification.

```
*POR *RG 1 1 1 *ALL  
.08 .079 .078 .081 .08 .076  
*MOD + 0.03
```

**Example:** Suppose fundamental blocks (1,1,1) and (1,1,2) each contain a 3x2x1 refined grid. Use a UBA range to specify the same porosity distribution in each locally refined grid.

```
*POR *RG 1 1 1:2 *ALL  
.08 .079 .078 .081 .08 .076  
**MOD not allowed with UBA range
```

---

## Entering Wellbore Grid Properties      \*WELLBORE, \*ANNULUS, \*TUBING

### PURPOSE:

Assign data to wellbore element of grid blocks.

### KEYWORD:

*WELLBORE	i1(:i2)	j1(:j2)	k1(:k2)
*ANNULUS	i1(:i2)	j1(:j2)	k1(:k2)
*TUBING	i1(:i2)	j1(:j2)	k1(:k2)

### DEFINITIONS:

#### \*WELLBORE

Indicates that the data is to be assigned to the wellbore element(s) in the grid block. This applies to both annulus and tubing for circulating wells.

#### \*ANNULUS

Indicates that the data is to be assigned to the annulus element in the grid block. It also applies to the wellbore when no tubing exists.

#### \*TUBING

Indicates that the data is to be assigned to the tubing element of the grid block.

### CONDITIONS:

These array input qualifiers are valid only when the discretized wellbore option is enabled via the reservoir description keywords \*WELLBORE and/or \*CIRCWELL.

### EXPLANATION:

Example: To specify the different initial oil saturations in a circulating well found in blocks 1:6 2 5:

```
*SO *MATRIX *CON 0.70      ** 70% in matrix
*SO *ANNULUS 1:6 2 5 *CON 0.05 ** 5% in annulus
*SO *TUBING 1:6 2 5 *CON 0    ** None in tubing
```

To specify no oil in annulus and tubing:

```
*SO *MATRIX *CON 0.70      ** 70% in matrix
*SO *WELLBORE 1:6 2 5 *CON 0 ** 0% in annulus/tubing
```

These array qualifiers are the preferred way to refer to discretized wellbore blocks.

---

## Assigning Grid Properties to all Elements

\*ALLELEM

### PURPOSE:

Assign data to all elements of grid blocks.

### KEYWORD:

\*ALLELEM

### EXPLANATION:

This array input qualifier indicates the data is to assigned to all elements of a grid block. This qualifier is necessary only if a natural fracture option is used. Since this is the default, this keyword is not needed explicitly.

Example: To specify the same initial temperature of 40.5 degrees in both fracture and matrix:

```
*TEMP *ALLELEM *CON 40.5  
-OR-  
*TEMP *CON 40.5
```

---

## Constant Value Arrays

\*CON

### PURPOSE:

\*CON indicates that a constant value is entered for all array elements. The value may be entered on the same line or the next line.

### KEYWORD:

\*CON value

### EXPLANATION:

Example: Assume you have a reservoir with a constant value of porosity of 0.16, and a constant permeability in the I direction of 100 md.

```
*POR *CON  
0.16  
*PERMI *CON 100.
```

---

## Array Input In IJK Notation

\*IJK

### PURPOSE:

\*IJK assigns a constant value of a grid property within the region defined by the minimum and maximum block number in each of the three directions.

### KEYWORD:

\*IJK  
{ i1(:i2) j1(:j2) k1(:k2) value }

### DEFINITIONS:

i1(:i2)                    I-direction grid block index range.

j1(:j2)                    J-direction grid block index range.

k1(:k2)                    K-direction grid block index range.

value                      Constant value of the array for the defined region.

{ ... }                    Indicates that any number of lines (but at least one) may be used.

### CONDITIONS:

In general, you must define ALL blocks in the grid with any one usage of \*IJK. Care must be taken with usage of \*IJK, because it is possible to omit some blocks in the assignment (unlike the other array- reading options). If you do skip at least one block a fatal error message will inform you. In all the data sections except RECURRENT, it is safest to use one of the other array-reading options in conjunction with the \*MOD option.

The \*IJK array-reading option is most useful in the RECURRENT DATA section, where assignments to grid blocks usually are over-writing of default or previously assigned data. Referring to only select grid blocks is allowed in that section.

### EXPLANATION:

The \*IJK array reading option assigns a value of a grid property within the region defined by the block number ranges in each of the three directions. Later lines in the same array variable invocation will overwrite previous lines if they refer to the same grid blocks.

For example, in assigning porosity to a 10 x 10 x 3 grid where the value is the same except in a 5 x 5 region, the usage

```
*POR *IJK 1:10 1:10 1:3 0.246  
      1:5   1:5   1    0.17  
-OR-  
*POR *CON 0.246  
*MOD 1:5 1:5 1 = 0.17
```

(which is preferred) are correct, whereas

```
*POR *CON 0.246  
*POR *IJK 1:5 1:5 1 0.17
```

will result in an error message stating that some of the grid blocks have not been assigned values. This is because the data from the \*POR \*CON assignment is discarded when \*POR \*IJK is encountered. The usage of \*CON in itself is correct, but \*POR \*IJK must cover the entire grid.

In the recurrent data section, \*IJK may refer to select grid blocks. To change the relative permeability rock type in the 5x5 region to #4, use

```
*KRTYPE *IJK 1:5 1:5 1 4
```

---

## **Array Input of Values that Vary in the I Direction**

**\*IVAR**

### **PURPOSE:**

\*IVAR is used to indicate values that vary in the I direction, but which are constant in the other two directions.

### **KEYWORD:**

\*IVAR value(1) ... value(ni)

### **DEFINITIONS:**

value(1)

Value assigned to all grid blocks with an I direction index of 1.

ni

Number of grid blocks in the I direction.

### **EXPLANATION:**

Enter nj values separated by spaces or commas.

Example: I direction block sizes where ni = 10:

```
*DI *IVAR  
2*1000 1100 1050 3*800 860 1010 1100
```

Note that the structure '2\*1000' indicates the value '1000' occurs twice.

Example: I direction block sizes where ni = 3:

```
*DI *IVAR 3000.0 4000.0 5000.0
```

---

## **Array Input of Values that Vary in the J Direction**

**\*JVAR**

### **PURPOSE:**

\*JVAR is used to indicate values that vary in the J direction, but which are constant in the other two directions.

### **KEYWORD:**

\*JVAR value(1) ... value(nj)

### **DEFINITIONS:**

value(1)

Value assigned to all grid blocks with a J direction index of 1.

nj

Number of grid blocks in the J direction.

### **EXPLANATION:**

Enter nj values separated by spaces or commas.

Example: The J direction increments for a problem where nj=10 are: 755, 755, 755, 825, 825, 1000, 1000, 1100, 800, 800.

\*DJ \*JVAR 3\*755 2\*825 2\*1000 1100 2\*800

Example: The J direction has just 3 blocks:

\*DJ \*JVAR 3000.0 4000.0 3000.0

## **Array Input of Values That Vary in the K Direction**

**\*KVAR**

### **PURPOSE:**

\*KVAR is used to indicate values that vary in the K direction, but which are constant in the other two directions.

### **KEYWORD:**

\*KVAR value(1) ... value(nk)

### **DEFINITIONS:**

value(1)

Value assigned to all grid blocks with a K direction index of 1.

nk

Number of grid blocks in the K direction.

### **EXPLANATION:**

Enter nk values separated by spaces or commas. This is convenient for entering properties vary only by layer.

Example: Porosity varies for each of the layers of a system where nk=5, but is constant within each layer. The layer porosities are: .0810, .210, .180, .157, and .200.

\*POR \*KVAR .081 .21 .18 .157 .2

Example:

```
** Each of the I, J, and K permeabilities
** are constant within each layer of the
** reservoir but vary from layer to layer.
** Hence use *KVAR to input them layer
** by layer.
```

```
*PERMI *KVAR 200.0 50.0 500.0
*PERMJ *KVAR 200.0 50.0 500.0
*PERMK *KVAR 20.0 40.0 60.0
```

---

## **Values that Vary for Most or All Grid Blocks**

**\*ALL**

### **PURPOSE:**

\*ALL is used to indicate that values vary in most or all the grid blocks. The number of values expected is the number of grid blocks in the grid, including all null or zero-porosity blocks.

### **KEYWORD:**

\*ALL value(1) ... value(ni\*nj\*nk)

### **EXPLANATION:**

Values are entered starting with block (1,1,1) and in increasing block order where the I direction block index increases fastest and then the J direction block index second fastest and the K direction block index the slowest.

Example: Porosities for each grid block in a three-dimensional system vary in almost every grid block: ni=10, nj=3, nk=2

```
*POR *ALL
.08  .08  .081  .09  .12  .15  .09  .097  .087  .011
.15  .134  .08  .087  .157  .145  .12  .135  .18  .092
.074  .12  .12  .154  .167  .187  .121  .122  .08  .08
.095  .13  .12  .157  .17  .18  .184  .122  .084  .09
.11  .12  .134  .157  .157  .18  .18  .098  .09  .09
.08  .09  .144  .143  .123  .16  .165  .102  .10  .10
```

---

## Values Stored in Binary Form

\*BINARY\_DATA

### PURPOSE:

Builder uses this keyword to indicate that grid-array data is stored in binary form.

### KEYWORD:

\*BINARY\_DATA

### EXPLANATION:

#### In Builder

Normally Builder writes data in text-format files. However, Builder is able to write some grid definition and property data in binary form to a separate binary-format file. This option is invoked in Builder via menu "File/Save As.../Array Saving Method/Binary File Format (\*.cmgbin)". The binary file is saved in the same folder as the main 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

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.

## Data File Pathname

The path to the binary data file is specified via \*FILENAME \*BINDATA-IN or its default. Specifying a non-default path allows different main data files to refer to the same binary data file. In all cases a line in the log/diary reports the path name used to open the file.

Note that specification of the pathname immediately after \*BINARY\_DATA (as supported in versions 2008-2009) is obsolete.

## Examples

These are examples of data fragments you might see in the text data file written by Builder in binary file format.

```
** Optionally specified path name to binary data file
*FILENAME *BINDATA-IN 'My_Study.cmgbin'

** 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
```

## 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 ([ \* | - | + | / ] value)

### CONDITIONS:

When \*RG *uba\_range* is used together with \*EQUALSI, *uba\_range* must be only a single UBA (e.g., 1 2 3), not a UBA range (e.g., 1:3 2 3).

### EXPLANATION:

\*EQUALSI is used with direction-dependent keywords, such as the transmissibility, permeability and dispersion coefficients. This keyword works with \*MATRIX and \*FRACTURE separately.

**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
*PERMJ *MATRIX *EQUALSI
*PERMK *MATRIX *EQUALSI * 2.
*PERMI *FRACTURE *CON 10000
*PERMJ *FRACTURE *EQUALSI
*PERMK *FRACTURE *EQUALSI * 2.
```

**Example:** Assume the first example has a 3x2x1 refinement in fundamental blocks (1,1,1) and (1,1,2), whose permeability values are non-uniform but follow the same direction trend. Note that UBA range is not allowed with \*EQUALSI, so use single UBAs.

```
*PERMI *RG 1 1 1 :2 *ALL 120 150 180 135 165 195
*PERMJ *RG 1 1 1 *EQUALSI
*PERMK *RG 1 1 1 *EQUALSI * 2.
*PERMJ *RG 1 1 2 *EQUALSI
*PERMK *RG 1 1 2 *EQUALSI * 2.
```

---

## Modifying Array Data (Conditional)

\*MOD

### PURPOSE:

\*MOD indicates the modification of an input grid property.

### FORMAT:

\*MOD

{ *i1(:i2) j1(:j2) k1(:k2) opr value* }

-or-

\*MOD *opr value*

where

*opr* is one of +, -, \*, /, =

### DEFINITIONS:

*i1(:i2) j1(:j2) k1(:k2)*

I-J-K index ranges of the region to modify. The second index of a range must not be less than the first index; for example, “2:1” is not allowed. If the second index of a range is the same as the first index, the colon and second index may be omitted; for example, “5” is equivalent to “5:5”.

+

Indicates that *value* is added to the existing grid property value in the region defined.

-

Indicates that *value* is subtracted from the existing grid property value in the region defined.

\*

Indicates that the existing grid property is multiplied by *value* in the region defined.

/

Indicates that the existing grid property is divided by *value* in the region defined.

=

Indicates that the existing grid property is overwritten by *value* in the region defined.

## CONDITIONS:

The \*MOD keyword must appear immediately after the array property data, and may appear at most once.

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.

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.

Example: Suppose for a 10 x 6 x 1 grid you want to modify the porosities in the region with I indices 1 through 3, J indices 1 through 4 and with K index of 1 by adding 0.01. You further wish to assign the value of .13 to the block with I=5, J=2, and K=1. Enter \*MOD after the array values. The data looks like this:

```
*POR *ALL
.08 .08 .081 .09 .12 .15 .09 .097 .087 .011
.15 .134 .08 .087 .157 .145 .12 .135 .18 .092
.074 .12 .12 .154 .167 .187 .121 .122 .08 .08
.095 .13 .12 .157 .17 .18 .184 .122 .084 .09
.11 .12 .134 .157 .157 .18 .18 .098 .09 .09
.08 .09 .144 .143 .123 .16 .165 .102 .10 .10
*MOD
 1:3 1:4 1 + .01
 5   2   1 = .13
```

Example: 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 are not permitted to repeat a required array keyword after it has been entered once.

For example, the user wants to change some of the porosities to 0.22 after initially assigning 0.30 to all grid blocks. The following data entry is incorrect.

```
*POR *CON 0.3  
*POR *IJK 5:8 14:23 4 0.22 ** Incorrect
```

The correct procedure is to use the \*MOD keyword on the line immediately following \*POR:

```
*POR *CON 0.3  
*MOD 5:8 14:23 4 = 0.22
```

Note that if \*EQUALSI and \*MOD appear together, then \*EQUALSI is processed first and then the \*MOD values are processed.

The \*MOD keyword may be used together with \*RG for a single UBA. If fundamental block (3,7,5) has a 2x2x1 locally refined grid, then the following is valid.

```
*POR *RG 3 7 5 *ALL 0.30 0.39 0.29 0.26  
*MOD * 1.08
```

---

## Interpolating Table Data (Optional)

\*INT

### PURPOSE:

\*INT indicates that the corresponding table entry should be filled by interpolation.

### EXPLANATION:

The \*INT keyword may be used in table input. This keyword enables the calculation of the table entry by interpolation. Essentially the table entry corresponding to \*INT is replaced by a linearly interpolated value. This option is useful when not all table entries are known. This feature is explained in further detail with the help of an example.

Suppose that it is required to enter a water-oil relative permeability table into the simulator. Also assume that the water and oil relative-permeabilities are known at different saturations

*SWT		
**Sw	Krw	Krow
0.2	0.0	1.0
0.3	0.05	*INT
0.4	*INT	0.7
0.5	0.40	*INT
0.6	*INT	0.5
0.7	0.8	*INT
0.8	1.0	0.0
1.0	1.0	0.0

In the above table values denoted by \*INT will be calculated by linear interpolation by the simulator.

### NOTE:

Interpolation is done with respect to the first column. Thus the \*INT keyword cannot appear in the first column of the table.

At least one non \*INT entry must appear in a column. If only one non \*INT entry appears in the column then the entire column is assigned the same value.

# Input/Output Control

## Summary of Input/Output Control

Define parameters that control the simulator's input and output activities such as file names, units, titles, choices and frequency of writing to both the output and SR2 file, and restart control.

### List of Options

Names and paths of various input and output files:

- only required name is input data file
- restart run requires only one input restart file name
- complete set of consistent file-name defaults
- file names specified or defaulted independently
- defaults depend on root names of data and output file names
- bootstrapping of restart runs is made easy

Input/Output units have the following options:

- units sets available are SI, Field and Lab
- in addition to unit sets, individual units may be changed
- output units may be different from input units
- refer to mass instead of moles in mole-based quantities

The output file has the following writing options:

- well, grid and numerical performance are available
- frequency and amount of each are variable
- very long list of quantities available for entire grid
- control over orientation of grid printout appearance
- some quantities available with special units (ppm, pH, etc.)

The SR2 file has the following writing options:

- well, grid and special histories are available
- frequency and amount of each are variable

- very long list of quantities available for entire grid
- some quantities available with special units (ppm, pH, etc.)
- the binary file may be written in native or XDR format
- the binary file may be written in single precision

The restart facility has the following options:

- frequency of writing
- frequency of rewinding
- reading specified or last timestep

The interrupt handling facility has the following options:

- terminate run immediately after flushing and closing files
- terminate run after finishing current timestep and writing restart record
- prompt user interactively for instructions

### **Required Data**

There are no required or mandatory keywords in this section. Each keyword has a default value which can be used.

### **Critical Keyword Ordering**

\*FILENAME, if present, must be the first keyword to appear.

\*MASSBASIS and \*PARTCLSIZE, if present, must appear before \*OUTPRN and \*OUTSRF.

### **Usage in Other Sections**

Some of the keywords in this section may be used also in the Well and Recurrent Data section:

<b>May Appear in Recurrent Data</b>	<b>May Not Appear in Recurrent Data</b>	
*MAXERROR	*TITLE1	*RESTART
*SRFASCII	*TITLE2	*MASSBASIS
*PRINT_REF	*TITLE3	*PARTCLSIZE
*WRST	*CASEID	*OUTSRF *WELL
*REWIND	*CHECKONLY	*OUTSRF *SPECIAL
*OUTSOLVR	*INUNIT	*XDR
*OUTPRN	*OUTUNIT	
*WPRN	*PRNTORIEN	
*OUTSRF *GRID	*DIM	
*WSRF		
*DYNGRDFREQ		

### **Static Dimensioning Limits**

The following quantities have dimension limits that are static and so cannot be changed by the user.

<b>Value</b>	<b>Description</b>
100	Number of rock-fluid table entries
10	Number of IFT temperature entries
20	Number of IFT isotherm entries
10	Number of rock-fluid endpoint temperature entries
30	Number of reaction frequency factor table entries
30	Number of blockage resistance table entries
40	Number of temperature-viscosity table entries

### Run-Time Dimensioning

The amount of memory needed just to start STARS is less than 15 Mb, most of which is the executable file itself. However, there is no internal limit to the total amount of storage STARS will attempt to allocate as directed by the user's data. Therefore, the user has great flexibility in running larger data sets but needs to be aware of the corresponding storage requirement. The bulk of information required to allocate sufficient internal storage is obtained from a preliminary scan of the data. The few remaining dimension parameters, listed in the table **Static Dimensioning Limits**, above, are absolute maximums compiled into the executable.

In the following, '>' at the beginning of a line indicates an output line to the screen or diary file if redirected.

Normally the beginning of the screen or diary output looks like

```
> Banner . . .
>
> Opened data file on unit 72, filename is 'correl.dat'
>
> Scanning data for dimensioning info . . .
> Done.
>
> Opened output file      on unit 73, filename is 'correl.out'
> Opened INDEX-OUT        on unit 74, filename is 'correl.irf'
> Opened MAIN-RESULTS-OUT on unit 76, filename is 'correl.mrf'
>
> ====== SUMMARY (from subroutine: INDATA) ======
> Reading of initial data is complete.
> Simulation will stop if there were error messages.
> 3 Warning messages.      0 Error messages.
> ======
```

indicating that the data file is opened and then scanned for values of dimensioning parameters such as number of blocks, components and wells. Then the output files are opened and the initialization (non-recurrent) data is read, processed and echoed.

When \*DIM \*DIMSUM or command line argument '-dimsum' is used the following type of scan report is printed:

```

> Summary of Dimensions Obtained from Data Scan
>
>      2 NUMY      - Number of fluid components
>      2 NUMX      - Number of condensable components
>      2 NW        - Number of wells
>      34 MDPTGL   - Number of unique completions
>      1 MFORM     - *TFORM flag: 1 for *SXY, 2 for *ZH, 3 for *ZT
>      1 MISOTH    - *ISOTHERMAL flag: 1 for thermal, 2 for isothermal
>      1 NPTGN    - Number of grids
> 1190 NPTSS   - Number of matrix blocks
> 1190 NPTCS   - Number of blocks including nulls
>      1 M9PT      - *NINEPOINT flag: 1 - no, 2 - yes
>      3 NDIM      - Number of dimensions (= 3 for *REFINE)
>      0 NREF      - Number of refinements per fundamental block
>      0 MINC      - Number of *MINC or *SUBDOMAIN subdivisions
>      8 NORTH     - Number of orthogonalizations
>      0 NDWGL     - Number of discretized wellbore blocks from *WELLBORE
> 4320 NCLU     - Number of LU connections
>      0 NGAUSS    - Bandwidth for *SDEGREE *GAUSS

```

This report shows what dimensioning information was obtained from the preliminary scan of the data file. For this particular data, there are 2 components, 1190 blocks, 2 wells and 34 global well completion layers.

Accompanying the above scan report are two other reports: a detailed summary of storage used by each module, and a complete list of dimensioning parameters.

```

> Summary of Storage Required
>
> Storage used by STARS      2175170
> Storage used by WELLGRP    214
> . .
> Storage used by AIMSOL    1160524
> Storage used by Total =   7331035

> Dimensioning Parameters
>
> 1190 MDPTCS - Total blocks, including nulls
> 1190 MDPTPS - Total non-null blocks
> . .
>      2 NUMY      - Fluid components
> . .
>      2 MDWELL    - Source/sink wells
>      34 MDPTGL   - Global well layers
> . .
> 4320 MDICLU   - Block entries in each of L & U
> 38880 MDLU    - Size of each of L & U

```

These last two reports appear also when an allocation error occurs. The main cause of such an error is an attempt to allocate more memory than is available. STARS will allocate storage until the first failure, print the two reports and stop.

Test data "verify25.dat" in the "verify" directory of the STARS template area is designed to test handling of allocation errors. On a machine with 480 Mb of process space, it gives:

```
> ERROR: Memory allocation failure for array: t1, 38901600 bytes
>
> The following summaries will help you find the reason for
> the allocation error. The most common reason is that this
> data requires more swap space (virtual memory) than is
> available on this computer at this time. To get a summary
> of dimension parameters generated by your data use keywords
> *DIM *DIMSUM in the I/O Control section or command-line
> argument "-dimsum".
```

followed by the two reports, the last indicating a total of 449043917 bytes or about 450 Mb. The allocation of array 't1' at 39 Mb would have put the total at 489 Mb, exceeding the available process space. Increasing the process space (up to the 2 Gb 32-bit limit) usually solves this problem, but use of process space significantly larger than the physical memory (RAM) will result in paging that may degrade performance (especially for PC's).

For most data sets the dimensions obtained by scanning is sufficient. However, it is possible that several dimensioning parameters may be insufficient, in which case the user may enter values directly via \*DIM subkeywords.

See "Optimizing Memory Requirements" in the Tutorial section of this User Guide.

### **32-bit and 64-bit Process Space Limits**

A computer based on a 32-bit architecture has a hard limit in how much space can be addressed (and hence allocated) in a single process such as a STARS run. For most such operating systems this limit is 2 Gb. However, the limit is 3 Gb for the following Windows operating systems:

- Windows .NET server family
- Windows XP Professional Edition
- Windows 2000 Datacenter Server
- Windows 2000 Advanced Server

To access this option add switch /3GB to boot.ini on the target machine. The STARS Win32 executable is already enabled for 3 Gb.

A 64-bit machine will have a process space limit orders of magnitude larger than 3 Gb. However, the limiting factor for performance will be still the amount of physical memory, especially for Win64 machines.

---

## Command-Line Arguments (Optional)

### PURPOSE:

Specify some run information via command line.

### FORMAT:

```
stars.exe      (-f input_data)
              (-log (log_path) )
              (-r input_restart)
              (-restart (nstart) )
              (-restime restime)
              (-stoptime stoptime)
              (-checkonly)
              (-dimsum)
              (-onestep)
              (-maxsteps nstop)
              (-wd path | -dd )
              (-wait)
              (-doms (ipldom) )
              (-parasol (n) )
              (-aimsol)
              (-file_no_replace )
```

### DEFINITIONS:

*stars.exe*

STARS invocation command, usually the name of an executable file. It can be a local file, a link to a file or merely accessible via search rules.

*-f input\_data*

Specifies that *input\_data* is the path name to a STARS main data file.

*-log ( log\_path )*

Specifies that consol “diary” output will be redirected to a file. If *log\_path* is present, this file has pathname *log\_path* and extension “.log” will be added if it is not already present. If *log\_path* 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.

*-r input\_restart*

Specifies that *input\_restart* is the path name to a STARS input restart IRF generated by a previous STARS run. The MRF and possibly RRF files required for restart also will be obtained from similar pathnames. This option overrides pathnames specified by subkeywords \*INDEX-IN, \*MAIN-RESULTS-IN, and \*REWIND-IN of keyword \*FILENAME that may occur in the data.

- restart ( *nstart* )  
Equivalent to putting \*RESTART in your data, with or without *nstart*. See manual entry for \*RESTART. This command-line argument overrides \*RESTART data in the file. If both -restart and -restime appear in the command line, -restart is ignored.
- restime *restime*  
Equivalent to putting \*RESTIME *restime* in your data. See manual entry for \*RESTIME. This command-line argument overrides \*RESTIME data in the file. If both -restart and -restime appear in the command line, -restart is ignored.
- stoptime *stoptime*  
Stops the simulation at *stoptime* (days | days | mins) which must correspond to a simulation reference time specified via \*TIME or \*DATE in the recurrent data section before the first \*STOP keyword.
- checkonly  
Equivalent to putting \*CHECKONLY in your data. See the manual entry for \*CHECKONLY.
- dimsum  
Equivalent to putting \*DIM \*DIMSUM in your data.
- onestep  
Equivalent to putting \*MAXSTEPS 1 in your data.
- maxsteps *nstop*  
Equivalent to putting \*MAXSTEPS *nstop* in your data.
- wd *path*  
Output files will be written to the directory given by *path*. This option is useful in an environment where the “current directory” may not be defined.
- 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.
- wait  
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.) This is useful when several jobs are submitted via the CMG Technology Launcher at one time (e.g., over the night or weekend) and the number of licenses is limited. An alternate way to run a series of jobs sequentially is to use a batch file. See **Running Your Simulation** in the Tutorial chapter.

- doms ( *ipldom* )**  
Enables parallel processing for Jacobian building. Optional *ipldom* specifies the target number of planes per Jacobian domain (default 4). This argument overrides all data specified via keywords \*DPLANES and \*DTYPE.
- parasol ( *n* )**  
Enables parallel processing for matrix solution via PARASOL. Optional *n* specifies the number of threads to use (default 2). See keyword \*SOLVER.
- aimsol**  
Enables AIMSOL. See keyword \*SOLVER.
- file\_no\_replace**  
Prevents overwriting of all output files, including the log/diary file if "-log" is used. With this argument, an attempt to overwrite an existing output file will result in a fatal error message issued to the log file (if "-log" is used) or the default output device (screen). If you wish this prevention in place for all Launcher runs of STARS, enter this argument in the Launcher's STARS icon, in the field labeled "Additional command line switches for the executable". Note that the message will not appear if you attempt to overwrite the log/diary file using Launcher in screen-less (script) mode.

## DEFUALTS:

If an input data file name is not supplied here via argument "-f", then STARS will prompt for it. If "-log" is absent, the prompted data is read from the default input device (standard\_in, keyboard). If "-log" is present then the prompted data is read from a file whose name is (a) the same as *log\_path* but with extension ".in", or (b) "cmg.in" if *log\_path* is absent.

If "-f" is absent but "-log" is present without *log\_path*, the "diary" output goes to a file named "cmg.log".

If "-log" is absent, "diary" output will go to the default display (standard\_out, screen).

If this is a restart run and the input restart file name is not supplied here via argument "-r" or via keywords \*FILENAME \*INDEX-IN, then STARS will prompt for it.

If neither -wd nor -dd is supplied, then output file names are obtained from the \*FILENAME keyword. If \*FILENAME is absent, then the output files are written to the current working directory.

If "-wait" is absent and all available licenses are being used, the run stops.

If "-file\_no\_replace" is absent, an existing file is replaced with a new file of the same name generated by the current run.

---

## **Input/Output File Names (Optional)**

**\*FILENAME**

### **PURPOSE:**

Specify names for input and output files. Needed only to override default file names, or specify input restart when not using command-line argument.

The main data file must be specified via the standard input device (keyboard/job-running script) or the command-line argument "-f".

### **FORMAT:**

**\*FILENAME {*file\_type* (*name\_option*) }**  
where *file\_type* is one of:

- \*OUTPUT
- \*INDEX-OUT
- \*MAIN-RESULTS-OUT
- \*REWIND-OUT
- \*INDEX-IN
- \*MAIN-RESULTS-IN
- \*REWIND-IN
- \*BINDATA-IN
- \*GEOMECHOUT

and *name\_option* is one of:  
"  
'*filename*'  
\*PROMPT

For *file\_type* \*OUTPUT, the additional *name\_option* \*SCREEN is allowed.

### **DEFINITIONS:**

#### **\*FILENAME**

File name keyword.

#### **\*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.

When the restart run is started, some information is copied from the SR2 input files to the SR2 output files. Time-based histories are not copied but are accessed by parts from each set of SR2 files that lead up to the current run; do not delete the input SR2 files until the information they contain is no longer needed.

#### **\*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\_type* is necessary only for restart runs.

**\*MAIN-RESULTS-IN**

Indicates the main-results-file from which the simulation results and restart records (binary) are read. This *file\_type* is necessary only for restart runs.

**\*REWIND-IN**

Indicates the rewritable-results-file from which the rewound restart records (binary) are read. This *file\_type* is necessary only for restart runs.

**\*BINDATA-IN**

Indicates the binary data file used by the \*BINARY\_DATA array reading option.

**\*GEOMECHOUT**

Indicates the file to which the formatted output generated by the geomechanical model is written, if this option is chosen. If \*GEOMECHOUT is absent, or \*PROMPT is used, this output is written to the main output file given by \*OUTPUT.

"

Empty string, denoting that an internally generated default file name will be used.

**'filename'**

A character string which is the file name. Characters after the first 80 will be ignored. Acceptable file names depend on the operating system being used. If *filename* is a relative pathname, each directory delimiter that is invalid for the system will be changed internally to a valid delimiter. This allows the same main data file to be used on both system types.

**\*PROMPT**

Indicates that the user will be prompted for this file name via the standard input device (keyboard or job-running script), if the file is required. All file types except \*INDEX-IN have an internally generated default file name available. To use it, enter a null response at the prompt.

**\*SCREEN**

Indicates that data for this file type will go to the standard output device (screen/job diary file).

## **DEFAULTS:**

There is no default file name available for the main data file. It must be specified by the user, either via the standard input device (keyboard/job-running script) or the command-line argument "-f". CMG Launcher uses command-line argument "-f".

There is no default file name available for the input restart file \*INDEX-IN. It must be specified by the user, either via the standard input device (keyboard or job-running script), keyword \*FILENAME \*INDEX-IN (both appropriate when using the CMG Technology Launcher) or the command-line argument "-r".

If any other required file name is not specified via \*FILENAME (including \*PROMPT), an internally generated file name is used. See "Internally Generated Default File Names", below.

## **CONDITIONS:**

\*FILENAME keywords, if present, must occur before any other keyword. Any subsequent \*FILENAME keyword will be ignored.

Output files are not protected from overwriting unless (1) file writing permission is removed at the operating system level, or (2) command-line argument “-file\_no\_replace” is used.

Input files must be present for the simulation to proceed. Only read permission is required.

Command-line argument -r will override all restart input file names that have been specified via \*FILENAME.

## **EXPLANATION:**

### **CMG's Simulation Results File System (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). The graphics post-processor RESULTS and the Report Writer require the IRF and the MRF files.

These files are required also for restart runs. If the \*REWIND option was used to write restart records, then the RRF is required also for restart runs. Only restart information that was written to the RRF after the last rewinding are available.

### **Internally Generated Default File Names**

There is a consistent set of file names generated internally. Each file type's file name is one of the three file-name roots appended with a unique suffix. These file names are available as defaults for each file type individually. This consistency is very useful in doing series of restart runs in a manageable manner. There are three base file types from which the default file-name roots are derived: input data, output and input restart.

**Input Data:** This file name is entered via the prompt or the command line. The input data root name is this file name minus the suffix '.dat' if it exists, and contains the full path name to the input data file's directory. The default path name to another input data file type (presumably in the same directory as the input data file) is this root name with a unique suffix appended.

Output: The default file name for \*OUTPUT is the input data root name with the directory path stripped off (to make it "local") and '.out' appended on the end. Command-line argument –wd and –dd will override the directory portion of this default pathname. This file name or another specified via the \*FILENAME keyword is used to open the file. The output root name is the \*OUTPUT file name minus the suffix '.out' if it exists, and possibly contains the full path name to that output file's directory. The default path name to another output file type such as \*INDEX-OUT is this output root name with a unique suffix appended.

Input Restart: The file name for file type \*INDEX-IN may be entered via prompting, \*FILENAME or the command line; its file name must end with the suffix '.irf'. The input restart root name is this file name minus the suffix '.irf' if it exists, and contains the full path name to that file's directory. The default path name to another input restart file type such as \*MAIN-RESULTS-IN (presumably in the same directory as the input restart file) is this root name with a unique suffix appended.

The source of default file name for each file type is summarized here:

File Type	Root Based On	Suffix
*OUTPUT	input data	.out
*INDEX-OUT	*OUTPUT	.irf
*MAIN-RESULTS-OUT	*OUTPUT	.mrf
*REWIND-OUT	*OUTPUT	.rrf
*GEOMECHOUT	*OUTPUT	.geo
*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 \*INDEX-IN file name for each run.

### Binary Data File Pathname

Subkeyword \*BINDATA-IN causes the \*BINARY\_DATA option to use the file whose path is given by one of the following cases. **Note:** An absolute pathname starts with a directory delimiter; on Windows there may be a leading drive (letter and colon). Any other pathname is treated as relative.

1. *filename* or \*BINDATA-IN is **absent**: The path is assumed to have the same root name (including directory) as the main data file, but extension ".cmgbin".
2. *filename* is **absolute**: The path is *filename*.
3. *filename* is **relative**: *filename* is relative to the directory containing the main data file and is converted internally to the corresponding absolute or relative pathname.

---

## Dimension Over-Rides (Optional)

\*DIM

### PURPOSE:

Over-ride default dimension estimates based on the preliminary data scan.

### FORMAT:

```
*DIM    (*DIMSUM )
      (*MDPTGL mdptgl)
      (*MDICLU mdiclu)
      (*MDJCM mdjcm)
      (*MDCALP mdcalp)
      (*MDALP mdalp)
      (*MDV mdv)
      (*MDDD mddd)
      (*MDLU mdlu)
      (*MDPTCN mdptcn)
      (*MD-GM-DBINT mdgrig)
      (*MD-GM-DBREAL mdgrrg)
```

### DEFINITIONS:

#### \*DIMSUM

Enables detailed report of dimensioning parameters and storage requirements, written to the screen or diary file if redirected. This report can be enabled also with command-line argument '-dimsum'. See "Run-Time Dimensioning" at the beginning of this chapter.

#### *mdptgl*

Maximum number of global completion layers expected. Over-ride this quantity only if the automatic estimation process fails.

#### *mdiclu*

Maximum number of solver fill connections expected. Over-ride this quantity only if the automatic estimation process fails. See **Solver Matrix Fill** in EXPLANATION, below.

#### *mdjcm, mdcalp, mdalp, mdv, mddd, mdlu*

Matrix solver dimension parameters. Over-ride only if necessary. See **Other Matrix Solver Dimensions** in EXPLANATION, below.

#### *mdptcn*

Maximum number expected for the sum of interblock connections and well completion layers. Over-ride this quantity only if the automatic estimation process fails.

#### *mdgrig*

Dimension of Grid Module integer data base.

## *mdgrrg*

Dimension of Grid Module real data base.

### **DEFAULTS:**

If \*DIM \*DIMSUM is absent, and the command-line argument '-dimsum' is absent, the detailed report is not enabled.

Each of the other \*DIM subkeywords defaults independently to the value obtained from the data scan.

### **EXPLANATION:**

Run-time dimensioning in STARS is designed to obtain all its needed information for storage allocation from a preliminary scan of the data. However, it is possible that several dimensioning parameters may be insufficient after this scan, in which case the user may enter values directly via \*DIM subkeywords.

See also **Optimizing Memory Requirements** in the TUTORIAL chapter.

### **Solver Matrix Fill**

Dimensioning for the matrix solver arrays is complex, and has been automated to a large extent. However, two quantities may need manual over-rides under certain circumstances: \*MDICLU and \*MDLU which correspond to matrix "fill". These determine the sizes of the largest solver arrays, which together can make up over half of the total STARS storage requirement.

Normally, estimates for \*MDICLU and \*MDLU from the data scan are sufficient for default values of matrix solver controls \*SORDER, \*SDEGREE (1 and \*GAUSS) and \*MAXLAYPRE. When the estimates are not sufficient, STARS issues messages in the output (.out) file along with a brief message in the diary (screen or log file). From these messages the user obtains the required values for these quantities, and enters them via keywords \*DIM \*MDICLU. The keyword \*OUTSOLVR allows you to examine solver storage requirements at any time.

The internal estimate for \*MDICLU is obtained in stages, first for the grid (without wells) and then for the grid plus each set of active wells defined by each recurrent data segment. Since notification of insufficient \*MDICLU (or \*MDLU, for that matter) can occur in any of these places, the activation of large wells at later times can cause the run to stop part way through. A restart with increased \*MDICLU should work. This will be especially true for higher \*SDEGREE and \*MAXLAYPRE where wells induce significant matrix fill. Keyword \*OUTSOLVR is useful in finding what value of \*MDICLU is required (remember to add at least 1 to the value reported). It is advisable to use keyword \*CHECKONLY to detect insufficient dimensioning before a large run is submitted.

Note that for \*SDEGREE greater than 1 the error message does not indicate the required value of \*MDICLU. In this case, use \*DIM \*MDICLU to enter double the initial estimate, use \*OUTSOLVR to examine the actual requirement, and re-enter \*MDICLU with at least the required value.

### **Other Matrix Solver Dimensions**

The other matrix solver dimensions corresponding to \*MDJCM, \*MDCALP, \*MDALP, \*MDV, \*MDDD and \*MDLU normally are sufficient and can be defaulted. However, runs with large grids tend to be over-dimensioned, so these subkeywords can be used to minimize the storage allocated for a given grid definition. Use keyword \*OUTSOLVR \*ON to find the current values of these matrix solver dimensions.

---

## Scan Mode for Checking Errors (Optional)

\*CHECKONLY

### PURPOSE:

Enable scan mode for checking of entire data.

### FORMAT:

\*CHECKONLY

### DEFAULTS:

If this keyword is absent, timestep calculations are performed.

### EXPLANATION:

Normally (i.e., without \*CHECKONLY), syntax, storage allocation and range checking of data is done as it is read. Initialization data (all but recurrent) is processed at the beginning of the run, so errors in that part of the data are detected and reported immediately.

However, recurrent data is read when it is needed, as the simulation time progresses.

Therefore, errors in recurrent data will be detected and reported later in the run. This can be inconvenient for large runs. Keyword \*CHECKONLY allows you to scan your entire data set to the end very quickly so that data errors are detected and reported immediately.

In fact, the only part of the simulation not done in scan mode is the timestep calculation. This means that all the reading, storage allocation, checking, echoing, printing and SR2 dumping are done. For example, you can view the initial conditions, and hence your grid, in RESULTS from an SR2 generated by a scan mode run.

It is recommended that you keep near the top of your data set a line consisting of \*CHECKONLY. Normally this keyword is disabled (commented out). You can quickly enable the keyword and run the data in scan mode. Remember to disable the keyword before submitting the actual run.

A license is not required to run STARS in scan mode, allowing you to validate data while your licenses are occupied running simulations.

Command-line argument “-checkonly” performs the same function as \*CHECKONLY without the requirement of changing the data file.

---

## **Project Main Title (Optional)**

**\*TITLE1, \*TITLE2, \*TITLE3, \*CASEID**

### **PURPOSE:**

Identify the project and individual run cases with titles and comments.

### **FORMAT:**

<b>*TITLE1</b>	string
<b>*TITLE2</b>	string
<b>*TITLE3</b>	string
<b>*CASEID</b>	string

### **DEFINITIONS:**

#### **\*TITLE1**

Character string used for project identification, appearing in both printed output and in the SR2 file. Characters after the first 40 will be ignored.

#### **\*TITLE2**

Character string used for project identification, appearing in both printed output and in the SR2 file. Characters after the first 80 will be ignored.

#### **\*TITLE3**

Character string used for run identification, appearing in both printed output and in the SR2 file. Characters after the first 80 will be ignored.

#### **\*CASEID**

Character string used to identify specific cases, used also in the SR2 file to identify data curves for plots. Characters after the first 8 will be ignored.

### **DEFAULTS:**

The default for each keyword is a blank string.

### **CONDITIONS:**

This keyword must appear in the INPUT/OUTPUT CONTROL keyword group, at the start of the data file.

### **EXPLANATION:**

Examples:

```
*TITLE1 'DUAL POROSITY/DUAL PERMEABILITY RUN NO. 1'  
*TITLE2 'Run by A.B. staff, Dec. 16, 1988. C.D. Co.'  
*TITLE3 '4200 grid blocks; var. thickness'  
*CASEID 'No Gas'
```

---

## **Input/Output Data Units (Optional)**

**\*INUNIT, \*OUTUNIT**

### **PURPOSE:**

\*INUNIT specifies the input data units.  
\*OUTUNIT specifies the output data units.

### **FORMAT:**

\*INUNIT ( \*SI | \*FIELD | \*LAB )  
  { \*EXCEPT qnty\_no unit\_no }

\*OUTUNIT ( \*SI | \*FIELD | \*LAB )  
  { \*EXCEPT qnty\_no unit\_no }

### **DEFINITIONS:**

#### **\*INUNIT**

Indicates that the following unit identifiers are for input data units.

#### **\*OUTUNIT**

Indicates that the following unit identifiers are for output data units.

#### **\*SI**

This option specifies the SI unit system (see UNITS TABLE, below).

#### **\*FIELD**

This option specifies the FIELD unit system (see UNITS TABLE, below).

#### **\*LAB**

This option specifies the LAB unit system (see UNITS TABLE, below).

#### **\*EXCEPT**

This option allows alternate input units for selected quantities.

#### **qnty\_no**

Quantity number from the list below.

#### **unit\_no**

Unit number from the list below.

### **DEFAULTS:**

If \*INUNIT is absent, then \*INUNIT \*SI is assumed. If \*OUTUNIT is absent, the output units will be the same as the input units.

## **EXPLANATION:**

Each dimensioned quantity in this manual appears with at least two unit labels:

- the first set in the \*SI system,
- the second set in the \*FIELD system,
- and the third set in the \*LAB system (if different from \*SI).

For example, the unit of mass density is reported as

( kg/m<sup>3</sup> | lb/ft<sup>3</sup> | kg/cm<sup>3</sup> )

where

kg/m <sup>3</sup>	corresponds to *SI,
lb/ft <sup>3</sup>	corresponds to *FIELD, and
kg/cm <sup>3</sup>	corresponds to *LAB.

The unit actually used is determined by the choice of \*SI, \*FIELD or \*LAB after \*INUNIT. Defaults may appear in the text of this manual in \*SI unit only, but they will be converted to and echoed in your chosen units at run time.

In addition to the three unit systems, selected quantities can be given units different from those implied by the keywords \*SI, \*FIELD, or \*LAB through use of the \*EXCEPT keyword. For example, to use degrees F instead of C with the \*SI system, put

```
*INUNIT *SI *EXCEPT 2 2 ** use F, not C
```

Once the unit set is specified via \*INUNIT, including exceptions to the unit system via \*EXCEPT, that set must be used consistently throughout the data. There is no facility to enter data in one unit system in one part of the data, and another unit system in another part of the data.

In contrast, \*OUTUNIT can be changed freely from one run to the next of the same data, since it affects only the output and not the input data. In any one run, the output unit set chosen will be applied consistently throughout the output.

Information stored in the SR2 files are in STARS internal units, and so are independent of the output units chosen. However, the Index Result File (IRF) records the chosen output units for the run, and uses these as the default output units when post -processing (graphing or report generating) is done.

Table 7 gives some selected unit conversion factors.

## UNITS TABLE

<b>QUANTITY</b>	<b>*SI</b>	<b>*FIELD</b>	<b>*LAB</b>
Time	days	days	minutes
Temperature	deg C	deg F	deg C
Pressure	kPa	psi	kPa
Length	m	ft	cm
Volume	m <sup>3</sup>	ft <sup>3</sup>	cm <sup>3</sup>
Permeability	md	md	md
Mass	kg	lb	kg
Molar Mass (mass basis)	gmole (kg)	lbmole (lb)	gmole (kg)
Viscosity	cp	cp	cp
Energy	Joules	Btu	Joules
Well Liquid Volume	m <sup>3</sup>	ft <sup>3</sup>	cm <sup>3</sup>
Well Gas Volume	m <sup>3</sup>	ft <sup>3</sup>	cm <sup>3</sup>
Interfacial Tension	dyne/cm	dyne/cm	dyne/cm
Electrical Potential	V (volts)	V	V
Electrical Current	A (amperes)	A	A
Electrical Power	kW (10 <sup>3</sup> Watts)	kW	kW
Electrical Conductivity	siemens/m	siemens/m	siemens/m

## ALTERNATE UNIT CHOICES

<b>QUANTITY</b>	<b>qnty_no</b>	<b>unit no</b>				
		<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
Time	1	days	hr	min	yr	
Temperature	2	deg K	deg C	deg F	deg R	
Pressure	3	kPa	psi	atm	bar	kg/cm <sup>2</sup>
Length	4	m	ft	cm		
Volume	5	m <sup>3</sup>	ft <sup>3</sup>	bbl	cm <sup>3</sup>	
Permeability	6		darcy	micro-m <sup>2</sup>	md	
Mass	7	kg	lb			
Molar Mass	8	gmol	lbmol			
Viscosity	9	kPa-day	kPa-hr	cp		
Energy	10	J	BTU			
Well Liquid Volume	11	m <sup>3</sup>	ft <sup>3</sup>	bbl	cm <sup>3</sup>	
Interfacial Tension	12	kPa-m	N/m	dyne/cm		
Well Gas Volume	13	m <sup>3</sup>	ft <sup>3</sup>	bbl	cm <sup>3</sup>	

---

## **Mass Basis Indicator (Optional)**

**\*MASSBASIS**

### **PURPOSE:**

\*MASSBASIS enables the mass basis option.

### **FORMAT:**

\*MASSBASIS

### **DEFINITIONS:**

\*MASSBASIS

Component property data is based on mass, that is, each instance of unit “Molar mass” is interpreted as mass (kg or lb).

### **DEFAULTS:**

If keyword \*MASSBASIS is absent then the component property data is based on moles, that is, each instance of unit “Molar mass” is interpreted as moles (gmole or lbmole).

### **CONDITIONS:**

The \*MASSBASIS option should not be used when any component vapourizes, since the vapour/liquid K value needs to use the mole fraction definition. This is especially true of steam processes. To find how to disable the default vapourization of water components, see the DEFAULTS section of manual page “K Value Correlations” (keywords \*KV1, etc.).

Viscosity option \*GVISCOR cannot be used together with \*MASSBASIS.

### **EXPLANATION:**

In some chemical flood processes it is desirable to work with composition in mass fraction instead of mole fraction (the default). For example, when the component set includes a polymer of very large molecular weight, the corresponding mole fraction is very small, and mixing rules based on mole fraction weighting may no longer be appropriate.

Keyword \*MASSBASIS causes almost every instance of moles for data entry to be interpreted as mass. Component properties are on a per mass basis, and K values are defined as the ratio of phase mass fractions instead of mole fractions. Phase compositions are reported in mass fraction. Reaction stoichiometric coefficients are based on mass.

The Manual entry for each component contains a generic definition of its unit. For example, the density unit is (molar mass/volume); molar mass is interpreted according to the mole/mass basis. In SI units this density will have unit gmol/m<sup>3</sup> normally, but has unit kg/m<sup>3</sup> under \*MASSBASIS.

One exception is the definition of molecular weight, which must retain the unit (mass/mole), i.e., (kg/gmole) in SI units and (lb/lbmole) in field units. The other exception is reaction activation energy obtained via \*EACT or \*EACT\_TAB, which retains its per-mole unit.

If the mass basis option is used, keyword \*MASSBASIS must appear before \*OUTPRN and \*OUTSRF since it affects the default unit of concentration and composition output quantities.

---

## **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

The maximum number of error messages allowed. The allowed range for num is 1 to 100.

### **DEFAULTS:**

\*MAXERROR 20

### **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.

Certain types of syntax error will cause the keyword processor to issue many error messages even though there is only one error. When in doubt, correct errors starting from the top and work your way down; you may find that fixing one error removes many error messages.

---

## Starting Timestep or Time

\*RESTART, \*RESTIME

### PURPOSE:

Specify the starting timestep or time.

### FORMAT:

\*RESTART    (*nstart*)  
\*RESTIME    *restime*

### DEFINITIONS:

#### \*RESTART (*nstart*)

Specify number of the timestep from which to restart the simulation.  
Command-line argument ‘-restart’ is another way to specify \*RESTART.

#### \*RESTIME *restime*

Specify time (days | days | mins) of the timestep from which to restart the simulation. This option works best when *restime* corresponds to a simulation reference time specified via \*TIME or \*DATE in the recurrent data section of the previous run. Also, *restime* may be a non-reference time but it must match the time of the target restart record within the first 7 decimal digits. Use \*RESTART when \*RESTIME picks incorrectly from a group of records whose times do not differ in the first 7 digits. Command-line argument ‘-restime’ is another way to specify \*RESTIME.

### DEFAULTS:

If \*RESTART and \*RESTIME are absent, no restart records are read and the first timestep number is 1.

If \*RESTART is present without *nstart*, the last restart record in \*INDEX-IN is used.

### CONDITIONS:

If \*RESTART or \*RESTIME is present, then the restart files denoted by \*INDEX-IN and \*MAIN-RESULTS-IN (and possibly \*REWIND-IN) are required and must contain the restart record corresponding to the specified time or timestep number.

If both \*RESTART and \*RESTIME are present, or each appears multiple times, only the last occurrence is used. For example, if \*RESTART 10 appears before \*RESTIME 50.5 then the restart will come from the timestep at 50.5 days.

### EXPLANATION:

See **How To Do a Restart** in the Tutorial section.

### Version Backward Compatibility

Restarts are generally backward compatible starting with version 2008.10. For example, version 2010.10 can read restarts written by versions 2008.10 and 2009.11. Restart compatibility requires that the restart run data be compatible with the original run data and the newer version. Due to changes and improvements, the newer version may not give exactly the same numerical performance or engineering result as the older version.

---

## Restart Record Writing (Optional)

\*WRST, \*REWIND

### PURPOSE:

\*WRST and \*REWIND control the frequency of writing and rewinding restart records in the output restart file.

### FORMAT:

\*WRST (*freq* | \*TIME | \*TNEXT )  
\*REWIND (*num*)

### DEFINITIONS:

*freq*

A restart record is written at each timestep number evenly divisible by *freq*, as well as each subsequent \*DATE or \*TIME in recurrent data. If *freq* is zero a restart record is not written.

\*TIME

A restart record is written at each subsequent \*TIME or \*DATE in recurrent data. This is equivalent to specifying a large *freq*.

\*TNEXT

A restart record is written only for the next recurrent data time after which writing is disabled, resulting in one restart record per keyword occurrence. This option is useful when writing restarts at infrequent but known times in recurrent data.

*num*

The maximum number of restart records allowed to accumulate on the restart file before it is rewound. If *num* = 0 then no rewinding is done. If *num* = 1, then only the last written restart record will be available.

### DEFAULTS:

If \*WRST is not present, no restart is written. If \*WRST is not followed by *freq*, \*TIME or \*TNEXT, then \*WRST \*TIME is assumed.

If \*REWIND is not present the restart file is never rewound. If \*REWIND is present but *num* is absent, then *num* = 1 is assumed.

### CONDITIONS:

\*WRST may appear also in recurrent data to vary the frequency or time of restart record writing with time.

### 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:

```
*WRST 10 ** Write restart record every 10 timesteps.  
*WRST      ** Write restart at the every time change.  
*WRST *TNEXT ** Write restart at next *TIME/*DATE only.  
*REWIND 3 ** Rewind restart file every 3 restarts.
```

Examples of entire data sets using the restart option can be found in directory "restart" in the STARS template release area.

### A Quick Restart Check

To find out quickly at what timesteps a restart was written in a simulation, use your text editor to look in the SR2 index file which has file-name suffix ".irf". A timestep with a restart written produced the following lines in the IRF generated by the test data "rrfa.dat" in the template directory "restart":

```
TIME 21      10.0000000000 19731005  
TIMCHR '      10.00000 days' ' 5 Oct 1973'  
FILE 2  
REWIND 2  
RESTART-CONTROL ( 3 ) 8 1 2 0 1  
RESTART ( 34 ) IFLGGN . . .  
FILE 1  
WELL ( 2 ) 1 2  
GROUP ( 2 ) 1 2  
SPEC-HISTORY ( 1 ) SPVALS /  
GRID-VALUE ( 3 ) PRES SG TEMP /
```

TIME indicates the timestep number and the simulation time and date. RESTART-CONTROL and RESTART must be present before a restart can be read from that timestep. FILE and REWIND are present only when the \*REWIND option was used. REWIND indicates that the restart file was rewound which means that all restarts up to that point are lost. Therefore, only the restart records after the last REWIND will be accessible.

## **Output Printing Frequency (Optional)**

**\*WPRN**

### **PURPOSE:**

\*WPRN controls the frequency of writing to the output print file information flagged by \*OUTPRN.

### **FORMAT:**

```
*WPRN (*GRID | *ITER) (freq | *TIME | *TNEXT )
*WPRN *SECTOR (freq | *TIME )
```

### **DEFINITIONS:**

#### **\*GRID**

Pertains to the conditions of the reservoir and fluids in it, as well as the detailed well performance report.

#### **\*ITER**

Pertains to the brief well rate report as well as simulator performance, e.g., material balance.

#### **\*SECTOR**

Pertains to statistics reported by sector. Sector statistics are written either not at all (*freq* = 0) or at the same times as \*GRID.

#### *freq*

Write indicated results to the output file at timestep numbers evenly divisible by non-negative integer *freq*. If *freq* = 0, no results are written.

#### **\*TIME**

Write indicated results to the output file at every time specified by subsequent recurrent \*TIME or \*DATE keywords in the input file.

#### **\*TNEXT**

Write indicated results to the output file once at the next time specified by \*TIME or \*DATE in recurrent data. \*TNEXT may be used to specify output at multiple times in a single run.

### **DEFAULTS:**

If \*WPRN \*GRID is absent or is not followed by a valid sub-option, \*WPRN \*GRID \*TIME is assumed. This applies to \*ITER as well.

If \*WPRN \*SECTOR is absent, no sector statistics are written to the output file.

### **CONDITIONS:**

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail in the print file may be changed during the simulation.

When \*WPRN \*GRID \*TNEXT is used in the I/O Control data section, results are written at the initial simulation time.

### **EXPLANATION:**

Examples:

```
** Write grid results every 10 timesteps.  
*WPRN *GRID 10  
** Write grid results once at next *TIME  
*WPRN *GRID *TNEXT
```

See keyword \*OUTPRN.

### **Sector Statistics**

The following statistics are available by sector.

Wells:	Rates and accumulations of produced water, oil and gas phases Rates and accumulations of injected water, oil and gas phases Produced liquid rate, WOR and GOR Phase mass rates and accumulations (*OUTSRF *WELL *MASS only) SOR and OSR based on injected water and produced oil at surface conditions, instantaneous and accumulated Recovery factors of water, oil and gas phases Enthalpy rate and accumulation of produced well streams Enthalpy rate and accumulation of injected well streams
Aquifer:	Water aquifer accumulation
Averages:	Average pressure weighted by pore volume Average pressure weighted by hydrocarbon volume Average datum pressure weighted by hydrocarbon volume Average water, oil and gas saturations Average temperature (simple gross-volume weighting)
In-place:	Void pore volume and change, when solid present Solid/adsorbed/trapped volume and change, when present Fluid pore volume and change Hydrocarbon volume Surface-condition volumes of water, oil and gas phases Reservoir-condition volumes of water, oil and gas phases Steam chamber volume (Pore volume times Sg times gas mole fraction of component #1 – usually water) Enthalpy
Heaters:	Net heater rate and accumulation Electrical heating rate and accumulation

Surface-condition volumes are component masses over surface densities, summed over the components found in each phase at surface conditions (see keyword \*SURFLASH).

---

## Items in Output Print File (Optional)

\*OUTPRN, \*PARTCLSIZE,

\*AQSTAT

### PURPOSE:

\*OUTPRN identifies what information is written to the output print file at a frequency given by \*WPRN.

\*PARTCLSIZE provides the properties required to print out in some optional units.

### FORMAT:

```
*OUTPRN *GRID ( *ALL | *NONE | (*REMOVE) item_list )
*OUTPRN *WELL ( *ALL | *NONE | well_var )
*OUTPRN *ITER ( *BRIEF | *NEWTON | *UNCONV | *TSS | *NONE )
*OUTPRN *RES ( *ALL | *NONE | *ALLSMALL )
*PARTCLSIZE vol
*AQSTAT ( *ON | *OFF )
```

### DEFINITIONS:

\*GRID ( \*ALL | \*NONE | (\*REMOVE) *item\_list* )

This subkeyword causes the specified grid quantities (one value for each grid block) to be written to the .out file at times determined by \*WPRN \*GRID.

Generally, each item on the PRN\_GRID list is flagged for writing as either enabled or disabled. The simulation starts with all items disabled. Use *item\_list* (keywords in the PRN\_GRID list) to enable individual items, or use \*ALL to enable all items. Use \*REMOVE with *item\_list* to disable individual items, or use \*NONE to disable all items.

Enabling PRN\_GRID items for writing can increase the size of the .out file. Some items cause writing of more than one set of block values. An item whose description starts with “component” will write one set for each appropriate component. The availability of some items depends on the use of other keywords or options.

### PRN\_GRID List

The PRN\_GRID list consists of the following properties and quantities.

Compositions Y, X, W and Z are mole fraction normally but are mass fraction when \*MASSBASIS is used. For Y, X and W see **Mole Fractions of Absent Phase** in EXPLANATION.

PRES:	pressure (oil phase)
SW:	water saturation
SO:	oil saturation
SG:	gas saturation
TEMP:	temperature
Y:	component composition in gas phase
X:	component composition in oil phase
W:	component composition in water phase
Z:	component composition over all phases

BPP:	bubble point pressure (only volatile components based in oil phase, see section <b>Bubble Point Pressure</b> with *KVTABLE)
OBHLOSS:	over/underburden heat loss rate (see *HLOSSPROP)
CCHLOSS:	net heater rate (see *HEATR, *UHTR, *ADHEAT, *HTWELL)
CCHLOSSCUM:	net heater accumulation "
HEATCAP:	Volumetric heat capacity (see *ROCKCP, *CPG1, *SOLID_CP)
VPOROS:	void porosity
FPOROS:	fluid porosity (required to get oil column in RESULTS 3D)
POREVOL:	pore volume
VISW:	water viscosity
VISO:	oil viscosity
VISG:	gas viscosity
KRW:	water relative permeability
KRO:	oil relative permeability
KRG:	gas relative permeability
PCOW:	water/oil capillary pressure
PCOG:	gas/oil capillary pressure
MOLDENW:	water phase molar density
MOLDENO:	oil phase molar density
MOLDENG:	gas phase molar density
MASDENW:	water phase mass density
MASDENO:	oil phase mass density
MASDENG:	gas phase mass density
RFW:	water phase resistance factor
RFO:	oil phase resistance factor
RFG:	gas phase resistance factor
FRCFLOW:	phase fractional flow
KRINTER:	relative perm interpolation value (needs *KRINTRP)
IFT:	local interfacial tension (needs *IFTTABLE)
CAPN:	local capillary number (needs *IFTTABLE)
LOGIFT:	natural logarithm of IFT (needs *IFTTABLE)
LOGCAPN:	natural logarithm of CAPN (needs *IFTTABLE)
FLUIDH:	fluid enthalpy
WATERHEAD:	depth to top of equivalent water column (referenced to *DTOP)
AQWATCUM:	net water influx to aquifer
AQWATRATE:	rate of water influx to aquifer
AQHEATCUM:	net heat influx to aquifer
AQHEATRATE:	rate of heat influx to aquifer
IMEXMAP:	implicit/IMPES map
THCONDUCT:	thermal conductivity of formation (rock + fluids)

VERDSPLPOR:	vertical displacement “up” based on porosity; see EXPLANATION for *OUTSRF
SUBSIDPOR:	vertical displacement “down” (subsidence) based on porosity; see EXPLANATION for *OUTSRF
SBDZ:	incremental vertical subsidence; “ ”
DATUMPRES:	pressure referenced to a datum; needs *DATUMDEPTH

The following PRN\_GRID keywords correspond to concentrations which may be more usefully reported in alternate unit types and may be preceded by a subkeyword *unit* that indicates a non-default unit type. The default unit type is MASS if \*MASSBASIS has been encountered; otherwise, the default is MOLE. If *unit* is absent before one of these keywords then the unit type used will be the previously assigned (or defaulted) key composition unit type.

( <i>unit</i> ) SOLCONC:	component solid concentration
( <i>unit</i> ) ADSORP:	component adsorbed
( <i>unit</i> ) ICECONC:	ice concentration (needs *ICE)

The choices for *unit* are:

MOLE:	moles per pore volume
MASS:	mass per pore volume (Depends on molecular mass specified via *CMM)
VOL:	solid volume per pore volume
NUM:	particles per pore volume (see *PARTCLSIZE)

The following PRN\_GRID keywords correspond to key components whose compositions may be more usefully reported in alternate unit types, for example, where trace amounts are involved. Each keyword may be preceded by a subkeyword *unit* that indicates a non-default unit type. The default unit type is MASFR if \*MASSBASIS has been encountered; otherwise, the default is MOLFR. If *unit* is absent before one of these keywords then the unit type used will be the previously assigned (or defaulted) key composition unit type.

( <i>unit</i> ) VLKVCMP:	composition of key component used in the calculation of vap/liq K value, given by *KVKEYCOMP
( <i>unit</i> ) LLKVCMP:	composition of key component used in the calculation of liq/liq K value, given by *KVKEYCOMP
( <i>unit</i> ) VISCCMP:	composition of key component used in nonlinear mixing of water and oil viscosity, given by *VSMIXCOMP
( <i>unit</i> ) ADSPCMP:	composition of key component used in the calculation of adsorbing component
( <i>unit</i> ) RLPMCMP:	composition of key component used in the calculation of relative permeability, given by *INTCOMP

The choices for *unit* are:

MOLFR:	mole fraction
MASFR:	mass fraction
PPM:	parts per million
VOLFR:	volume fraction
MOLAR:	molarity
PH:	pH = 14 + log <sub>10</sub> (molarity)
NUM:	particles per phase volume (see *PARTCLSIZE)

*The following are available only with \*ELECHEAT:*

ELCONDUCT:	Bulk electrical conductivity in all three directions
ELPOTENT:	Real electrical potential V <sub>r</sub>
ELPOTENTI:	Imaginary electrical potential V <sub>i</sub>
ELPOTMAG:	Magnitude of multi-phase electrical potential V <sub>m</sub>
ELPOTPHS:	Phase $\alpha$ of multi-phase electrical potential, 0°-360° ELPOTENTI, ELPOTMAG and ELPOTPHS are available only in multi-phase mode, in which case $V_m = [V_r^2 + V_i^2]^{1/2}$ , $V_r = V_m \cos(\alpha)$ and $V_i = V_m \sin(\alpha)$ .
ELPOWER:	Electrical heat dissipation rate
ELPOWERDEN:	Electrical heat dissipation rate per volume
ELCUMENRGY:	Cumulative electrical heat dissipation

\*WELL ( \*ALL | \*NONE | *well\_var* )

This subkeyword causes the specified information to be written to the output print file at times determined by \*WPRN \*GRID. Use \*NONE to skip printing all this information, including echo of operating conditions and well layer indices for each well. Use \*ALL to print all this information. The layer printouts are available only for multi-layer wells. The *well\_var* list is

LAYPWF:	Layer identifier and BHP
LAYPHASE:	Layer phase rates and accumulations
WELLCOMP:	Well component/phase summary

For notes on well layer reports for discretized wellbores, see **Reporting of Flow Performance** in the manual section of Discretized Wellbores in the RESERVOIR DESCRIPTION chapter.

\*ITER ( \*BRIEF | \*NEWTON | \*TSS | \*UNCONV )

This subkeyword specifies that the following iteration results will be printed:

BRIEF:	Basic convergence statistics
NEWTON:	BRIEF + summary of each Newton iteration
TSS:	NEWTON + timestep size and phase switching + Parasol classes report (see keyword *PPATTERN)
UNCONV:	TSS + details of unconverged variables Used for debugging only.

**\*RES ( \*ALL | \*NONE | \*ALLSMALL )**

Controls printing of grid definition and reservoir rock properties, along with other per-grid data. \*ALLSMALL causes these properties to be printed only for grids with no more than 1000 blocks. Use \*ALL to print these properties for any size grid, and use \*NONE to defeat printing for any size grid. This keyword is effective only for non-restart runs, since these properties are not printed for restart runs.

**\*PARTCLSIZE *vol***

Specify the volume *vol* of one particle of solid, adsorbed or key entrained component (m<sup>3</sup> | ft<sup>3</sup> | cm<sup>3</sup>). This quantity is used only to calculate number density for the special unit subkeyword \*NUM. The default is 10<sup>-11</sup> cm<sup>3</sup>, corresponding to a sphere of radius 1.33·10<sup>-4</sup> cm.

**\*AQSTAT ( \*ON | \*OFF )**

Enables (\*ON) or disables (\*OFF) reporting in column form the net and rate influx of water and heat to aquifer regions with non-zero accumulations. This report will appear only in the text output file. See **Detailed Output** in the \*AQUIFER manual page in the Reservoir Description section.

**DEFUALTS:**

Optional keyword. If it is not present in the input data file, the defaults are:

```
*OUTPRN ( *GRID | *WELL ) *NONE  
*OUTPRN *ITER *BRIEF  
*OUTPRN *RES *ALLSMALL
```

If \*PARTCLSIZE is absent, vol = 1.e-11 cm<sup>3</sup> is used.

If \*MASSBASIS is not used, the default unit for concentration is \*MOLE. If \*MASSBASIS is used, the default unit for concentration is \*MASS.

The special units specified in a previous usage of \*OUTPRN apply unless overwritten.

If \*AQSTAT is absent, then \*AQSTAT \*OFF is used.

**CONDITIONS:**

This keyword may appear in the INPUT/OUTPUT CONTROL keyword group and may also occur as part of recurrent data. Thus, the amount of detail in the print file may be changed during the simulation.

**EXPLANATION:**

An example of \*OUTPRN \*GRID, when using the list option is:

```
*OUTPRN *GRID *OILSAT *GASSAT *WATSAT *PRES
```

To specify coke concentration in mass terms, use

```
*CMM ... 13 ** Coke Mw is 13 lb/lbmole  
*OUTPRN *GRID *MASS *SOLCONC
```

## Mole Fractions for Absent Phase

A component mole fraction in a phase is defined as moles of the component in the phase divided by total moles in the phase. Therefore, gas, oil and water mole fractions are defined only when the phase is present. However, for an absent phase the PRN\_GRID items Y, X and W do report a useful quantity that is related directly to a “true” mole fraction.

Consider a simple oil-gas system with component K values  $K_i$  and mole fractions  $x_i$  and  $y_i$ . The relation between mole fractions is  $y_i = x_i \cdot K_i$  and the mole fractions of each phase sum to 1 when each phase is present. However, when the gas phase is absent it is convenient to identify  $y_i$  with the well-defined quantity  $x_i \cdot K_i$  and note that the sum of  $y_i$  is less than 1. When changing conditions cause the sum of  $y_i$  to reach 1, the gas phase will appear and the sum of  $y_i$  will be equal to 1. Therefore, reported PRN\_GRID quantity Y is always  $x_i \cdot K_i$ , but it is the “true” gas mole fraction when the gas phase is present.

In that same oil-gas system, if the oil phase disappears from vapourization then  $x_i$  is identified with the well-defined quantity  $y_i/K_i$ , in which case the sum of  $x_i$  is less than 1. Therefore, reported PRN\_GRID quantity X is always  $y_i/K_i$ , but it is the “true” oil mole fraction when the oil phase is present.

A three-phase water-oil-gas system has water mole fractions  $w_i$  for which  $y_i = w_i \cdot K_i$  for an aqueous component like water. If the water phase disappears from vapourization then  $w_i$  is identified with  $y_i/K_i$  for the aqueous components, in which case the sum of  $w_i$  is less than 1. Therefore, reported PRN\_GRID quantity W is always  $y_i/K_i$ , but it is “true” water mole fraction when the water phase is present.

---

## SR2 Output Frequency (Optional)

\*WSRF, \*DYNGRDFREQ

### PURPOSE:

Control the frequency of dumping information flagged by \*OUTSRF to the SR2 output files.

### FORMAT:

```
*WSRF (*GRID | *WELL | *GRIDDEFORM) (freq | *TIME | *TNEXT )
*WSRF *SECTOR (freq | *TIME )
*DYNGRDFREQ dynfreq
```

### DEFINITIONS:

#### \*GRID

Controls the frequency of writing information flagged by \*OUTSRF \*GRID. Dumping \*GRID information more frequently than the default can increase the size of the SR2 files significantly.

#### \*WELL

Controls the frequency of dumping information flagged by \*OUTSRF \*WELL and \*SPECIAL. Dumping history information less frequently than the default will decrease the size of the SR2 files.

Since history information is needed when reading a restart record, it is always dumped at a restart time, no matter what \*WSRF \*WELL option was used.

#### \*SECTOR

Pertains to writing of statistics that are reported by sector. Sector statistics are written either not at all (*freq* = 0) or at the same times as \*WELL.

#### \*GRIDDEFORM

Pertains to writing of information for grid deformation due to geomechanics effects. Dumping \*GRIDDEFORM information more frequently than the default can increase the size of the SR2 files significantly.

#### *freq*

The indicated information is dumped to the SR2 if the timestep number is evenly divisible by non-negative integer *freq*.

For \*GRID, no results are written if *freq* = 0. A value of *freq* = 0 is not allowed for \*WELL.

#### \*TIME

Dump indicated results to the SR2 files at every time specified by subsequent recurrent \*TIME or \*DATE keywords in the input file.

#### \*TNEXT

Dump indicated results to the SR2 at the single time specified by the next \*TIME or \*DATE keyword in recurrent data. \*TNEXT may be used to specify output at multiple times in a single run.

### **\*DYNGRDFREQ *dynfreq***

In addition to the times indicated by \*WSRF \*GRID, dump \*GRID results to the SR2 once for every *dynfreq* of the timesteps at which a dynamic grid change check is done as specified via keyword \*DYNAGRID-TSINT. No extra dumps are done if *dynfreq* = 0; a grid dump is done at all grid change times if *dynfreq* = 1. This keyword is active only with the \*DYNAGRID feature. See the manual entry for \*DYNAGRID in the Well and Recurrent Data chapter.

#### **DEFAULTS:**

If \*WSRF \*GRID does not appear, then \*WSRF \*GRID \*TIME is assumed.

If \*WSRF \*WELL does not appear, then \*WSRF \*WELL 1 is assumed.

If \*WSRF \*GRIDDEFORM does not appear, then no grid deformation information is dumped to the SR2.

If \*WSRF \*SECTOR does not appear, then no sector statistics are dumped to the SR2.

If \*DYNGRDFREQ is absent then *dynfreq* = 0 is assumed.

#### **CONDITIONS:**

This keyword may appear in the INPUT/OUTPUT CONTROL section as well as the RECURRENT DATA section of your data. Thus, the amount of detail in the SR2 files may be changed during the simulation.

When \*WSRF \*GRID \*TNEXT is used in the I/O Control data section, results are written at the initial simulation time.

Keyword \*GRIDDEFORM is available only with \*GEOMECH.

#### **EXPLANATION:**

To dump grid results every 10 timesteps use \*WSRF \*GRID 10. To dump results at \*TIME or \*DATE times only, use \*TIME or a value of *freq* larger than the expected maximum timestep number. To skip dumping \*GRID results at \*TIME or \*DATE times, use \*WSRF \*GRID 0 or \*TNEXT before the desired time.

#### **Visualizing Geomechanics Grid Deformation**

Keyword \*GRIDDEFORM allows the user to view grids in Results that deform with time as calculated by the geomechanics module. This feature is available in Results only for corner-point grid type. STARS writes to the SR2 file grid definition data that tells Results the type, structure and appearance of the grid. If the \*GRIDDEFORM option is requested, STARS tells Results that the grid is corner-point type and does any necessary conversion. If the user specified \*GRID \*CART in data then the conversion is exact, that is, Results will draw an initial grid that looks exactly like the user's Cartesian grid. If the user specified \*GRID \*RADIAL in data then the conversion at initial conditions is exact in the I-K (R-Z) plane only. Several issues arise from this technique.

1. Initial conditions are plotted with the original grid whereas the "converted" grid is used to display all subsequent times. For the radial grid case you may notice a change in the grid when going from initial time to subsequent times.

2. For a “converted” 3D radial grid, circular arcs are replaced with straight lines between block corners; for example, for  $n\theta = 3$  the areal (I-J) plane appears as a triangle instead of a circle. The angular direction of a 2D radial grid cannot be displayed at all.
3. A “converted” 3D radial grid does not include the innermost radial block, so both fluid-flow and geomechanics quantities cannot be viewed for that block. However, the fluid-flow equations are still solved for that block.
4. Without \*GRIDDEFORM the grid is displayed as "radial" and the innermost radial block shows a value that is (1) correct for fluid-flow quantities and (2) an average of surrounding block values for geomechanics quantities.

The writing of grid deformation information is controlled also by the frequency of geomechanics updating specified by keyword \*GCUPDATE in the Geomechanics section.

---

## Items in Simulation Results File (Optional)

\*XDR

\*OUTSRF, \*SRFASCII,

### PURPOSE:

\*OUTSRF identifies what information is written to the Simulation Results File.

### FORMAT:

```
*OUTSRF *WELL { comp_unit | *DOWNHOLE
    | *COMPONENT ( *NONE | *ALL | comp_list )
    | *LAYER ( *NONE | *ALL ) }
*OUTSRF *GRID ( *ALL | *NONE | (*REMOVE) item_list )
*OUTSRF *SPECIAL { special_his }
*SRFASCII
*XDR ( *ON | *OFF )
```

### DEFINITIONS:

#### \*WELL

Indicates that keywords will follow which will cause additional information, over and above the minimum default information, to be written to the SR2 for each well at timesteps determined by keyword \*WSRF.

#### Comp\_unit

Causes well performance to be saved in mass and/or mole units in addition to volumes (see DEFAULTS, below). Comp\_unit may be one or both of

\*MASS: save well performance in mass terms,

\*MOLE: save well performance in mole terms,

The use of comp\_unit requires \*COMPONENT \*ALL and increases the size of the SR2 files.

#### \*DOWNHOLE

Causes production well performance (volumes units as well as mass and mole units specified by comp\_unit) to be written at downhole conditions in addition to surface conditions (see DEFAULTS, below). This option increases the size of the SR2 file. An item referenced to downhole, or reservoir, conditions will have "RC" appended to its title. Note that no downhole statistics are written for injectors.

The \*DOWNHOLE option may be enabled or disabled at a restart, resulting in the following two cases:

1. No downhole statistics in the restart record but \*DOWNHOLE is present in the restarting data. Subsequent downhole accumulations will start at zero at the restart time but rates will be correct.

- Downhole statistics are in the restart record but \*DOWNHOLE is absent in the restarting data. No more downhole statistics will be written but those up to the restart time are available.

**\*COMPONENT ( \*NONE | \*ALL | comp\_list )**

Well performance will be written for the components specified by this keyword. The default is \*NONE (see DEFAULTS, below). Use \*ALL to specify all the components, or enter a list of component names. \*ALL is assumed if \*MASS or \*MOLE is specified.

Use of this option, especially with \*ALL, can increase the size of the SR2 file substantially.

**\*LAYER ( \*NONE | \*ALL )**

Well performance will be written for individual layers for either all wells (\*ALL) or no wells (\*NONE). The default is \*NONE (see DEFAULTS, below). Use of \*ALL can increase the size of the SR2 file substantially.

For notes on well layer reports for discretized wellbores, see **Reporting of Flow Performance** in the manual section of Discretized Wellbores in the RESERVOIR DESCRIPTION chapter.

**\*GRID ( \*ALL | \*NONE | (\*REMOVE) item\_list )**

This subkeyword causes the specified grid quantities (one value for each grid block) to be written to the SR2 file at times determined by \*WSRF \*GRID. Generally, each item on the SRF\_GRID list is flagged for writing as either enabled or disabled. The simulation starts with all items disabled. Use *item\_list* (keywords in the SRF\_GRID list) to enable individual items. Use \*ALL to enable all items except FLUXSC, VELOCSC, FLUXRC, VELOCRC and STRMLN. Use \*REMOVE with *item\_list* to disable individual items, or use \*NONE to disable all items.

Enabling SRF\_GRID items for writing can increase the size of the SR2 files. Some items cause writing of more than one set of block values. An item whose description starts with “component” will write one set for each appropriate component. The availability of some items depends on the use of other keywords or options.

#### **SRF\_GRID List**

The SRF\_GRID list consists of all properties and quantities in the following table, as well as all items in the **PRN\_GRID list** (see \*OUTPRN \*GRID) with these exceptions: FRCFLOW is replaced by WATFRFL, OILFRFL and GASFRFL, and POREVOL and IMEXMAP are disallowed.

The alternate concentration and composition unit types used by some items in the PRN\_GRID list can be used here as well. The unit type choices made in \*OUTPRN \*GRID, \*OUTSRF \*GRID and \*OUTSRF SPECIAL are independent of each other. For \*OUTSRF \*GRID and \*SPECIAL one item can be dumped with more than one unit type at the same time. For example,

**\*OUTSRF \*GRID VOL ADSORP NUM ADSORP**

causes adsorbed components to be reported in RESULTS in units of both volume fraction and number density.

KVALYW:	component gas/water K value (y/w)
KVALYX:	component gas/oil K value (y/x)
KVALXW:	component oil/water K value (x/w)
KVALWX:	component water/oil K value (w/x)
SWC:	obsolete; use SWCON
SWCON:	connate water saturation; needs *BSWCON
SWCRIT:	critical water saturation; needs *BSWCRT
SORW:	residual oil saturation to water; needs *BSORW
SOIRW:	irreducible oil saturation to water; needs *BSOIRW
SGC:	obsolete: use SGCRIT
SGCRIT:	critical gas saturation; needs *BSGR
SGCON:	connate gas saturation; needs *BSGCON
SORG:	residual oil saturation to gas; needs *BSORG
SOIRG:	irreducible oil saturation to gas; needs *BSOIRG
SWRG:	residual water saturation to gas (oil-wet); needs *BSWRG
SWIRG:	irreducible water saturation to gas (oil-wet); needs *BSWIRG
KRWRO:	obsolete; use KRWIRO (see *BKRWRO)
KRWIRO:	water relative permeability at Soirw; needs *BKRWIRO
KROCW:	oil relative permeability at Swcon; needs *BKROCW
KRGCW:	gas relative permeability at Swcon; needs *BKGCGW
PCWMAX:	maximum water-oil capillary pressure; needs *BPCWMAX
PCGMAX:	maximum gas-oil capillary pressure; needs *BPCGMAX
CMPDENW:	component mass density in water phase
CMPDENO:	component mass density in oil phase
CMPVISW:	component viscosity in water phase
CMPVISO:	component viscosity in oil phase
CMPVISG:	component viscosity in gas phase
PERMI:	I direction absolute permeability ( $k_x$ )
PERMJ:	J direction absolute permeability ( $k_y$ )
PERMK:	K direction absolute permeability ( $k_z$ )
	For natural-fracture grid option, fracture block value is “effective” whereas matrix block value is “intrinsic”, matching the input. See <b>Natural Fracture Permeabilities</b> in the EXPLANATION, below.
PERMEFFI:	I direction effective absolute permeability
PERMEFFJ:	J direction effective absolute permeability
PERMEFFK:	K direction effective absolute permeability
	Available only for natural-fracture grid option. See <b>Natural Fracture Permeabilities</b> in the EXPLANATION, below.
PERMINTI:	I direction intrinsic absolute permeability
PERMINTJ:	J direction intrinsic absolute permeability
PERMINTK:	K direction intrinsic absolute permeability
	Available only for natural-fracture grid option. See <b>Natural Fracture Permeabilities</b> in the EXPLANATION, below.

WATMOB:	water phase mobility in I direction ( $\lambda_w = k_x k_{rw} / \mu_w$ )
OILMOB:	oil phase mobility in I direction ( $\lambda_o = k_x k_{ro} / \mu_o$ )
GASMOB:	gas phase mobility in I direction ( $\lambda_g = k_x k_{rg} / \mu_g$ )
TOTMOB:	total mobility in I direction $\lambda_w + \lambda_o + \lambda_g$
WATFRFL:	water phase fractional flow $\lambda_w / (\lambda_w + \lambda_o + \lambda_g)$
OILFRFL:	oil phase fractional flow $\lambda_o / (\lambda_w + \lambda_o + \lambda_g)$
GASFRFL:	gas phase fractional flow $\lambda_g / (\lambda_w + \lambda_o + \lambda_g)$
VISOCOM:	component composition of the key component in the nonlinear mixing of oil viscosity given by *VSMIXCOMP
VISWCOM:	component composition of the key component in the nonlinear mixing of water viscosity given by *VSMIXCOMP
KRSETN:	relative permeability data set number
INSETN:	initialization region set number (*GRID only, once per run)
STEAMQUAL:	Steam quality (in-place, all aqueous components)
QUALBLK:	Steam quality (flowing, component #1, *SPECIAL only)
VELOCSC:	Effective velocity of each phase, after flashing fluids to surface conditions. Also the same as keyword VELOC. *GRID only. See EXPLANATION, below.
FLUXSC:	Flux of each phase, after flashing fluids to surface conditions. Also the same as keyword FLUX. *GRID only. See EXPLANATION, below.
VELOCRC:	Effective velocity of each phase at reservoir conditions. *GRID only. See EXPLANATION, below.
FLUXRC:	Flux of each phase at reservoir conditions. *GRID only. See EXPLANATION, below.
STRMLN	Allows Results to generate streamlines of each phase at reservoir conditions. *GRID only. See EXPLANATION, below.
TRMI:	Transmissibility multipliers *TRANSI
TRMJ:	Transmissibility multipliers *TRANSJ
TRMK:	Transmissibility multipliers *TRANSK
TRLI:	Transmissibility multipliers *TRANLI
TRLJ:	Transmissibility multipliers *TRANLJ
TRLK:	Transmissibility multipliers *TRANLK
ENINPLRAT:	Rate of increase of in-place term of energy balance (*GRID only)
ENCONVRAT:	Rate of increase of convective term of energy balance (*GRID only)
ENREACRAT:	Rate of increase of reaction term of energy balance (*GRID only)
ENCOND RAT:	Rate of increase of conductive term of energy balance (*GRID only) See <b>Energy Balance</b> in EXPLANATION, below.
THCONDUCTR:	Rock thermal conductivity from *THCONDUCT table
THCONDUCTS:	Solid thermal conductivity from *THCONDUCT table
SCLASS	Parasol class number (*SOLVER *PARASOL only)

*The following are available only with \*SHEARTHIN, \*SHEARTHICK or \*SHEARTAB:*

VISCVELW:	Magnitude of Darcy velocity of water phase
VISCVELO:	Magnitude of Darcy velocity of oil phase
SHEARW:	Shear rate of water phase (see *SHEAR_FAC)
SHEARO:	Shear rate of oil phase (see *SHEAR_FAC)
SHEARSTRSW:	Shear stress (viscosity times shear rate) of water phase
SHEARSTRSO:	Shear stress (viscosity times shear rate) of oil phase

***The following are available only with \*GEOMECH:***

STRESI:	Effective I-direction stress ( X or R )
STRESJ:	Effective J-direction stress ( Y or theta )
STRESK:	Effective K-direction stress ( Z )
STRESINVF	First invariant of effective stress tensor
STRESINVS	Square root of second invariant of effective stress tensor
STRESSH:	Shear stress ( Y-Z or R-Z ) for plane strain only
STRESSHIJ:	Shear stress on IJ plane
STRESSHIK:	Shear stress on IK plane
STRESSHJK:	Shear stress on JK plane
STRESMXP:	Maximum principal stress (+ for compressive, - for tensile)
STRESMNP:	Minimum principal stress (+ for compressive, - for tensile)
STRESINT:	Intermediate principle stress (+ for compressive, - for tensile)
VMSTRESS:	Von Mises stress
STRNEPL:	Effective Plastic strain
STRESEFF:	Mean effective stress (+ for compressive, - for tensile)
STRESSM:	Mean total stress (+ for compressive, - for tensile)
TSTRESI:	Total normal stress in I direction
TSTRESJ:	Total normal stress in J direction
TSTRESK:	Total normal stress in K direction
STRESNORM:	Effective stress normal to fracture
PRMXDIR:	Vector of maximum principle effective stress (*GRID only)
PRMNDIR:	Vector of minimum principle effective stress (*GRID only)
STRAINI:	I-direction normal strain ( X or R )
STRAINJ:	J-direction normal strain ( Y or theta )
STRAINK:	K-direction normal strain ( Z )
STRAINSH:	Shear strain
STRAINSHIJ:	Shear strain on IJ plane
STRAINSHIK:	Shear strain on IK plane
STRAINSHJK:	Shear strain in JK plane
STRNMXP:	Maximum principle strain
STRNMNP:	Minimum principle strain
STRAINVOL:	Volumetric strain
VOIDRATIO:	Void ratio = porosity/(1-porosity) – for geomechanics only.
VPOROSGEO:	Reservoir porosity calculated from geomechanics module
VPOROSTGEO:	True porosity calculated from geomechanics module
PORDIFF:	Difference between geomechanics and reservoir porosity (VPOROSGEO minus VPOROS)
See section “Geomechanics Porosities” in the introduction of the GEOMECHANICS chapter.	
VERDSPLGEO:	Vertical displacement “up” based on geomechanics, at centre of cell
TVERDPLGEO:	Vertical displacement “up” based on geomechanics, at top of cell
SUBSIDGEO:	Vertical displacement “down” (subsidence) based on geomechanics, at centre of cell (negative of VERDSPLGEO)
TSUBSIDGEO:	Vertical displacement “down” (subsidence) based on geomechanics, at top of cell (negative of TVERDPLGEO)

VDISPL:	Vector of grid displacement (*GRID only)
	<b>For more on displacement outputs see EXPLANATION, below.</b>
YLDSTATE:	Stress state = 0 In Elastic state = 1 On shear failure envelope = 2 On the compressive cap = 3 At the corner (intercept between cap and shear failure envelope) = 4 On the tensile cutoff surface
BIOT:	Biot's constant
GCOHESION:	Cohesion value
HARDENING:	Hardening parameter
POISSON:	Poisson's ratio
YIELD:	Yielding stress
YOUNG:	Young's elastic modulus
THEXPCOEF:	Linear thermal expansion coefficient of rock
THCOMPR:	Thermal compressibility of rock
BULKVOL:	Bulk volume
GEORTYPE:	Rock type number for geomechanical material

*The following are available only with \*ELECHEAT:*

ELCONDUCT:	Bulk electrical conductivity in all three directions. In the anisotropic case, this triggers ELCONDI, etc.
ELCONDI	Bulk electrical conductivity in I, J and K directions. For the isotropic case, any of these will trigger ELCONDUCT.
ELCONDJ	
ELCONDK	
ELPOTENT:	Real electrical potential $V_r$
ELPOTENTI:	Imaginary electrical potential $V_i$
ELPOTMAG:	Magnitude of multi-phase electrical potential $V_m$
ELPOTPHS:	Phase $\alpha$ of multi-phase electrical potential, $0^\circ$ - $360^\circ$ ELPOTENTI, ELPOTMAG and ELPOTPHS are available only in multi-phase mode, in which case $V_m = [V_r^2 + V_i^2]^{1/2}$ , $V_r = V_m \cos(\alpha)$ and $V_i = V_m \sin(\alpha)$ .
ELPOWER:	Electrical heat dissipation rate
ELPOWERDEN:	Electrical heat dissipation rate per volume
ELCUMENRGY:	Cumulative electrical heat dissipation
ELCDEN	Real value of the scalar current density, equal to the magnitude of the current density vector
ELCDENI	Imaginary value of the scalar current density
ELCDENM	Complex magnitude value of the scalar current density ELCDENI and ELCDENM are available only in multi-phase mode.
ELCURDEN	Vector plots of real current density and current. See
ELCUR	<b>Electrical Heating Vector Plots</b> , below. (*GRID only)
ELCURDENI	Vector plots of imaginary current density and current.
ELCURI	Available only in multi-phase mode. (*GRID only)

**\*SPECIAL { *special\_his* }**

This subkeyword defines one or more special histories, each of which writes a single value to the SR2 at times specified by \*WSRF \*WELL. Each *special\_his* may be one of the following:

**BLOCKVAR *srf\_prop uba* ( *comp\_name* )**

Property *srf\_prop* in block *uba*, possibly for component *comp\_name*. *srf\_prop* is from the SRF\_GRID list. *uba* is a User Block Address. Valid component name *comp\_name* is required only when the *srf\_prop* description starts with “component”.

**MAXVAR *srf\_prop* ( *comp\_name* )**

Maximum over the entire grid of SRF\_GRID item *srf\_prop*, possibly for component *comp\_name*. See BLOCKVAR.

**MINVAR *srf\_prop* ( *comp\_name* )**

Minimum over the entire grid of SRF\_GRID item *srf\_prop*, possibly for component *comp\_name*. See BLOCKVAR.

**AVGVAR *srf\_prop* ( *comp\_name* )**

Average over the entire grid of SRF\_GRID item *srf\_prop*, possibly for component *comp\_name*. See BLOCKVAR.

**WOR *well* ( INST | CUM )**

Water-oil ratio for *well* based on rate (INST), the default, or accumulation to date (CUM), where *well* is well name.

**GOR *well* ( INST | CUM )**

Gas-oil ratio for *well* based on rate (INST), the default, or accumulation to date (CUM) , where *well* is well name.

**DELP *well1 well2***

BHP of *well1* minus BHP of *well2*, where *well1* and *well2* are well names.

**OSR *well1 well2* ( INST | CUM | OIL-PHASE-COMP (‘namec’))**

Oil produced from *well1* divided by water injected in *well2*, based on rate (INST), the default, or accumulations to date (CUM), where *well1* and *well2* are well names. When OIL-PHASE-COMP is used then only the specified oil phase component(s) are used to calculate the OSR. When OIL-PHASE-COMP is used without component name ‘namec’ then component numw+1 is assumed.

**SOR** *well1 well2* ( INST | CUM | OIL-PHASE-COMP ('namec') )

Water injected in *well1* divided by oil produced from *well2*, based on rate (INST), the default, or accumulations to date (CUM) , where *well1* and *well2* are well names. When OIL-PHASE-COMP is used then only the specified oil phase component(s) are used to calculate the SOR. When OIL-PHASE-COMP is used without component name 'namec' then component numw+1 is assumed.

**MASSFRAC** *well comp\_name* ( WATER | OIL | GAS )

Mass fraction of *comp\_name* in the surface condition fluid stream of *well*, where *comp\_name* is component name and *well* is well name. The trailing phase indicator is needed only if *comp\_name* occurs in more than one phase at surface conditions (see \*SURFLASH).

**MOLEFRAC** *well comp\_name* ( WATER | OIL | GAS )

Similar to MASSFRAC, but for mole fractions.

**VOLFRAC** *well comp\_name* ( WATER | OIL | GAS )

Similar to MASSFRAC, but for volume fractions.

**STMQUAL** *well*

Quality of injected steam specified by the input keyword \*QUAL in the well data for *well*, where *well* is well name. Valid only for injection wells.

**PRODSTEAMR** *well*

Steam production rate expressed in CWE, for well named *well*. Steam is sum of all aqueous components in the gas phase at downhole conditions. This calculation is similar to the \*OPERATE \*STEAM constraint but is available with any production constraint. If \*WELL subkeyword \*FRAC is used, PRODSTEAMR corresponds to the fractional well.

**WELLENERGY** *well* ( RATE | CUM )

Energy of fluid stream of *well*, either rate (J/day | Btu/day) or cumulative (J | Btu), where *well* is well name. This is enthalpy referenced to the same conditions as the reservoir fluids (see \*TEMR, etc. and \*CPL1, etc.).

**MATBAL** *stat* ( *comp\_name* | ENERGY )

Material balance statistic type *stat* for *comp\_name* or energy, where *comp\_name* is component name. All *stat* values are available for fluid components and ENERGY; for solid components only CURRENT and REACTION are available.

The choices for *stat* are:

CURRENT	Amount currently in place.
REACTION	Net cumulative amount created (+) or consumed (-) by reactions.
WELL	Net cumulative amount injected (+) or produced (-). Includes source/sink term reported in CCHLOSS.
AQUEOUS	Amount currently in place in the aqueous (water) phase.
OLEIC	Amount currently in place in the oleic (oil) phase.
GASEOUS	Amount currently in place in the gas phase.
ADSORBED	Amount currently adsorbed.

Note: The precise meaning of the value of ENERGY in place (any phase or total) may be obscured by the fact that it is referenced to a base phase and temperature. "Amount" is mass for \*MASSBASIS and moles otherwise.

#### TFRONT *ideg* ( *i1(:i2)* *j1(:j2)* *k1(:k2)* | FORWARD | BACKWARD )

Position of the *ideg* contour of a temperature front, found by scanning a specified column of blocks in the specified direction. Complicated variations in temperature may reduce TFRONT's usefulness.

The block column may be specified by an I-J-K address in which a range is specified in exactly one direction. Scanning is performed from the first index to the second index in the range, allowing you to choose between a forward-facing front and a backward-facing front. For example, a combustion tube modelled with a 2D cylindrical grid (*ni* = 5, *nk* = 30) has injection at *k* = 1 so the front moves in the direction of increasing K index. Scanning in the center axis (*i* = 1), and avoiding 2 end blocks, the forward-facing front is found with indices "1 1 28:3" while the backward-facing front is found with "1 1 3:28". Older options FORWARD and BACKWARD assume that the grid is 1D and scanning is done over the entire 1D grid. FORWARD scans from high index values to low, and BACKWARD the reverse. Using the above example with *ni* = 1, FORWARD would scan from *k* = 30 to *k* = 1 and so would find the forward-facing front.

No block in the scanning column may be null.

#### OBHLOSSCUM (previously OBHLOSS)

Net cumulative energy lost (-) or gained (+) by the overburden heat loss model.

#### OBHLOSSRATE

Rate of net energy lost (-) or gained (+) by the overburden heat loss model.

## CCHLOSSCUM (previously CCHLOSS)

Net cumulative energy lost (-) or gained (+) by a constant/convective heat transfer model. See also \*WPRN \*SECTOR.

## CCHLOSSRATE

Rate of net energy lost (-) or gained (+) by a constant/convective heat transfer model. See also \*WPRN \*SECTOR.

## DELPBLK *uba1 uba2*

Pressure in block *uba1* minus pressure in block *uba2*.

## CPUSRATE, CPUSCUM

CPUSRATE gives CPU seconds per simulation time over individual timesteps. CPUSCUM gives accumulated CPU seconds from the start of timestepping, starting at zero for both restart and non-restart runs.

## PHWELL *well quantity* ( SURFACE | DOWNHOLE | PUMP )

*well* is well name; *quantity* can be TEMP (fluid temperature), PRES (fluid pressure) or STQUAL (steam quality). Values may be obtained at the surface, at the downhole entrance to the portion of the well or the pump location (producer only) modelled by \*PHWELLCBORE.

## WELL\_LAYER *well uba quantity*

*well* is well name; *uba* is a User Block Address that appears in the perforation list of *well*; *quantity* can be:

TEMP	fluid temperature from *HEAD-METHOD *GRAV-FRIC-HLOS method
STQUAL	steam quality from *HEAD-METHOD *GRAV-FRIC-HLOS method, injectors only
LEPDIFRATE	Critical LEP rate minus actual, LEP injectors only

## FRACIMPES

FRACIMPES gives fraction of active blocks that are IMPES. See \*AIM in Numerical Control section.

## AQFRTOT ( WATER | HEAT ) ( RATE | CUM )

AQFRTOT gives aquifer flow statistics of either water or heat, either rate or accumulation, over the entire grid. See keyword \*AQUIFER. Option HEAT is available only if keyword \*HFPROP is used to enable heat influx in the aquifer. HEAT actually reports fluid enthalpy which is referenced to base temperature and phase (see \*TEMR and \*CPG1).

### ELHEAT ( RATE | CUM )

Instantaneous rate in kW and accumulation in kW-hr of electrical heat dissipated over the entire grid.

### EBNDSTAT *ibnd* ( POTENTIAL | CURRENT | CUMCURRENT )

For electrical boundary # *ibnd*, the following are written:

POTENTIAL: (real) potential in V; in multi-phase mode, imaginary component, magnitude and phase of complex potential.

CURRENT: (real) current in A; in multi-phase mode, imaginary component and magnitude of complex current.

CUMCURRENT: (real) accumulated charge in A times the user time unit; in multi-phase mode, the imaginary component.

### EBLAYSTAT *uba* ( CURRENT | CUMCURRENT )

For electrical boundary layer in block *uba*, the following are written.

See **User Block Address** in chapter Keyword Data Entry System.

CURRENT: (real) current in A; in multi-phase mode, imaginary component and magnitude of complex current.

CUMCURRENT: (real) accumulated current in A times the user time unit; in multi-phase mode, imaginary component of complex accumulated charge.

### EBNDRESIS *ibnd1 ibnd2*

Electrical resistance between boundaries # *ibnd1* and # *ibnd2*:

$$|V_{ibnd1} - V_{ibnd2}| / \min [|I_{ibnd1}|, |I_{ibnd2}|]$$

where V and I are the potential and current, respectively, of each boundary and |x| denotes the magnitude of real or complex argument x. The two boundary numbers must be different. This quantity will never be negative.

When there are only two boundaries, the two currents are equal and this formula gives the expected result. The calculation is more complex when there are more than two boundaries. For example, if one boundary is at ground and two are at the same  $V > 0$ , current through ground is the sum of the currents of the other two boundaries. In this case, when calculating the resistance between ground and one of the other boundaries the “min” function ensures that only the current from the one non-ground boundary is included, to give the expected result.

This formula is not meaningful when there are more than two different boundary potential levels in the system.

### EPOTGRADB *uba1 uba2*

Magnitude of electrical potential gradient between blocks *uba1* and *uba2*. See **User Block Address** in chapter Keyword Data Entry System. The quantity reported is

$$|V_1 - V_2| / \|P_1 - P_2\|$$

where  $P_i$  is the position vector of the center of block  $i$ ,  $V_i$  is the potential at that position,  $|x|$  denotes the magnitude of its real or complex argument  $x$ , and  $\|x\|$  denotes the length of vector  $x$ . The two blocks may be adjacent or non-adjacent, but they must be different. If the grid is extended to the surface, this quantity can be used to estimate the "step voltage" experienced on the ground.

### HTRWELL *well* ( HEATRATE | HEATCUM )

Instantaneous rate and accumulation of net heat transferred into (+) and out of (-) the reservoir for heater well named *well*. See \*HTWELL. To get histories for individual completion layers use special history type \*BLOCKVAR with SRF\_GRID list items CCHLOSS for rate and CCHLOSSCUM for accumulation.

### STRESSDIFF *uba*

Stress differential in block *uba*. Available only with \*GEOMECH. See **User Block Address** in chapter Keyword Data Entry System. Stress differential is maximum principal stress minus minimum principal stress; it is useful when plotted versus strain in the same *uba* to produce a true triaxial stress-strain plot. See \*STRESMXP, \*STRESMNP and strain quantities in the **SRF\_GRID** list.

### \*SRFASCII

Specifies that a textual copy of the main SR2 data file will be written in addition to the binary copy (MRF). The default filename suffix is "asc".

### \*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 work station 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).

### DEFUALTS:

If \*OUTSRF \*WELL is not present, the effect is

1. only volumes are written (not moles or mass),

2. only surface conditions are written,
3. only phases are written (not components), and
4. only well totals are written (not layers).

After \*OUTSRF \*WELL:

1. if \*MOLE and \*MASS is absent, only volumes are written,
2. if \*DOWNHOLE is absent, only surface condition performance is written,
3. if \*COMPONENT is absent, no individual component performance is written, and
4. if \*LAYER is absent, no individual layer performance is written. To get the default action of versions earlier than 96, use \*COMPONENT \*ALL, \*LAYER \*ALL, and \*MOLE or \*MASS when needed.

If \*OUTSRF \*SPECIAL is not present, no special histories will be written.

The special units specified in a previous usage of \*OUTSRF apply unless overwritten.

If \*SRFASCII is absent, then no textual copy of the binary SR2 file is written.

If \*XDR is absent, then \*XDR \*ON is assumed.

### **CONDITIONS:**

\*OUTSRF \*GRID may appear in the I/O CONTROL data group, in which case it applies to initial conditions. It may also appear anywhere in the recurrent data, such that the amount of grid detail dumped may be controlled. Initial conditions can be dumped without doing any timesteps, and viewed, by using \*CHECKONLY.

\*WELL and \*SPECIAL may occur in only the I/O CONTROL section.

Special history definitions should not be changed or removed at a restart. Special history definitions are read from the data file, not the restart, so a change made to a special history definition may result in a change in meaning of that quantity. A special history added at a restart will be ignored by RESULTS Graph. Each special history definition must appear either immediately after \*OUTSRF \*SPECIAL or on a subsequent new line.

When the geomechanics module (\*GEOMECH) is used, geomechanical properties such as \*BIOT, \*GCOHESION, \*HARDENING, \*POISSON, \*YIELD and \*YOUNG are available but are dumped only at the start of a simulation.

### **EXPLANATION:**

An example of \*OUTSRF \*GRID is

```
*OUTSRF *GRID *OILSAT *GASSAT *WATSAT *PRES
```

To specify coke concentration in mass terms, use

```
*CMM ... 13 ** Coke Mw is 13 lb/lbmole
*OUTSRF *GRID *MASS *SOLCONC
```

Save the history of coke concentration in block #20 in mass and moles

```
*OUTSRF *SPECIAL *BLOCKVAR *MASS *SOLCONC 6 20
*BLOCKVAR *MOLE *SOLCONC 6 20
```

## Velocity and Flux Vector Plots

\*OUTSRF \*GRID subkeywords VELOCSC, FLUXSC, VELOCRC and FLUXRC cause information to be written to the SR2 which allows RESULTS to overlay grid information with flux or velocity vectors indicating both magnitude and direction. These vectors are available for each phase (oil, gas, water), in both surface conditions and reservoir conditions.

Each keyword results in writing of a grid-length array to the SR2 for each phase and each direction, a total of 9 arrays. All four keywords will add 36 grid-length arrays, which can increase the size of the MRF file very significantly. Cautious use of these keywords is advised.

Keywords VELOCSC and FLUXSC report information at surface conditions. Components are assigned to phases according to the surface phase keyword \*SURFLASH or its default. For example, the default would report water component as fully condensed in the water phase, liquid oil components as fully condensed in the oil phase, and soluble (or non-soluble) gaseous components as fully evolved in the gas phase. It is not possible to isolate the flux and velocity information for an individual component as such, but through \*SURFLASH it is possible to specify which components are reported in which surface phase. The surface statistic is based on component flow and so accounts also for dispersion.

Keywords VELOCRC and FLUXRC report information at reservoir conditions. Since reservoir phase volumes and mobilities are obtained directly from in-situ fluid conditions, individual components do not directly influence these statistics as it does through \*SURFLASH for the surface statistics.

Keywords VELOCRC and VELOCSC produce a velocity value that is “effective”, that is, flux divided by cross-sectional area. Contrast this with intrinsic velocity, the velocity of fluid in the pore space, whose value is “effective” velocity divided by porosity.

## Streamline Plots

\*OUTSRF \*GRID subkeyword STRMLN causes information to be written to the SR2 which allows RESULTS to overlay grid information with streamlines. This subkeyword adds three connection-length arrays to each grid dump.

## Geomechanics Vector Plots

Several **SRF\_GRID** items cause geomechanics information to be written to the SR2 which allows RESULTS to generate the corresponding vector plot. Each of these keywords results in writing of three grid-length arrays to the SR2, one for each direction. These keywords are available only with the geomechanics module (\*GEOMECH).

- |          |                                    |
|----------|------------------------------------|
| PRMXDIR: | Maximum principle effective stress |
| PRMNDIR: | Minimum principle effective stress |
| VDISPL:  | Grid displacement                  |

## Vertical Displacement and Subsidence

Several items in the **SRF\_GRID** list specify vertical formation displacement and are summarized as follows. Each quantity is the displacement of each block centre since the start of the simulation.

Quantity	Basis	Sign	Special History
VERDSPLPOR	Porosity	+ is up	yes
SUBSIDPOR	Porosity	+ is down	yes
VERDSPLGEO	Geomechanics	+ is up	yes
SUBSIDGEO	Geomechanics	+ is down	yes
VDISPL (K-dir)	Geomechanics	+ in *KDIR	no

Quantities **VERDSPLPOR** and **SUBSIDPOR** are based on a simple porosity calculation, assuming the grid bottom is fixed and all porosity change goes toward gross thickness change. These quantities are available for all porosity options including **\*GEOMECH**. They are intended as a rough estimate of vertical displacement and may not be an accurate indicator of displacement in more complex situations. The value reported for each block is the sum of incremental values from that block's centre to the bottom of the block column. A block's incremental value is  $(\varphi - \varphi_0) \cdot \Delta_z \cdot |g_z/g|$ , where  $\varphi_0$  is initial porosity,  $\varphi$  is current porosity,  $\Delta_z$  is Z-direction block size and  $g_z/g$  is gravity component in the Z direction. The value assigned to blocks in **\*HYBRID**, **\*MINC** and **\*SUBDOMAIN** refined grids will be the value of that grid's parent block. For a zero-porosity block this incremental value is zero. This incremental value is what is reported for output quantity **SBDZ** in the subsidence sense (+ is down).

**VERDSPLPOR** and **SUBSIDPOR** differ only in the sign. **VERDSPLPOR** reports vertical displacement upward (heave) as positive and downward as negative, so its curve on an X-Y plot (versus time or distance) will rise and fall in the same sense as the block centre position. **SUBSIDPOR** shows vertical downward displacement (subsidence) as positive and upward displacement as negative. You can choose which quantity to plot according to your preference. Note that the grid's K direction defined by **\*KDIR** does not enter into the definition of these output quantities.

**VERDSPLGEO** and **SUBSIDGEO** are based on formation strain calculated by the **\*GEOMECH** option which estimates the movement of grid nodes (block corners) in two or three dimensions. For these quantities K-direction grid node displacements are averaged to obtain the vertical displacement of each block centre. Note that local strain calculations are performed also for zero-porosity and geomechanics-only blocks. Under suitable conditions this displacement will correspond roughly to the porosity-based calculation described above. The relationship between **VERDSPLGEO** and **SUBSIDGEO** is the same as for the porosity-based quantities.

**VDISPL** is the three-dimensional displacement vector of the block centre, derived from strain calculations of the **\*GEOMECH** option and split into X, Y and Z directions. It applies also to zero-porosity and geomechanics-only blocks. Displacement is relative to the grid origin and axes directions so its sign will depend on **\*KDIR**. **VDISPL** in the Z (vertical) direction corresponds to **VERDSPLGEO** for **\*KDIR \*UP** and to **SUBSIDGEO** for **\*KDIR \*DOWN**. No special history is available for **VDISPL** but **VERDSPLGEO** or **SUBSIDGEO** can be used for vertical displacement.

### Energy Balance

The following quantities are useful in the detailed analysis of energy dynamics, for example, at the steam/oil interface of a SAGD chamber.

ENINPLRAT is the energy accumulation term (F2.10 in STARS Appendix F) divided by gross block volume ( $V$  in F2.10). As such it is a specific (per gross volume) rate of change of energy accumulation with units  $J/day\cdot m^3$  or  $Btu/day\cdot ft^3$ . It is per-gross-volume so that a useful comparison can be made between blocks of different gross volume.

ENCONVRAT is the convective energy flow terms ( $v$ -terms in F2.12 in STARS Appendix F) divided by gross block volume. As such it is a specific net convective flow rate of energy in the same units as ENINPLRAT. Note that a single flow, say  $f_{12}$ , is between two blocks, say  $b_1$  and  $b_2$ . The contribution of  $f_{12}$  to  $b_1$  is  $f_{12}/V_1$  while the contribution of  $f_{12}$  to  $b_2$  is  $f_{12}/V_2$ , which is different if  $V_1$  (gross volume of  $b_1$ ) is not equal to  $V_2$ . Therefore, this statistic is most useful for comparison with other types of energy terms in the same block.

ENCOND RAT is the conductive energy flow term (T-term in F2.12 in STARS Appendix F) divided by gross block volume. As such it is a specific net conductive flow rate of energy in the same units as ENINPLRAT. The above comments involving  $f_{12}$ ,  $b_1$  and  $b_2$  apply here as well.

ENREACRAT is assigned zero.

### **Electrical Heating Vector Plots**

Several **SRF\_GRID** items cause electrical current density and current to be written to the SR2 which allows RESULTS to generate the corresponding vector plot. Current density is more useful since it is independent of block size. Each of these keywords results in writing of three grid-length arrays to the SR2, one for each direction and each of which is available for normal viewing. Note that the direction of current density may be different from that of potential gradient when the bulk electrical conductivity is non-isotropic.

### **Natural Fracture Permeabilities**

For natural fracture grid options the absolute permeabilities can be quoted as either “intrinsic” or “effective” (see Appendix E.8, especially equations E8.11 and E8.12). Consistent output of permeabilities is complicated by the fact that the input values are “effective” for fracture and “intrinsic” for matrix. Therefore, there are three sets of keywords for dumping absolute permeabilities:

1. \*PERMI, etc., which mimic the input (“effective for fracture, “intrinsic” for matrix);
2. \*PERMEFFI, etc., which dump “effective” values for both matrix and fracture;
3. \*PERMINTI, etc., which dump “intrinsic” values for both matrix and fracture.

### **Steam Quality Plots**

There are several different quantities called *steam quality* corresponding to different data sources and contexts.

<b>Keyword</b>	<b>Usage</b>
STEAMQUAL	*OUTSRF *GRID STEAMQUAL
QUALBLK	*OUTSRF *SPECIAL <i>special_his</i> QUALBLK <i>special_his</i> = BLOCKVAR   MAXVAR   MINVAR   AVGVAR
STMQUAL	*OUTSRF *SPECIAL STMQUAL <i>well</i>
STQUAL	*OUTSRF *SPECIAL PHWELL <i>well</i> STQUAL ...
STQUAL	*OUTSRF *SPECIAL WELL_LAYER <i>well uba</i> STQUAL

Generally *steam quality* refers to mass fraction of water vapour in a water gas/liquid system. In a *flowing* context it is the mass fraction of flowing water vapour in a flowing water gas/liquid system and so will depend upon the relative mobility of the phases. Each of the above quantities except STEAMQUAL is in a flowing context. The well-based quantities STMQUAL and STQUAL are in a source-sink context where flow (and not accumulation) is modelled.

QUALBLK gives the flowing water mass fraction for current conditions in grid cells, whereas STMQUAL is the specified injection value. QUALBLK in the injection cell seldom matches STMQUAL exactly, although it can be close after conditions reach a pseudo-steady state. Varying pressure can cause varying steam quality when energy is conserved. This is seen even along an isolated 1-dimensional cell row (e.g., discretized wellbore tubing with no heat loss). The value of QUALBLK for a cell will reflect the value for fluid flowing from that cell to a downstream cell.

STEAMQUAL applies the mass fraction concept to the *in-place* context and so will depend upon the relative saturation (volume fraction) of the phases. Because gas phase is usually much more mobile than liquid phases, STEAMQUAL is usually much less than QUALBLK for the same conditions.

## Grid Printout Orientation (Optional)

\*PRNTORIEN, \*PRINT\_REF

### PURPOSE:

\*PRNTORIEN overrides the default grid printout orientation.

\*PRINT\_REF defeats the printing of refined grid printout.

### FORMAT:

\*PRNTORIEN irotat ijkord  
\*PRINT\_REF ( \*ON | \*OFF )

### DEFINITIONS:

#### irotat

Axis rotation flag for printing out grid variables. The allowed range is 0 to 6.

Effect of irotat is:

irotat	rows	columns	planes
0		(most compact printout)	
1	I	J	K
2	I	K	J
3	K	I	J
4	K	J	I
5	J	I	K
6	J	K	I

#### ijkord

Axis reversal flag for printing out grid variables. The allowed range is 0 to 8.

Effect of ijkord is:

ijkord	rows	columns	planes
0		(bottom layer at bottom of page)	
1	normal	normal	normal
2	normal	normal	reversed
3	normal	reversed	normal
4	normal	reversed	reversed
5	reversed	normal	normal
6	reversed	normal	reversed
7	reversed	reversed	normal
8	reversed	reversed	reversed

#### \*PRINT\_REF

Allows the user to enable and disable the printing of values requested via

\*OUTPRN \*GRID for the fine grids as well as the fundamental grid values.

\*ON enables the printing, and \*OFF disable it.

## **DEFAULT:**

If \*PRNTORIEN is absent then \*PRNTORIEN 0 0 is assumed, giving the most compact printout with the bottom of the reservoir toward the page bottom.

If \*PRINT\_REF is absent then \*PRINT\_REF \*ON is assumed. If \*PRINT\_REF is present but neither \*ON nor \*OFF follow it, then \*PRINT\_REF \*ON is assumed.

## **EXPLANATION:**

When grid variables are printed in the output, axes for up to three dimensions are required. One axis direction is along the horizontal rows. Another axis direction is along the vertical columns. The remaining axis is printed as planes containing the rows and columns of the first two axes. Normally, the I, J and K axes are oriented on the printout to give the most compact result. In some cases it may be necessary or desired to change this default printout orientation using 'irotat'.

Normally, ordering along the rows is with the axis indices increasing from left to right; ordering along the columns and planes is with the axis indices increasing down the page. In some cases it may be necessary to reverse the order for one or more of the axes using 'ijkord'.

HINT: \*PRNTORIEN 1 1 causes grid arrays to be written to the output file in "standard" order, corresponding to the \*ALL grid array input option. To use output of one run as input for another, use \*PRNTORIEN 1 1, copy and paste the desired data to the new data file, and delete the "K=", "J=" and "I=" axis annotations.

---

## **Matrix Solver Printout (Optional)**

**\*OUTSOLVR**

### **PURPOSE:**

\*OUTSOLVR controls printout of detailed results from the matrix solving package AIMSOL.

### **FORMAT:**

\*OUTSOLVR ( \*ON | \*OFF )

### **DEFAULT:**

The default is \*OUTSOLVR \*OFF.

### **EXPLANATION:**

The actual number and dimension value of many pertinent quantities such as interblock connections used by the simulator will be shown. These numbers may be used to create common storage that will optimize the use of available storage capacity. See the tutorial **Optimizing Memory Requirements**.

Also printed are details of the residual reduction iterations taken by the linear solver.

---

## Trap Control-C Interrupt (Optional)

\*INTERRUPT

### PURPOSE:

Specify the action taken when an interrupt signal is detected.

### FORMAT:

\*INTERRUPT ( \*INTERACTIVE | \*STOP | \*RESTART-STOP | \*WRST-CONT )

### DEFINITIONS:

#### \*INTERACTIVE

Prompt the user interactively for instructions. Choices correspond to the remaining subkeywords.

#### \*STOP

Terminate the simulation run immediately. The current timestep is not completed but the output files are closed to prevent file corruption.

#### \*RESTART-STOP

Complete the current timestep, write all output specified by \*OUTPRN and \*OUTSRF, write a restart record and stop the run.

#### \*WRST-CONT

Complete the current timestep, write all output specified by \*OUTPRN and \*OUTSRF, write a restart record (if restart writing frequency is not zero) and continue the run.

### DEFAULTS:

If \*INTERRUPT is absent, \*INTERRUPT \*INTERACTIVE is assumed.

If \*INTERRUPT is present but none of the above subkeywords is present, subkeyword \*INTERACTIVE is assumed.

### EXPLANATION:

An interrupt signal can be sent to a running STARS program in two ways:

1. Typing "control" and "c" together will interrupt the current interactive process: on UNIX it will interrupt immediately; on Windows the interruption occurs after the current timestep is finished.
2. The UNIX command "kill -2 pid" will interrupt the process with ID "pid" (usually in background).

Interrupt handling is used to ensure that files are closed normally when a user aborts a run. Some platforms do not flush output file buffers upon an interrupt signal, so without interrupt handling some of the output would be lost.

---

## **Grid Array Data Echo Control (Optional)**

**\*NOLISTLIM**

### **PURPOSE:**

Control the detailed echoing of input data.

### **FORMAT:**

**\*NOLISTLIM**

### **DEFINITIONS:**

**\*NOLISTLIM**

Removes limit on number of data lines echoed for each grid array keyword.

### **DEFAULTS:**

If \*NOLISTLIM is absent, then grid array keyword data is limited to 20 echoed lines.

### **EXPLANATION:**

Data lines from the input data file are copied, or echoed, to the text output file as they are read (if keyword \*NOLIST is absent). For field-scale grids the number of data lines associated with grid and reservoir definition can be very large, especially for corner point grids and properties generated by other software packages (e.g., from maps). In addition, this echoed data is not needed after it has been debugged.

In order to keep the text output file to a reasonable size, the default action is to limit to 20 the number of echoed data lines per grid-based input data. For example, porosity keyword \*POR would have at most 20 lines echoed. This limit applies to each keyword separately.

Keyword \*NOLISTLIM allows you to defeat this limiting of echoed data lines. It is recommended that this keyword be used only for debugging data, and that it be removed for production runs.

## Convert Velocities to Shear Rates (Optional)

\*SHEAR\_FAC

### PURPOSE:

Convert fluid Darcy velocities to report equivalent porous media shear rates.

### FORMAT:

\*SHEAR\_FAC *factor*

### DEFINITIONS:

*factor*

Factor  $\gamma_{\text{fac}}$  in shear rate equation described below.

### DEFAULTS:

If \*SHEAR\_FAC is absent then  $\gamma_{\text{fac}} = 4.8$  is assumed (See Appendix D.20 Cannella<sup>1</sup>).

### EXPLANATION:

Keywords \*SHEARTAB, \*SHEARTHIN, and \*SHEARTHICK employ an optional velocity-dependent viscosity model. The Darcy phase velocities used in these calculations can be displayed in the output via \*OUTSRF \*GRID subkeywords \*VISCVELW, etc., in the SRF\_GRID list. In addition, subkeywords \*SHEARW, etc., make available equivalent shear rates which are converted from the fluid Darcy velocities via the equations below. This conversion depends upon factor  $\gamma_{\text{fac}}$  which the user can adjust via \*SHEARFAC.

The equation relating the effective porous media shear rate and the fluid Darcy velocity is

$$\dot{\gamma} = \frac{\gamma_{\text{fac}} |u_1|}{\sqrt{k k_{r,l} \phi S_l}}$$

where  $k$  and  $\phi$  are the absolute permeability and porosity, and  $u_l$ ,  $k_{r,l}$ , and  $S_l$  are the phase Darcy velocity, relative permeability, and saturation.

The shear rate factor itself is given by

$$\gamma_{\text{fac}} = C \left[ \frac{3n+1}{4n} \right]^{\frac{n}{n-1}}$$

where  $n$  is the shear thinning power exponent and  $C$  is a constant value, usually equal to 6. The default value of 4.8 corresponds to  $C = 6$  and  $n = 0.5$ .

A more detailed discussion on shear factors can be found in the references given in Appendix D.20.



# Reservoir Description

## Summary of Reservoir Description Data

This section contains data describing the basic reservoir definition and the simulation grid used to represent it. These data can be classified into the following groups:

1. Simulation Grid and Grid Refinement Options
2. Choice of Natural Fracture Reservoir Options
3. Well Discretization Option
4. Basis Reservoir Rock Properties
5. Sector Options

### Grid Options

STARS supports the following grid types:

- a) Finite-Difference (FD) Grid
  - i) Cartesian
  - ii) Radial
  - iii) Variable depth/thickness
- b) Corner Point

For the FD option, the following keywords are required:

*GRID	Grid type, should be followed by *CART or *RADIAL
*DI	Grid block dimension in the I direction.
*DJ	Grid block dimension in the J direction.
*DK	Grid block dimension in the K direction, value for each grid block is specified if variable thickness grid is used.

Optional keywords are:

*NINEPOINT	9-point option.
*DIP	Specify the dip angles in the I and J direction.
*REFINE	Using refine grid options.
*VAMOD	Volume and area modifier option.
*NULL	For specifying null blocks.

When variable depth grid is used, \*DTOP is required:

\*DTOP      Depth to the top of each grid column.

In this case the dip angles are assumed to be zero. \*DTOP can be used in conjunction with the fluid initialization keywords \*DWOC, \*DGOC and \*REFDEPTH.

### **Fractured Reservoir Option**

The options available for fractured reservoir simulation are specified by one of the following keywords:

\*DUALPOR \*DUALPERM \*MINC \*SUBDOMAIN

The fracture spacing in each of the coordinate directions are specified with the keywords:

\*DIFRAC \*DJFRAC \*DKFRAC

### **Discretized Wellbore Option**

This option models the wellbore with grid blocks whose equations are solved simultaneously with the reservoir flow equations. This option is invoked by the \*WELLBORE keyword. A circulating well can be specified with the subkeyword \*CIRCWELL such that both the tubing and the annulus are discretized.

### **Discretized Wellbore in Hybrid Grid**

In order to model effectively the single-well SAGD (Steam Assisted Gravity Drainage) process, the wellbore needs to be connected directly to blocks above it and below it, allowing steam to rise and liquid to migrate in from the bottom at the same time. This is accomplished by embedding the discretized wellbore inside a hybrid-refined grid. See the detailed explanation for \*WELLBORE.

### **Rock Properties**

The porosity, permeability, and transmissibility modifiers of the reservoir are specified by the keywords:

\*POR

\*PERMI, \*PERMJ, \*PERMK

\*TRANSI, \*TRANSJ, \*TRANSK, \*TRANLI, \*TRANLJ, \*TRANLK, \*TRANSMF

The qualifier \*MATRIX and \*FRACTURE are used to distinguish between matrix and fracture properties for the fractured reservoir options. The qualifier \*RG refers to refined grid blocks when the \*REFINE, \*MINC, \*SUBDOMAIN and \*WELLBORE options are used.

### **Corner Point Grid**

There are several ways to define a corner point grid, some using the same grid-size keywords as the Cartesian grid. See the descriptions for keywords \*ZCORN, \*XCORN, \*YCORN, \*COORD and \*CORNERS. Note that for a corner point grid \*KDIR \*DOWN is assumed and \*UP is not available.

## **Zero-Porosity Blocks**

STARS has two kinds of blocks with no porosity:

1. “Null” block for which no equations are solved. This block type may be specified via keyword \*NULL or \*VATYPE. In isothermal mode, this block type is specified also via zero porosity or permeability values. For example, it is common in isothermal mode to model a shale barrier between pay zones as a layer of null blocks or simply as a fluid transmissibility barrier between adjacent pay-zone grid layers. In thermal mode, null blocks commonly result from the definition of a symmetry element of a repeating pattern, e.g., one-eighth nine-spot. Each null block requires almost no array storage. No fluid properties are reported for a null block.
2. “Heat” block which may contain and conduct heat, and for which only the energy equation is solved. This block type is available only in thermal mode and is specified via zero porosity or permeability values. For example, a shale barrier may store heat as well as conduct heat between pay zones. Compared to the isothermal treatment, in thermal mode this method is required when the thickness of the shale layer is not small compared to the adjacent pay zone grid layers. Each heat block requires as much array storage as a fully active block. Only heat and temperature related fluid properties are reported for a “heat” block.

In general, STARS treats zero-porosity grid data the same as IMEX and GEM only when STARS is run in isothermal mode. When STARS is run in thermal mode (the default), then care must be taken to account for the intention of the original data. Note that the most common way to specify null blocks in the original data (zero porosity and permeabilities, especially from map-reading software) will translate by default in STARS thermal mode to “heat” blocks. This is the most accurate but most expensive treatment, so the user may need to change the ported grid data to force these blocks to be “null”.

## **Aquifer Models**

Aquifer water-influx models may be defined, based on Carter-Tracy or Fetkovitch treatments as in IMEX or based on the previously available semi-analytical method. Keyword data from IMEX may be ported directly. Only one keyword is needed to add thermal treatment to any water influx method.

## **Porting Grid Data from IMEX and GEM**

Most grid-definition data can be ported directly from the CMG simulators IMEX and GEM. However, IMEX and GEM may support some keywords in their Reservoir Description sections (e.g., lease options) that are not supported by STARS.

STARS supports some keywords in its Reservoir Description section that are not supported by IMEX or GEM:

- \*VAMOD, VATYPE (geometry modifiers)
- \*WELLBORE (discretized wellbore)
- \*NINEPOINT, \*NINEPTH

---

## Grid Type

\*GRID, \*KDIR

### PURPOSE:

\*GRID indicates the beginning of input of the reservoir conditions.

### FORMAT:

```
*GRID ( *CART | *VARI | *CORNER ) ni nj nk  
*GRID *RADIAL ni nj nk ( *RW rw )  
*KDIR ( *UP | *DOWN )
```

### DEFINITIONS:

#### \*CART

Keyword indicating rectangular Cartesian grid.

#### \*VARI

Keyword indicating a rectangular grid allowing variable depth/variable thickness layers.

#### \*CORNER

Keyword indicating a corner point grid, as described below. \*KDIR \*UP is not available with \*CORNER.

ni

Number of grid blocks in I direction. For \*RADIAL grids, ni must be greater than 1.

nj

Number of grid blocks in J direction.

nk

Number of grid blocks in K direction.

#### \*RADIAL

Keyword indicating radial-angular cylindrical grid.

#### \*RW rw

Specifies the radius (m | ft | cm) of the innermost block boundary; the radial blocks will start this far from the grid center. The value 0 is allowed.

#### \*UP

Indicates that the K direction points UP, putting layer 1 at the bottom of the reservoir. Not available with \*GRID \*CORNER.

#### \*DOWN

Indicates that the K direction points DOWN, putting layer 1 at the top of the reservoir.

## **DEFAULTS:**

If \*RW is absent after \*GRID \*RADIAL, then a radius of 8.6 cm is assumed.

If \*KDIR is absent, \*KDIR \*UP is assumed (except for \*CORNER).

## **CONDITIONS:**

\*GRID is a required keyword, and must be the first keyword in the RESERVOIR DESCRIPTION section. A radial grid requires ni > 1.

## **EXPLANATION:**

\*GRID defines the grid type and the number of fundamental grid blocks within this system.

Examples:

- a) Rectangular Cartesian grid with ten blocks in the "x" direction, five blocks in the "y" direction, and four blocks in the "z" direction (ni=10, nj=5, nk=4). Enter:

\*GRID \*CART 10 5 4

- b) Cylindrical grid with fifteen blocks in the radial direction, three block in the theta direction, and five blocks in the vertical direction (ni=15, nj=3, nk=5)

\*GRID \*RADIAL 15 3 5

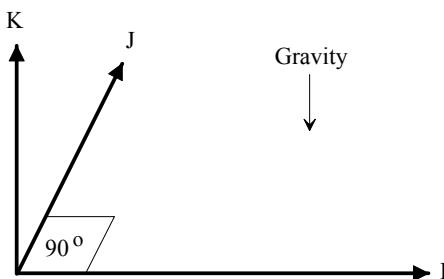
The innermost radial block of a radial grid is not discretized in the angular direction. In the example above, the radial block i = 1 has only one angular subdivision j = 1 which is connected to all three angular blocks for the next outer radial row i = 2. This means that i = 1 has only j = 1, but i = 2 to 15 has j = 1, 2 and 3. The result is that the center well is connected to only one block for each k layer.

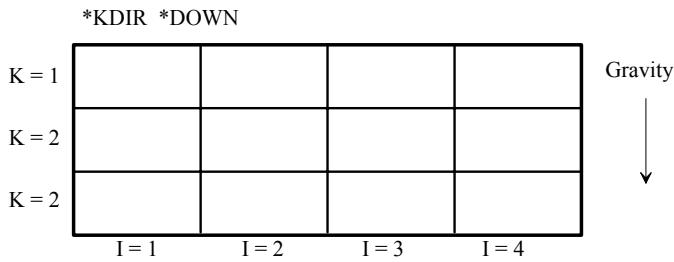
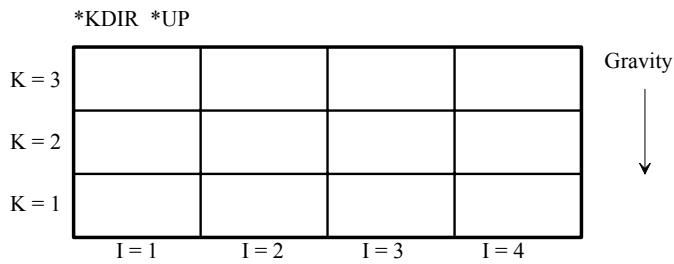
I, J, and K are used to indicate directions regardless of the grid type used.

<b>Grid Type</b>	<b>I</b>	<b>J</b>	<b>K</b>
*CART	x	y	z
*VARI	x	y	z
*CORNER	x	y	z
*RADIAL	r	theta	z

The K index can be made to increase downward or upward by using the \*KDIR keyword. The grid can be tilted by use of the \*DIP keyword.

An untilted \*UP coordinate system appears as:



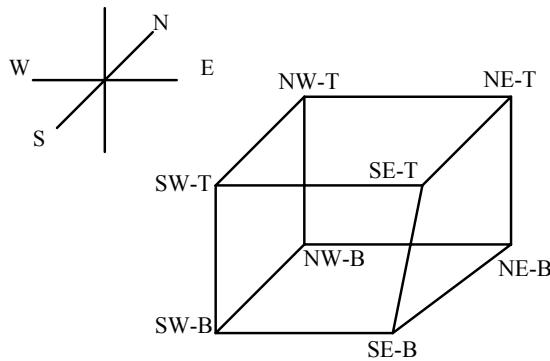


### Corner Point:

Corner point grids are made up of blocks each defined by their eight corner points. Each corner point is described by giving its three coordinates: an "x"-, "y"- and "z"-coordinate, which gives its location in the reservoir. The "x"- and "y"- coordinates are to be measured with respect to a horizontal reference surface, while the "z"- coordinate is to be the depth of the corner point measured downwards from the surface. Both positive and negative depths are valid, depending on the location of the reference surface with respect to the reservoir, although positive values are most common.

Thus, it takes  $3 \times 8 = 24$  numerical values to determine a general corner point block. The simulator places restrictions on the corner point data however, so that it will not be necessary to read  $24 \times ni \times nj \times nk$  values to define the grid for all cases. Details follow later.

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 made by connecting the corner points with line segments as shown.

The simulator requires that the 8 corner points (each with its three coordinates) can be arranged and labeled so that:

1. The "x"-coordinate difference from NW-T to NE-T is positive and the same as that from SW-T to SE-T;
2. The "y"-coordinate difference from NW-T to SW-T is positive and the same as that from NE-T to SE-T;
3. The "-B" points should lie directly below the "-T" points; that is, each "-T" and "-B" pair should have the same "x"- and "y"- coordinates, and the "-B" points should have the larger "z"-coordinate.

Thus, the corner point block's four side faces should be planar and the block should appear rectangular when viewed from the top (or bottom). The top and bottom faces are defined with a nonlinear (bilinear) interpolation and will not be planar in general.

When multiple blocks are defined using a corner point grid, the simulator requires that the grid must appear Cartesian when viewed from above (or below). Thus, it is required that the following hold for all valid indices I, J, K:

4. The NE-T corner point of block (I,J,K) and the NW-T corner point of block (I+1,J,K) must lie in the same vertical line, and similarly for the NE-B and NW-B corners, the SE-T and SW-T corners, and the SE-B and SW-B corners for blocks (I,J,K) and (I+1,J,K), respectively;
5. The SW-T corner point of block (I,J,K) and the NW-T corner point of block (I,J+1,K) must lie in the same vertical line, and similarly for the SW-B and NW-B corners, the SE-T and NE-T corners, and the SE-B and NE-B corners for blocks (I,J,K) and (I,J+1,K), respectively;

Note that vertical faulting is permitted, as (4) and (5) above only require that the corner points lie in the same vertical line, and not that they be the same points. (No faulting would occur if the phrase "must lie in the same vertical line" was replaced by "are the same" everywhere in conditions (4) and (5).) Faulting implies that partial face overlap is allowed.

Finally, the simulator requires that block tops should not cross through the bottoms of their vertical neighbours:

6. Each "-B" corner point of block (I,J,K) should not be deeper than the corresponding "-T" corner point of block (I,J,K+1).

The simulator requires actual contact of block faces before it will allow inter-block fluid flow on corner point grids. Nodes for corner point blocks are placed at their barycentre.

Example:

- a) Corner point grid with 20 blocks in the "x" direction, 20 blocks in the "y" direction, and 5 layers ( $ni=20$ ,  $nj=20$ ,  $nk=5$ ). Enter:

```
*GRID *CORNER 20 20 5
```

---

## Convert Cartesian Grid to Corner Point (Optional)

**\*CONVERT-TO-CORNER-POINT**

### PURPOSE:

Internally convert a Cartesian grid type to corner-point type.

### FORMAT:

**\*CONVERT-TO-CORNER-POINT**

### DEFAULTS:

If this keyword is absent then no grid type conversion is done.

### CONDITIONS:

This keyword converts only Cartesian grid types to corner-point type.

The option is not allowed when \*KDIR \*UP is used together with any natural fracture option.

This option cannot be used together with other grid modification keywords like  
**\*PINCHOUTARRAY**.

### EXPLANATION:

In a grid of type \*VARI it is possible that the corners of adjacent blocks do not coincide. This condition can occur in Cartesian-based \*VARI type grids that are known generally as variable-thickness and variable-depth.

Keyword \*CONVERT-TO-CORNER-POINT converts type \*VARI grid data with this condition to a corner-point grid that does not have this condition. The conversion is performed entirely at run time and does not appear in the data file or simulator data echo. Each new single corner location is simply the average of the previous different corner locations. Volumes and transmissibilities of individual grid blocks will differ from the previous grid, but fractional changes should be reasonable for a well-formed grid. Global quantities like total pore volume should be little different.

More extreme variable depth and thickness situations may not convert satisfactorily, in which case some manual adjustment of the original data is recommended. In addition, this conversion does not preserve the deliberate modelling of faults. In all cases, you can view both grid types in Results using data sets with and without this keyword.

---

## Nine-Point Spatial Discretization (Optional)

\*NINEPOINT, \*NINEPTH

### PURPOSE:

\*NINEPOINT controls the nine-point spatial discretization option.

### FORMAT:

```
*NINEPOINT ( *OFF | *IJ | *IK )  
*NINEPTH
```

### DEFINITIONS:

#### \*OFF

Five-point discretization is used in all three planes.

#### \*IJ

Nine-point discretization is used for the I-J plane, and five-point discretization is used for the J-K and I-K planes. This option is not available for \*GRID option \*RADIAL.

#### \*IK

Nine-point discretization is used for the I-K plane, and five-point discretization is used for the J-K and I-J planes.

#### \*NINEPTH

Nine-point discretization is used for thermal conduction calculations in the same plane as defined by NINEPOINT (I-J or I-K).

### DEFAULTS:

If \*NINEPOINT is absent then \*NINEPOINT \*OFF is assumed.

If \*NINEPTH is absent then five-point discretization is applied to thermal conduction.

### CONDITIONS:

\*NINEPTH can be used only if \*NINEPOINT is also used.

The \*NINEPOINT option may not be used together with \*REFINE or natural fracture grid options.

Sub-option \*IJ may not be used with \*GRID \*RADIAL.

Sub-option \*IK may not be used together with block pinch-outs.

### EXPLANATION:

See Appendix E.4 for discussion of grid orientation.

## Nine-Point Method

The transmissibilities in the nine-point finite difference approximation are calculated according to the Amoco method (SPE 16975, 1991). The following points compare the new method with the previously used method (Coats and Modine, SPE 12248):

1. Data sets with isotropic and uniform permeabilities and block sizes in the nine-point plane experience no change in results (e.g., Test Bed sttst07.dat) or CPU.
2. Modest variations from uniform and isotropic usually result in acceptably small differences in transmissibilities from the previous method. However, severely different block sizes and/or permeabilities can give significantly different local results. The new method is more likely to give an unphysical result for severely non-uniform or non-isotropic data, so such data should be used with caution.
3. The new method allows separation of the geometrical factors of the nine-point transmissibilities from the appropriate property. Therefore, processes in which that property changes with time can be discretized with the new nine-point scheme, unlike the previous method. The two processes of interest here are dilation, where the permeability changes with time in response to porosity changes, and thermal conduction, where thermal conductivity depends on current saturations and temperature.

## Thermal Conduction

The \*NINEPTH option carries a CPU penalty. The \*NINEPTH option is necessary only for processes that are dominated by thermal conductivity, such as experiments at the laboratory scale and detailed near-well thermal studies. For most field scale simulations, convection is the main heat transport mechanism and so \*NINEPTH is not needed.

## Pseudo 1-D Modelling

The \*NINEPOINT option will give unexpected results if the grid is not constructed properly when modelling a pseudo-1D problem with a 2-D grid. The two reservoir boundaries that are parallel to the pseudo-1D direction must have their nodes on the reservoir boundary. In other words, the reservoir must be treated like a repeated pattern. Use the geometry modifiers \*VAMOD to trim these boundary blocks in half so that the block nodes fall on the reservoir boundary.

## Corner-point Grids

The nine-point formulation is based on orthogonal grid assumptions that are built into \*GRID options \*CART (IJ and IK planes) and \*RADIAL (IK plane). On the other hand, corner-point grids generally are non-orthogonal and so violate nine-point assumptions. However, the target plane of a corner-point grid may in fact be orthogonal or only slightly non-orthogonal, in which case the nine-point option can be used with some confidence. Care should be taken that the plane to which 9-point is applied is not excessively non-orthogonal. For example, it is safe to use \*NINEPOINT \*IJ with a corner-point grid that is non-orthogonal only in the K direction, that is, the grid in the IJ plane looks Cartesian.

---

## Block Dimensions for I Direction (Required)

\*DI

### PURPOSE:

\*DI signals input of an array of grid block lengths for I direction. For cylindrical systems, it indicates the input of R direction block lengths.

### ARRAY:

\*DI

### DEFAULTS:

Conditional keyword. No defaults.

### CONDITIONS:

This keyword is required for all grid types except \*GRID \*CORNER.

This keyword may be used with \*GRID \*CORNER - see \*ZCORN.

All array reading options are valid; however, all blocks with the same I index must have the same block length.

### EXPLANATION:

In the following,  $ni$ ,  $nj$  and  $nk$  are from the grid-type data line (keyword \*GRID).

The keyword \*DI defines the dimensions of the grid blocks in the I direction. The unit is (m | ft | cm). The most commonly used array reading subkeywords are \*IVAR and \*CON. You may specify non-uniform block sizes in a locally refined grid.

The acceptable range of values for block lengths in the I direction is  $10^{-4}$  m ( $3.23 \cdot 10^{-4}$  ft) to  $10^9$  m ( $3.28 \cdot 10^9$  ft).

Examples:

- a) When I-direction grid-block sizes for a problem with  $ni=10$  are  
1000,1000,1500,400,400,400,400,1000 and 1000, use  
\*DI \*IVAR 2\*1000 1500 5\*400 2\*1000
- b) When I-direction size of each grid block is 1200, use  
\*DI \*CON 1200
- 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
- d) When radial widths of  $ni=10$  radial-angular cylindrical blocks are 2.00, 2.32, 5.01, 10.84, 23.40, 50.55, 109.21, 235.92, 509.68 and 1101.08, use  
\*DI IVAR  
2.00 2.32 5.01 10.84 23.40  
50.55 109.21 235.92 509.68 1101.08

Note that the inner radius of block I=1 is the value specified by \*RW on the \*GRID \*RADIAL data input line.

## **Pseudo-Infinite Blocks**

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

---

## Block Dimensions for J Direction (Required)

\*DJ

### PURPOSE:

Specify J-direction grid block size (length for Cartesian grid, angle for cylindrical grid).

### ARRAY:

\*DJ

### DEFAULTS:

Conditional keyword. No defaults.

### CONDITIONS:

This keyword is required for all grid types except \*GRID \*CORNER.

This keyword may be used with \*GRID \*CORNER - see \*ZCORN.

All array reading options are valid; however, all blocks with the same J index must have the same block length.

### EXPLANATION:

In the following,  $ni$ ,  $nj$  and  $nk$  are from the grid-type data line (keyword \*GRID).

For all grid types except cylindrical, keyword \*DJ specifies grid-block dimensions in the J direction in unit (m | ft | cm). The acceptable range of values for block lengths in the J direction is  $10^{-4}$  m ( $3.23 \cdot 10^{-4}$  ft) to  $10^9$  m ( $3.28 \cdot 10^9$  ft).

For cylindrical R-0-Z grid type (\*GRID \*RADIAL), keyword \*DJ specifies grid-block dimensions in the  $\theta$  direction in unit of degrees. The sum of the J-direction dimensions must no exceed 360 degrees. When  $nj > 1$  and the sum of the J-direction dimensions is 360 degrees (full circle), block (I,1,K) is connected to block (I, $nj$ ,K), I=2: $ni$ , K=1: $nk$ . Note that the innermost blocks (I=1) have no  $\theta$  subdivisions.

The most commonly used array reading subkeywords are \*JVAR and \*CON. You may specify non-uniform block sizes in a locally refined grid.

Examples:

- a) When J-direction grid-block sizes for a problem with  $nj=10$  are 2000,2000,2500,4000,1500,1500,400,400,1000 and 1000, use  
\*DJ \*JVAR 2\*2000 2500 4000 2\*1500 2\*400 2\*1000
- b) When J-direction size of each grid block is 2200, use  
\*DJ \*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  
\*DJ \*CON 10  
\*DJ \*RG 3 5 8 \*JVAR 4.0 2.0 4.0
- d) When a cylindrical grid with  $nj=6$  has uniform sectors in a full circle, use  
\*DJ \*CON 60

e) The following is a full-circle cylindrical grid with  $nj=6$  non-uniform sectors.

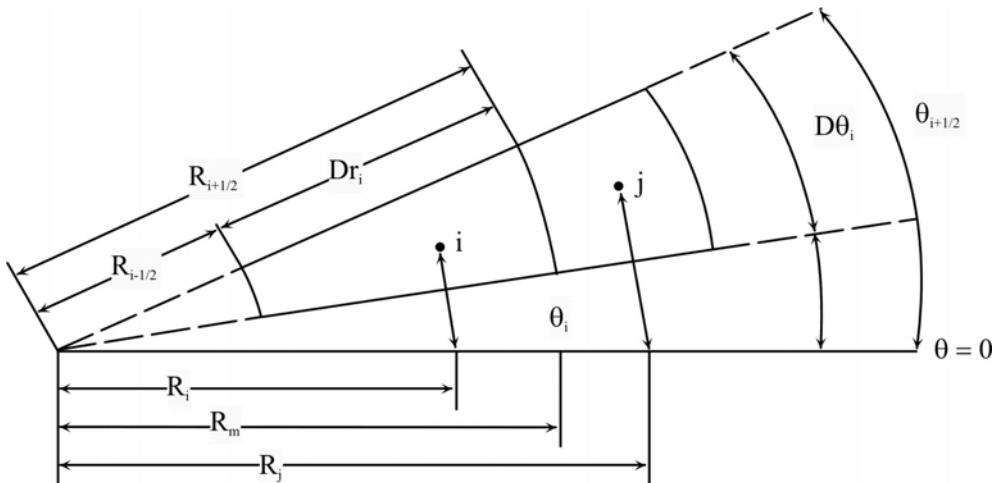
\*DJ \*JVAR 90 60 30 5 30 60 85

f) When a cylindrical grid with  $nj=6$  has uniform sectors in a quarter circle, use

\*DJ \*CON 15

### Radial Grid Geometry Details

For a description of how block sizes are used in the radial grid, see Figure 1 below. In the output echo, the J direction block size is the mid-block arc length. The product of the block sizes reported for each direction gives the correct block volume.



User enters block sizes  $DR_i$ ,  $D\theta_i$  (converted to radians) and  $DZ_i$

Internal calculations for block i:

- $R_{i-1/2}$  by summing  $DR$  of interior blocks and well radius
- radius of node at block centre is  $R_i = R_{i-1/2} + DR_i/2$
- $\theta_{i-1/2}$  and  $\theta_i$  in similar manner
- block size in  $\theta$  direction is arc length through block centre,  $R_i D\theta_i$
- block volume is  $R_i D\theta_i DR_i DZ_i$
- transmissibility between blocks i and j account for changing cross-sectional area from  $R_i$  to  $R_j$

*Figure 1: Block Dimensions in Cylindrical Coordinates*

### Pseudo-Infinite Blocks

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

---

## Block Dimensions for K Direction (Required)

\*DK

### PURPOSE:

\*DK signals input of array of grid block thicknesses in K direction.

### ARRAY:

\*DK

### DEFAULTS:

Conditional keyword. No defaults.

### CONDITIONS:

This keyword is required for all grid types except \*GRID \*CORNER.

This keyword may not be used with \*GRID \*CORNER - see \*ZCORN.

All array reading options are valid; however, all blocks with the same K index must have the same thickness for \*GRID \*CART and \*RADIAL. Different thicknesses in the same layer (K index) are allowed for \*GRID \*VARI only and not with \*DIP.

Blocks can be assigned a zero thickness if they are to be considered as pinched out. See the discussions for \*PINCHOUTARRAY and \*PINCHOUT-TOL.

### EXPLANATION:

This keyword defines the dimensions of the grid blocks in the K direction. The unit is (m | ft | cm). For cylindrical R-θ-Z grid type (\*GRID \*RADIAL), use \*DK to specify block sizes in the axial (Z) direction.

The acceptable range of values for block lengths in the K direction is  $10^{-4}$  m ( $3.23 \cdot 10^{-4}$  ft) to  $10^9$  m ( $3.28 \cdot 10^9$  ft).

Examples:

- a) When K-direction grid-block sizes for a problem where nk=8 are 20, 20, 25, 40, 15, 45, 45 and 45, use

\*DK \*KVAR 2\*20 25 40 15 3\*45

- b) When K-direction size of each grid block is 22, use

\*DK \*CON 22

- 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

\*DK \*CON 10

\*DK \*RG 3 5 8 \*KVAR 2.5 7.5

- d) The following specifies a variable thickness grid.

\*GRID \*VARI 5 4 2

\*DK \*ALL

20\*5.0 \*\* K=1 layer uniform  
6.0 6.1 6.2 6.3 6.4 \*\* J=1, K=2  
6.1 6.2 6.3 6.4 6.5 \*\* J=2, K=2  
6.2 6.3 6.4 6.5 6.6 \*\* J=3, K=2  
6.3 6.4 6.5 6.6 6.7 \*\* J=4, K=2

## **Pseudo-Infinite Blocks**

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

---

## Depth (Conditional)

\*DEPTH

### PURPOSE:

\*DEPTH indicates input of a reservoir depth for a single grid block. This depth is usually to be measured to the block's centre, unless \*TOP appears (see below).

### FORMAT:

\*DEPTH      (\*TOP)      i j k depth  
                  (\*CENTRE)

### DEFINITIONS:

\*TOP

Subkeyword indicating that the depth is to the top (centre of the top face) of the reference block.

\*CENTRE

Subkeyword indicating that the depth is to the centre of the reference block.

i

I direction index of the reference block.

j

J direction index of the reference block.

k

K direction index of the reference block.

depth

Depth to the centre (or top if \*TOP is used) of the reference block in the reservoir (m | ft). The value may be of any sign.

### DEFAULTS:

Conditional keyword. No defaults. \*CENTRE is assumed if \*TOP does not appear.

### CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. One of \*DEPTH, \*DTOP, \*DEPTH-TOP or \*PAYDEPTH must be specified for \*GRID \*CART, \*GRID \*VARI, or \*GRID \*RADIAL. This keyword should not be used with corner point grids. If depth modifications are required for corner point grids, the \*DEPTH-TOP or \*PAYDEPTH 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/top of the grid block. The value may be positive or negative depending on the location of the reference surface, although positive values are more common.

Some kind of depth information is required for all simulations.

When \*DEPTH is used, depths are assigned to all blocks based on the value provided. The calculation is made based on the blocks' thicknesses (\*DK keyword) and the dip angles provided by the \*DIP keyword (see \*DIP keyword description following).

The subkeyword \*CENTRE can be used if desired, although \*DEPTH defaults to centre.

Example:

```
*DEPTH 1 1 1 2000.0
```

The acceptable range of values for depth is:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>
min	-1.0E+4	-32,808.0
max	1.0E+4	32,808.0

## **Depth to the Tops of Grid Blocks (Conditional)**

**\*DTOP**

### **PURPOSE:**

\*DTOP specifies depth to the centre of the top face of each grid block in the top layer of the grid.

### **ARRAY:**

\*DTOP depth(1,1) ... depth(ni,1) depth(1,2) ... depth(ni,nj)

### **DEFAULTS:**

If \*DTOP is absent, then depth is obtained from \*DEPTH or \*PAYDEPTH. If \*DTOP, \*DEPTH and \*PAYDEPTH are absent, then depth to top of column (1,1) is zero.

### **CONDITIONS:**

This keyword must be in the RESERVOIR DESCRIPTION keyword group. One of \*DEPTH, \*DTOP, \*DEPTH-TOP or \*PAYDEPTH must be specified for \*GRID \*CART, \*GRID \*VARI, or \*GRID \*RADIAL. Use of this keyword, \*DEPTH-TOP or \*PAYDEPTH, is recommended for \*GRID \*VARI. \*DEPTH-TOP and \*PAYDEPTH, but not \*DTOP, can be used for corner point grids if depth modifications are required.

If this keyword is used with \*GRID \*CART, the values in the \*DTOP array should all be the same.

No array qualifiers or array reading options are permitted for this particular array keyword. A fixed number of values (ni \* nj) is always expected.

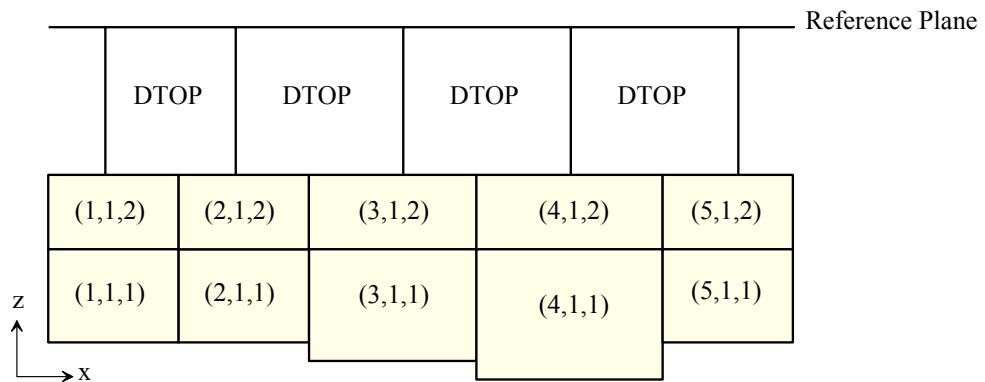
### **EXPLANATION:**

This keyword is usually used to define the depths of grid blocks for a variable depth/variable thickness grid (\*GRID \*VARI). A total of ni \* nj depth values must be entered. The unit is (m | ft | cm). The values are to be measured downwards from a horizontal reference surface to the centre of the tops of the grid blocks in the upper-most layer. The values may be positive or negative depending on the location of the reference surface. They are to be entered row by row with the I index changing fastest, and the J index slowest.

Note that the K index assumed for this array will be that of the uppermost layer; that is, it will be K = nk if \*KDIR does not appear in the data set, or if \*KDIR \*UP has been specified, or it will be K = 1 if \*KDIR \*DOWN appears.

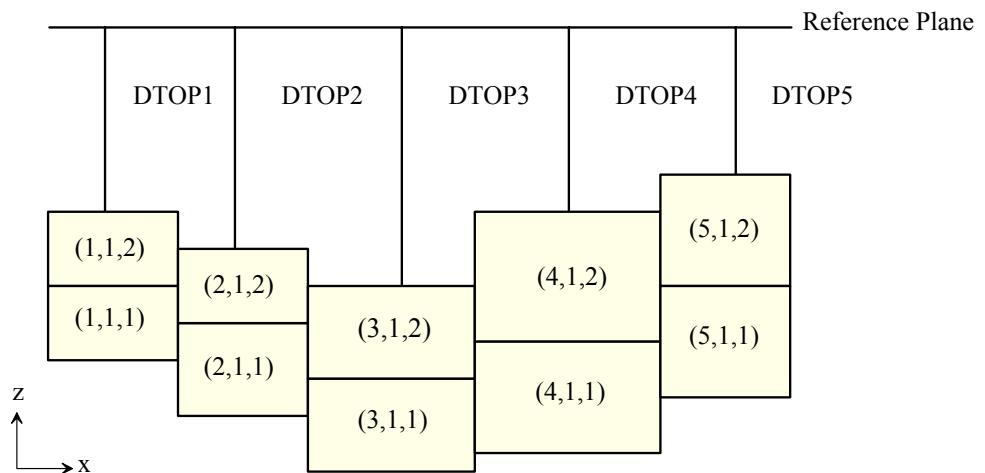
Depths are assigned to all blocks based on the depths provided by this keyword and the blocks' thicknesses (\*DK keyword).

See Figure 2 below.



- a) Depth is constant, but thickness of layer 1 varies. The data is

```
*GRID *CART 5 1 2
DI ...
DJ ...
DK ALL      10.5 10.5 15.6 21.3 10.5 5* 8.9
*DTOP 5*3600
```



- b) Both depths and thicknesses vary. The layers are the same as for (a). The data is

```
*GRID *CART 5 1 2
DI ...
DJ ...
DK ALL      10.5 10.5 15.6 21.3 10.5 5* 8.9
*DTOP      DTOP1 DTOP2 DTOP3 DTOP4 DTOP5
```

**Figure 2: Illustration of Variable Depth and Variable Thickness Options**

Example:

A variable depth/variable thickness grid with ni=6, nj=4 and nk=2 might use the following:

```
*DTOP
1000.0 1300.0 1250.0 1100.0 1200.0 1070.0
1070.0 1090.0 1080.0 1110.0 1120.0 1200.0
1000.0 1200.0 1110.0 1200.0 1200.0 1190.0
1070.0 1100.0 1100.0 1170.0 1070.0 1070.0
```

The acceptable range of values for depths is:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
min	-1.0E+4	-32,808.0	-1.0E+6
max	1.0E+4	32,808.0	1.0E+6

**NOTE:** Previous usage of \*DTOP with STARS allowed an extended syntax with array qualifiers \*CON and \*ALL. To maintain consistency with CMG pre-processors and other CMG simulators, it is recommended that data using the extended syntax be changed to conform to the standard syntax described above.

1. Qualifier \*ALL has the same operation as the standard syntax, and therefore keyword \*ALL can be removed with no effect.
2. \*DTOP \*CON followed by a single depth can be replaced with \*DEPTH \*TOP 1 k where k = 1 for \*KDIR \*DOWN and k = nk for \*KDIR \*UP.

## **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, \*DEPTH-TOP or \*PAYDEPTH must be specified for \*GRID \*CART, \*GRID \*VARI, or \*GRID \*RADIAL. Use of this keyword, or \*DTOP, is recommended for \*GRID \*VARI.

\*DEPTH-TOP or \*PAYDEPTH can be specified for corner point grids. These keywords will override depths calculated from the "z" components of the corner point locations.

If this keyword is used with \*GRID \*CART, the depths in each layer (blocks with the same K index) 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 ni \* nj \* nk values.

The values are to be measured downwards from a horizontal reference surface to the centre of the grid block, which is where the net pay is assumed to be positioned. The values may be positive or negative depending on the location of the reference surface in the reservoir, although positive values are most common. The unit is (m | ft | cm).

Since it is assumed that the (net) pay is centered in the block, the \*PAYDEPTH array's depths can be directly assigned to each block's node.

When used for corner point grids, the paydepth values override depths calculated from the "z" components of the corner point locations. If a \*PAYDEPTH value is not set for certain cells (as when \*PAYDEPTH is used with the \*IJK option and not all cells are touched) then depths for the remaining cells will revert to those taken from the "z" components. The actual corner point locations are not altered by \*DEPTH-TOP or \*PAYDEPTH, and grid visualizations are unaffected. Only the "Depth to Centers" array in the output echo (use \*OUTPRN \*RES \*ALL) shows the results of using \*DEPTH-TOP or \*PAYDEPTH with corner point grids. Use of \*DEPTH-TOP or \*PAYDEPTH with corner point grids works like a vertical position modifier for the cells.

Example:

A variable depth/variable thickness grid with ni=6, nj=4 and nk=2 might use the following:

```
*PAYDEPTH *ALL
1000.0 1300.0 1250.0 1100.0 1200.0 1070.0
1070.0 1090.0 1080.0 1110.0 1120.0 1200.0
1000.0 1200.0 1110.0 1200.0 1200.0 1190.0
1070.0 1100.0 1100.0 1170.0 1070.0 1070.0
2000.0 2300.0 2250.0 2100.0 2200.0 2070.0
2070.0 2090.0 2080.0 2110.0 2120.0 2200.0
2000.0 2200.0 2110.0 2200.0 2200.0 2190.0
2070.0 2100.0 2100.0 2170.0 2070.0 2070.0
```

The acceptable range of values for depths is:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
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 ni \* nj \* nk values.

The values are to be measured downwards from a horizontal reference surface to the top of the grid block. The values may be positive or negative depending on the location of the reference surface in the reservoir, although positive values are most common. The unit is (m | ft | cm).

When used for corner point grids, the depth to top values override depths calculated from the "z" components of the corner point locations. If a \*DEPTH-TOP value is not set for certain cells (as when \*DEPTH-TOP is used with the \*IJK option and not all cells are touched) then depths for the remaining cells will revert to those taken from the "z" components. The actual corner point locations are not altered by \*DEPTH-TOP, and grid visualizations are unaffected. Only the "Depth to Centers" array in the output echo (use \*OUTPRN \*RES \*ALL) shows the results of using \*DEPTH-TOP with corner point grids. Use of \*DEPTH-TOP with corner point grids works like a vertical position modifier for the cells.

Example:

A variable depth/variable thickness grid with ni=6, nj=4 and nk=2 might use the following:

```
*DEPTH-TOP *ALL
1000.0 1300.0 1250.0 1100.0 1200.0 1070.0
1070.0 1090.0 1080.0 1110.0 1120.0 1200.0
1000.0 1200.0 1110.0 1200.0 1200.0 1190.0
1070.0 1100.0 1100.0 1170.0 1070.0 1070.0
2000.0 2300.0 2250.0 2100.0 2200.0 2070.0
2070.0 2090.0 2080.0 2110.0 2120.0 2200.0
2000.0 2200.0 2110.0 2200.0 2200.0 2190.0
2070.0 2100.0 2100.0 2170.0 2070.0 2070.0
```

The acceptable range of values for depths is:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
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)

### DEFINITIONS:

idip

Tilt angle in degrees of the I axis above the horizontal. Allowed range is -90 to 90 degrees. See Figure 3 below.

jdip

Tilt angle in degrees of the J axis above the horizontal. Allowed range is -90 to 90 degrees. See Figure 3 below.

### DEFAULTS:

\*DIP 0 0

### CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. This keyword is optional with \*GRID \*CART and \*GRID \*RADIAL. If the variable-depth or variable-thickness option is specified, then idip and jdip are assumed to be zero.

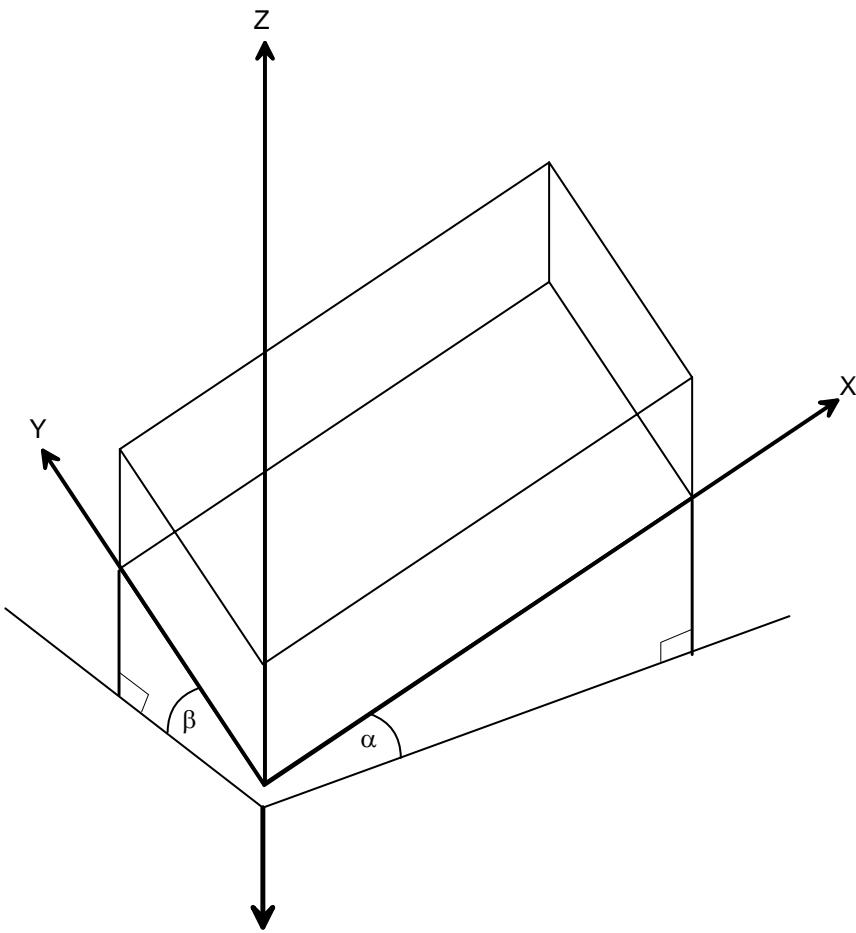
### EXPLANATION:

For radial grid systems, idip is the angle between the K axis and the vertical. The reference radial direction ( $\theta = 0$ ) lies in the plane defined by the K axis and the vertical direction. See Figure 4 below.

For a radial grid, modelling of gravity effects from tilting is possible in the r-theta plane only when both the radial and the theta directions have been discretized, that is,  $ni > 1$  and  $nj > 1$  in the \*GRID keyword. When  $ni = 1$ , then  $nj = 1$  automatically. When  $nj = 1$ , tilt causes part of the block to be raised and part of the same block to be lowered from the untilted elevation, since 'block center' is not well defined in this case. In any case, Z-direction gravity treatment is correct.

Note that RESULTS 3D does not display the reservoir tilted with the \*DIP angles.

Therefore, it is recommended that tilts near 90 degrees be accomplished by changing the grid axis being used. For example, a vertical 1D grid can use  $ni = nj = 1$  and no tilt as opposed to  $nj = nk = 1$  with 90 degrees tilt.



$\alpha$  = angle that X (or I) axis is raised from horizontal

$\beta$  = angle that Y (or J) axis is raised from horizontal

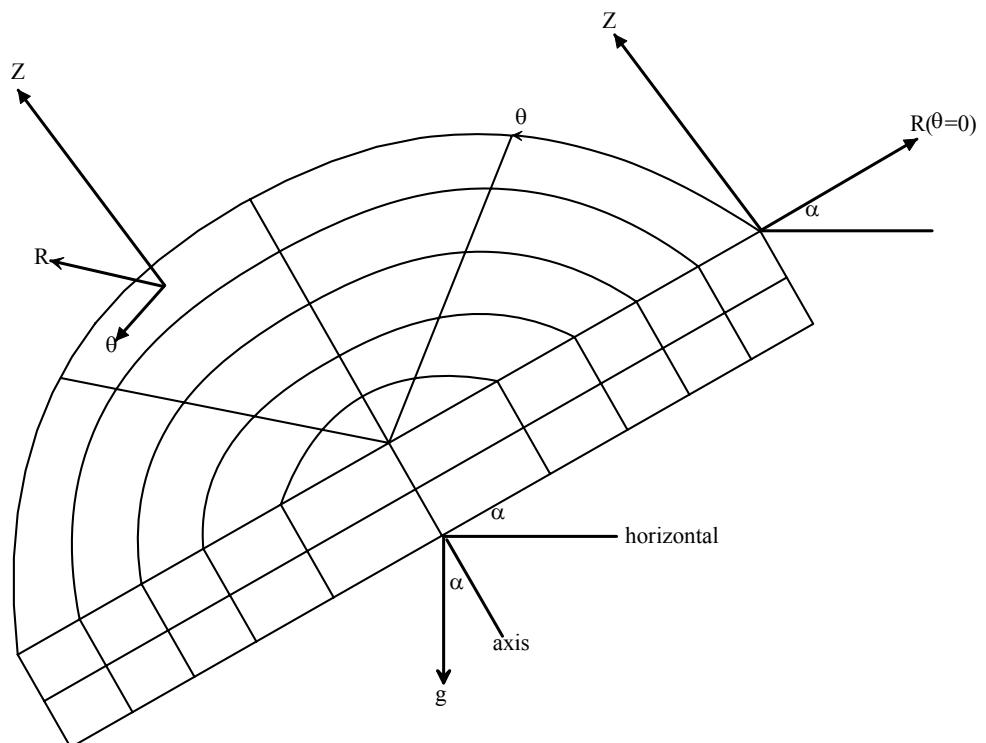
Direction cosines of gravity vector with respect to tilted coordinate axes are :

$$g_x = -\sin \alpha$$

$$g_y = -\sin \beta$$

$$g_z = -\sqrt{1 - g_x^2 - g_y^2}$$

*Figure 3: Internal Calculation of Gravity Components Tilted for Cartesian Coordinates*



$\alpha$  = angle that \$R\$ (or I) axis at  $\theta = 0$  is raised from horizontal

Direction cosines of gravity vector with respect to local coordinate axes vary with axis angle  $\theta$  as follows :

$$g_r = -\sin \alpha \cos \theta$$

$$g_\theta = -\sin \alpha \sin \theta$$

$$g_z = -\cos \alpha$$

*Figure 4: Internal Calculation of Gravity Cosines for Tilted Cylindrical Coordinates*

---

## **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:**

The keyword is available only with \*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.

The \*RG qualifier can be used with this keyword. The single array reading option is described below.

### **EXPLANATION:**

See the general corner point discussion given with the \*GRID \*CORNER keyword for discussions of the notation used here. The unit is (m | ft | cm).

The acceptable range of values for corner point depths is:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

### **Pseudo-Infinite Blocks**

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

### **Algorithm for \*ZCORN Ordering**

The \*ZCORN keyword causes the reading of all depths (Z-coordinates) of the  $8*ni*nj*nk$  corner points required to define the grid. The depths should be entered as follows:

Operations should be done in the order shown. Note that the text before each task [...] describes how often to carry out that task.

```

Do the following for K = 1, ..., nk: [
  Do the following for J = 1, ..., nj: [
    Write NW-T and NE-T depths for block ( 1,J,K),
    ...
    Write NW-T and NE-T depths for block (ni,J,K).
    Write SW-T and SE-T depths for block ( 1,J,K),
    ...
    Write SW-T and SE-T depths for block (ni,J,K).
  ]
  Do the following for J = 1, ..., nj: [
    Write NW-B and NE-B depths for block ( 1,J,K),
    ...
    Write NW-B and NE-B depths for block (ni,J,K).
    Write SW-B and SE-B depths for block ( 1,J,K),
    ...
    Write SW-B and SE-B depths for block (ni,J,K).
  ]
]
```

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.

Example:

Provide corner point depths for a ni = 4, nj = 2, nk = 1 \*CORNER grid for a reservoir dipping in the "x"-coordinate direction whose layers are a constant 10 units thick.

```
*ZCORN
 2000  2001  2001  2002  2002  2003  2003  2004
 2000  2001  2001  2002  2002  2003  2003  2004
 2000  2001  2001  2002  2002  2003  2003  2004
 2000  2001  2001  2002  2002  2003  2003  2004
 2010  2011  2011  2012  2012  2013  2013  2014
 2010  2011  2011  2012  2012  2013  2013  2014
 2010  2011  2011  2012  2012  2013  2013  2014
 2010  2011  2011  2012  2012  2013  2013  2014
```

---

## **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  
\*YCORN

### **DEFAULTS:**

Conditional keyword. No defaults.

### **CONDITIONS:**

The keyword is available only with \*GRID \*CORNER. Both keywords should appear in combination with \*ZCORN to define all the corner point locations.

The \*RG qualifier can be used with this keyword. The single array reading option is described below.

### **EXPLANATION:**

See the general corner point discussion given for the \*GRID \*CORNER keyword for discussions of the notation used here. The unit is (m | ft | cm).

The \*XCORN and \*YCORN keywords each cause reading of all the  $(ni+1)*(nj+1)*(nk+1)$  "x"- and "y"-coordinate values required to define the lateral locations of all points defining a \*CORNER grid. The values should be input as described in the following algorithm.

#### **Algorithm for \*XCORN/\*YCORN Ordering**

Operations should be done in the order shown. Note that the text before each task [...] describes how often to carry out that task.

Do the following for K = 1, ..., (nk + 1): [  
Do the following for J = 1, ..., (nj + 1): [  
Do the following for I = 1, ..., (ni + 1): [  
    I, J, K are less than ni, nj, nk, respectively:  
        write the "x"- (or "y"-) coordinate of the NW-T point;  
    J is less than nj, K is less than nk, and I = ni:  
        write the "x"- (or "y"-) coordinate of the NE-T point;  
    I is less than ni, K is less than nk, and J = nj:  
        write the "x"- (or "y"-) coordinate of the SW-T point;

I is less than ni, J is less than nj, and K = nk:  
 write the "x"- (or "y"-) coordinate of the NW-B point;  
 I is less than ni, and J = nj, K = nk:  
 write the "x"- (or "y"-) coordinate of the SW-B point;  
 J is less than nj, and I = ni, K = nk:  
 write the "x"- (or "y"-) coordinate of the NE-B point;  
 K is less than nk, and I = ni, J = nj:  
 write the "x"- (or "y"-) coordinate of the SE-T point;  
 I = ni, J = nj, K = nk:  
 write the "x"- (or "y"-) coordinate of the SE-B point;  
 where the choice of "x"- or "y"- is determined by  
 whether \*XCORN or \*YCORN is being written.

]]]

This completes the algorithm. Note that I is ranging fastest, and K slowest, in the above; J is intermediate.

Examples:

Provide the \*XCORN and \*YCORN data for a ni = 4, nj = 2, nk = 1 \*CORNER grid. Note that the "x"- direction grid spacing is uniformly 100 units and the "y"-direction grid spacing is uniformly 200 units.

```
*XCORN
 0 100 200 300 400
 0 100 200 300 400
 0 100 200 300 400
 0 100 200 300 400
 0 100 200 300 400
 0 100 200 300 400

*YCORN
 0 0 0 0 0
 200 200 200 200 200
 400 400 400 400 400
 0 0 0 0 0
 200 200 200 200 200
 400 400 400 400 400
```

The acceptable range of values for corner point coordinates is:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

### Pseudo-Infinite Blocks

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

---

## **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:**

The keyword is available only with \*GRID \*CORNER. Combine this keyword with \*ZCORN to define all the corner point locations.

The \*RG qualifier can be used with this keyword. The single array reading option is described below.

### **EXPLANATION:**

See the general corner point discussion given for the \*GRID \*CORNER keyword for discussions of the notation used here. The unit is (m | ft | cm).

The \*COORD keyword 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  $(ni+1) * (nj+1)$  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  $2 * 3 * (ni + 1) * (nj + 1)$  values as described in the following algorithm.

#### **Algorithm for \*COORD Ordering**

Operations should be done in the order shown. Note that the text before each task [...] describes how often to carry out that task.

Do the following for  $J = 1, \dots, (nj + 1)$ : [

Do the following for  $I = 1, \dots, (ni + 1)$ : [

Firstly, ...

If  $I$  and  $J$  are less than  $ni$  and  $nj$  respectively, write the "x"-, "y"-, "z"-coordinates of a point that lies on a vertical line through the NW corner of block  $(I,J,1)$ . This could be the "-B" or "-T" corner, or block  $(I,J,K)$ 's corner for any  $K$ , as all these points should be collinear.

If  $I = ni$  and  $J$  is less than  $nj$ , write the NE corner. If  $I$  is less than  $ni$  and  $J = nj$ , write the SW corner. If  $I = ni$  and  $J = nj$ , write the SE corner.

Secondly, ...

If I and J are less than ni and nj respectively, write the "x"-, "y"-, "z"-coordinates of another point that lies on a vertical line through the NW corner of block (I,J,1). This point should differ from the previous one only in its "z"- coordinate.

If I = ni and J is less than nj, write the NE corner. If I is less than ni and J = nj, write the SW corner. If I = ni and J = nj, write the SE corner.

] ]

This completes the algorithm.

Note that I is ranging fastest, J slowest in the above.

As \*COORD data only provides lines on which corner points must lie, \*ZCORN array data is still required to locate the corner points along the lines.

Examples:

Provide \*COORD data for a ni = 4, nj = 2, nk = 1 \*CORNER grid. Note that the "x"-direction grid spacing is uniformly 100 units and the "y"-direction grid spacing is uniformly 200 units. (This example appears the same regardless of the value for nk.)

```
*COORD
    0   0 0     0   0 1     100  0 0     100  0 1
    200 0 0     200 0 1     300  0 0     300  0 1
    400 0 0     400 0 1

    0   200 0     0   200 1     100  200 0     100  200 1
    200 200 0    200 200 1     300  200 0     300  200 1
    400 200 0    400 200 1

    0   400 0     0   400 1     100  400 0     100  400 1
    200 400 0    200 400 1     300  400 0     300  400 1
    400 400 0    400 400 1
```

The acceptable range of values for corner point coordinates is:

	<b>SI m</b>	<b>Field ft</b>	<b>Lab cm</b>
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

### Pseudo-Infinite Blocks

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

---

## **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:**

The keyword is available only with \*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. The single array reading option is described below.

### **EXPLANATION:**

See the general corner point discussion given with the \*GRID \*CORNER keyword for discussions of the notation used here. The unit is (m | ft | cm).

This keyword causes the processing of  $3*(8*ni*nj*nk)$  values, with the first group of  $8*ni*nj*nk$  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  $8*ni*nj*nk$  values is the same array that would be input using the \*ZCORN keyword.

#### **Algorithm for \*CORNERS Ordering**

Operations should be done in the order shown. Note that the text before each task [...] describes how often to carry out that task.

Do the following three times with:

- (1) "values" replaced by ""x"-coordinate values";
- (2) "values" replaced by ""y"-coordinate values";
- (3) "values" replaced by ""z"-coordinate values", the latter also being the depths:

[

```

Do the following for K = 1, ..., nk: [
  Do the following for J = 1, ..., nj: [
    Write NW-T and NE-T values for block ( 1,J,K)
    ...
    Write NW-T and NE-T values for block (ni,J,K)
    Write SW-T and SE-T values for block ( 1,J,K)
    ...
    Write SW-T and SE-T values for block (ni,J,K)
  ]]
  Do the following for J = 1, ..., nj: [
    Write NW-B and NE-B values for block ( 1,J,K)
    ...
    Write NW-B and NE-B values for block (ni,J,K)
    Write SW-B and SE-B values for block ( 1,J,K)
    ...
    Write SW-B and SE-B values for block (ni,J,K)
  ]]

```

This completes the algorithm.

This technique for corner point input will exhibit duplication in the first two groups of  $8*ni*nj*nk$  values, due to the fact that corner points must lie on vertical lines.

Examples:

Provide \*CORNERS data for a  $ni = 4$ ,  $nj = 2$ ,  $nk = 1$  grid in a reservoir dipping in the "x"-coordinate direction. Note that the single layer is 10 units thick and that the "x"-direction grid spacing is 100 units and the "y"-direction grid spacing is 200 units.

```

*CORNERS
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0   100   100   200   200   300   300   400
  0     0     0     0     0     0     0     0
  200  200   200   200   200   200   200   200
  200  200   200   200   200   200   200   200
  400  400   400   400   400   400   400   400
  0     0     0     0     0     0     0     0
  200  200   200   200   200   200   200   200
  200  200   200   200   200   200   200   200
  400  400   400   400   400   400   400   400
  2000 2001 2001 2002 2002 2003 2003 2004
  2000 2001 2001 2002 2002 2003 2003 2004
  2000 2001 2001 2002 2002 2003 2003 2004
  2000 2001 2001 2002 2002 2003 2003 2004
  2010 2011 2011 2012 2012 2013 2013 2014
  2010 2011 2011 2012 2012 2013 2013 2014
  2010 2011 2011 2012 2012 2013 2013 2014
  2010 2011 2011 2012 2012 2013 2013 2014

```

The acceptable range of values for corner points are:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
min	-1.0E+20	-3.28E+20	-1.0E+22
max	1.0E+20	3.28E+20	1.0E+22

### Pseudo-Infinite Blocks

The use of very large blocks to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

## Corner Point Tolerance (Optional)

\*CORNER-TOL, \*PARTOL-TOL

### PURPOSE:

\*CORNER-TOL controls the minimal spacing required to separate corner points (see above for descriptions of Corner Point grids). It is also used for miscellaneous tolerance checking for corner point applications.

### FORMAT:

\*CORNER-TOL *cptol*  
\*PARTOL-TOL *partol*

### DEFINITIONS:

*cptol*

Minimal spacing required to separate corner points and related quantities; that is, corner points that are closer than *cptol* are deemed to be the same. Dimensions are (m | ft).

*partol*

Minimal spacing required to separate corner points and related quantities. When a corner point grid has refinements with small cells, and a run terminates with a Fatal Error from subroutine GRCNC8, setting *partol* to a value below its default may remove those messages. Dimensions are (m | ft).

### DEFAULTS:

If \*CORNER-TOL is absent then *cptol* = 0.050 is assumed for all length units.

If \*PARTOL-TOL is absent then *partol* = 0.005 is assumed for all length units.

### CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group.

### EXPLANATION:

Corner points that lie within a distance of *cptol* are considered to be in the same place. If two corner points that belong to the same cell lie within a distance of *cptol*, then, either:

- one point belongs to the top of the cell and the other to the bottom, and that corner is pinched out;
- the cell is squeezed in the I or J direction and an error has occurred.

Points from neighbouring cells (four points from the top of one cell and four points from the bottom of the other) that are supposed to be touching to make a standard flow connection will be regarded as making contact if they lie within a distance of *cptol* (in top-bottom pairs).

If the average thickness of the cell as measured through its centre is less than a certain tolerance (see \*PINCHOUT-TOL), then that cell will be designated as pinched out.

However, for the Corner Point cells above and below to make a connection, those cells' top and bottom corner points must match to within the tolerance *cptol*.

---

## Local Refined Grid (Conditional)

\*REFINE, \*RANGE

### PURPOSE:

\*REFINE indicates the input of local refined grid.

### FORMAT:

```
*REFINE    block_address *INTO nir njr nkr
*REFINE    block_address *INTO nr ntheta nz *HYBRID
            (*IDIR | *JDIR | *KDIR) *RW rw
            (*ALPHAI alphai) (*ALPHA alpha)
-or-
*REFINE    nir njr nkr
*REFINE    *HYBRID nr ntheta nz (*IDIR | *JDIR | *KDIR) *RW rw
            (*ALPHAI alphai) (*ALPHA alpha)
*RANGE     block_address
```

### DEFINITIONS:

#### \*REFINE

Indicates the application of local grid refinement of the specified type and parameters to the specified block or range of blocks.

#### block\_address

The address of the grid block(s) to which this refinement applies. Two forms are allowed: multi-level single-block UBA (see **Multi-level Regular Refinement**, below), and single-level range i1(:i2) j1(:j2) k1(:k2).

#### \*INTO

Indicates as new refinement. This must be present at the first appearance of \*REFINE. Subsequent usage of \*REFINE without \*INTO causes the same refinement to be used.

#### nir

Number of refined blocks in the I direction within each fundamental grid block.

#### njr

Number of refined blocks in the J direction within each fundamental grid block.

#### nkr

Number of refined blocks in the K direction within each fundamental grid block.

#### nr

Number of radial subdivisions in the R-theta-Z local hybrid grid. Allowed values for nr are 2,3,4,... up to a maximum of 10.

<b>ntheta</b>	Number of theta subdivisions in the R-theta-Z local hybrid grid. Permitted values are 1 or 4. Theta subdivisions are not applied to the inner-most hybrid grid block.
<b>nz</b>	Number of Z-direction subdivisions in the R-theta-Z local hybrid grid. The "Z" direction of the hybrid grid is specified using the *IDIR, *JDIR, or *KDIR keywords. You are allowed to divide a fundamental grid into a maximum of 4 refined grids. Permitted values are 1,2,3 and 4.
<b>*HYBRID</b>	Indicates the use of hybrid grid refinement in which a Cartesian grid block (normally containing a well) is refined into a local cylindrical R-theta-Z grid.
<b>*IDIR</b>	Indicates that the "Z" axis of the hybrid grid is parallel to the I-direction of the fundamental grid.
<b>*JDIR</b>	Indicates that the "Z" axis of the hybrid grid is parallel to the J-direction of the fundamental grid.
<b>*KDIR</b>	Indicates that the "Z" axis of the hybrid grid is parallel to the K-direction of the fundamental grid. This is the default.
<b>*RW rw</b>	<p>Value for the well radius (m   ft   cm) must be greater than zero and 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.</p> <p>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.</p>
<b>*ALPHAI alphai</b>	Define the ratio of the outer radius of the first ring to "rw". Used only for isotropic *HYBRID cases. The value of alphai should exceed 1.
<b>*ALPHA alpha</b>	Defines the ratio of successive outer radii for rings i to i-1, for each i running from 2 through nir-1. (The condition is not applicable to ring nir, as it is truncated to have four planar sides in order to fit its neighbours.) Used only for isotropic *HYBRID cases. The value of alpha should exceed 1.

## **DEFAULTS:**

If \*REFINE is absent, there are no refined grids. If \*INTO is absent it will default to the previous refinement; the first \*REFINE must have \*INTO.

There are no defaults for nir, njr and nkr. For \*HYBRID there are no defaults for nr, ntheta, nz and rw. For hybrid grid, the inner radial block never has theta subdivisions.

When \*HYBRID is used and none of \*IDIR, \*JDIR and \*KDIR are specified, the default is \*KDIR.

The default value for \*ALPHA is chosen so that if the outer most ring (ring nir) was allowed to be circular with an outer radius equal to that of ring nir-1 multiplied by "alpha" (so that it was treated like the other rings), its area would equal the total area available. The outermost ring is truncated to have flat sides so that neighbouring blocks can be properly fitted.

For anisotropic media, the values for \*ALPHAI and \*ALPHA are calculated internally using much the same criteria as discussed above for the isotropic case, except that elliptical geometries are used.

## **CONDITIONS:**

For regular refinement, there is no internal limit to the magnitudes of nir, njr and nkr. However, values larger than 3-5 tend to produce numerically inconsistent results at the interface between coarse and fine blocks.

Hybrid grid refinements can be used only with Cartesian grids, i.e. \*GRID \*CART.

\*REFINE may not be used together with \*NINEPOINT.

When defining a discretized wellbore inside a hybrid grid, the hybrid grid must be defined first. See detailed explanation for \*WELLBORE.

The only types of multi-level refinement allowed are (1) Cartesian regular refinement, (2) Cartesian regular refinement with hybrid grid at the finest level, and (3) discretized wellbore in hybrid grid. At least 10 levels of refinement are available.

Areas with different degrees of refinement must be separated by at least one unrefined grid block. See the detailed descriptions below.

The \*REFINE keyword can be used with \*GRID \*CART, \*GRID \*VARI or \*GRID \*CORNER but not with \*GRID \*RADIAL.

Pinched out (and null) cells can be marked as refined in data, without terminating the simulation, although these cells will remain inactive.

Local grid refinement may be used with natural fracture options \*DUALPOR and \*DUALPERM but not \*MINC or \*SUBDOMAIN.

## **EXPLANATION:**

\*REFINE may occur several times to define multiple regions or refinement types.

By default, refined grid blocks are assigned the properties of the fundamental block in which they reside. Alternatively the properties of the refined grid may be entered by using the \*RG array qualifier keyword with any array keyword.

The variable depth/variable thickness option may be used with refined grids.

See Appendices E.6 and E.7 for further discussion.

## REGULAR REFINEMENT

The parent block is refined into a child grid that is of the same type and orientation as the parent block. In each refined direction the refined block sizes are uniform. The I-J-K indices in the local grid follow the same sense as the parent block, but the local origin starts at the corner of the parent block closest to the global origin.

One rule applies to adjacent parent blocks with regular refinement: in each refined direction, parent blocks adjacent in that direction must be refined by the same amount normal to that direction. For example, if block (I,J,K) has been refined into nir x njr x nkr, then

- blocks (I-1,J,K) and (I+1,J,K) must be refined with the same njr and nkr, if at all;
- blocks (I,J-1,K) and (I,J+1,K) must be refined with the same nir and nkr, if at all; and
- blocks (I,J,K-1) and (I,J,K+1) must be refined with the same nir and njr, if at all

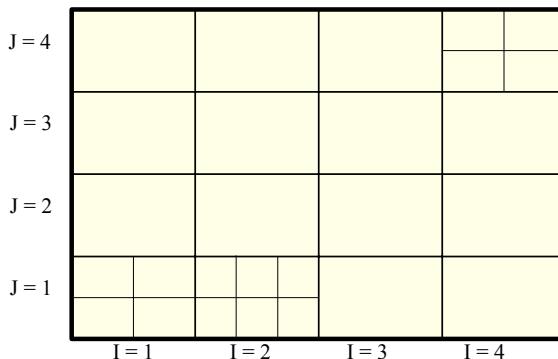
Example:

A grid system consists of 4 blocks in the I direction, 4 blocks in the J direction and 2 blocks in the K direction. Two columns of grid blocks are to be refined, with two refined blocks in each direction. The data file is as follows:

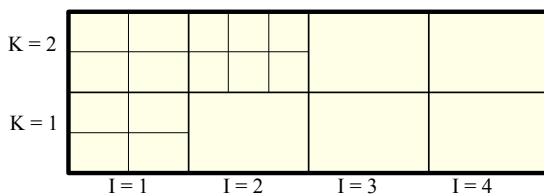
```
*GRID *CART 4 4 2  
.*.  
*REFINE 1 1 1:2      *INTO 2 2 2  
*REFINE 4 4 1:2  
*REFINE 2 1 2      *INTO 3 2 2
```

Note that two regions were assigned the same refinement type 2 x 2 x 2. Also, blocks (1,1,1) and (2,1,1) are adjacent in the I direction and so must have the same J- and K-direction refinement, but may have a different nir.

An areal view of the grid for K=2 would be:



A cross-section for J=1 would be:



## **Multi-level Regular Refinement**

Regular local grid refinement may extend to more than one level. For example, this data fragment specifies 5 levels of 3x3 refinement in a single fundamental block.

```
*refine 5 3 2 *into 3 3 3
*refine 5 3 2 / 2 2 *into 3 3 3
*refine 5 3 2 / 2 2 2 / 2 2 2 *into 3 3 3
*refine 5 3 2 / 2 2 2 / 2 2 2 / 2 2 2 *into 3 3 3
*refine 5 3 2 / 2 2 2 / 2 2 2 / 2 2 2 / 2 2 2 *into 3 3 3
```

Note that no range is allowed in the parent *block\_address* when it is refined, that is, has a slash in the UBA. Be aware that excessive use of multi-level refinement can increase the number of grid blocks significantly.

## **HYBRID REFINEMENT**

The hybrid grid option refers to refining a parent block from a Cartesian grid into a local cylindrical grid whose "axial" direction may be oriented in either the global I, J or K direction. There are nr divisions in the radial direction, of which the outermost is formed to fit the shape of the parent block. The hybrid's angular direction is divided into either 1 or 4 divisions; the innermost radial division is always a full circle. The hybrid's axial direction is divided into nz uniform sections. It is anticipated, but not required, that a well will be placed inside the innermost radial division.

The \*HYBRID option may affect results especially when a process is influenced strongly by near-wellbore phenomena, e.g., cyclic steam stimulation. The well can be horizontal or vertical. The wellbore and the corresponding hybrid grid axis must go through the centre of the grid block. Thus, the hybrid grid's local "Z-axis" may be in the global I, J or K direction depending on the well direction.

Perpendicular to this axis the permeability may be equal (isotropic case) or not equal (anisotropic case). The aspect ratio of grid dimensions normal to the axial direction should not be too different from the square root of the corresponding absolute permeability ratio. Deviations of more than a factor of 1.25 can lead to large errors and so is not allowed.

### **ISOTROPIC CASE:**

This is normally the case for a hybrid grid whose axis is in the vertical direction. The grid dimensions must be within a factor of 1.25 of square.

### **ANISOTROPIC CASE:**

This is normally the case for a hybrid grid whose axis is in a horizontal direction. The aspect ratio of block size normal to the axial direction should be within a factor of 1.5 to 2 of the square root of the ratio of the corresponding absolute permeabilities. For example, a well horizontal in the X-direction with Ky = 10 Kz should have a grid aspect ratio of about delta\_Y/delta\_Z = square\_root(10).

Two rules apply to adjacent parent blocks refined with \*HYBRID:

1. For hybrid grids adjacent in the hybrid's axial direction, nr and ntheta must be the same, and nz may be different. This case is typical for modelling a hybrid grid around a well that passes through more than one parent block.
2. For hybrid grids adjacent in a direction other than the hybrid's axial direction, only nz must be the same.

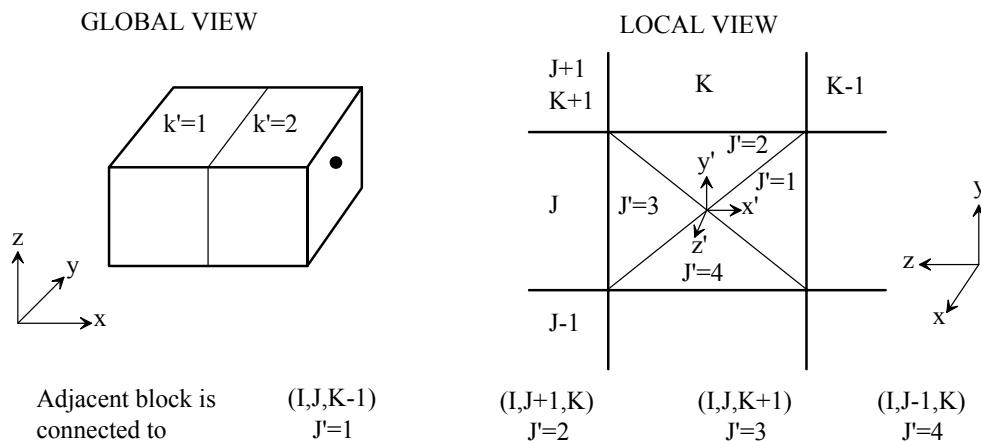
## Hybrid Grid Orientations

Normally, the user will need to know only which of \*IDIR, \*JDIR or \*KDIR was specified in order to interpret the position of the individual hybrid grid blocks relative to the surrounding fundamental blocks. However, the precise meaning of the hybrid grid's local J and K indices is needed in order to assign non-uniform properties and conditions, and to interpret in detail the textual output.

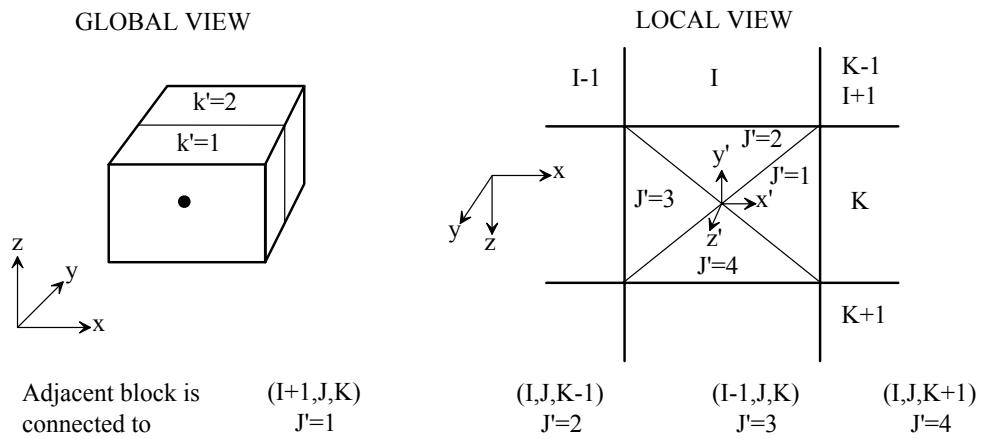
In the following, x, y, z, I, J and K refer to the fundamental grid and similar primed ('') symbols refer to the local cylindrical grid.

In each of the following orientation cases, both a "GLOBAL VIEW" and a "LOCAL VIEW" are shown. In the "LOCAL VIEW" the point-of-view is on the hybrid z' axis looking in the negative z' direction, i.e., z' points toward the viewer. Note that the only difference between the "LOCAL VIEW" of the cases is the relation to the global coordinates.

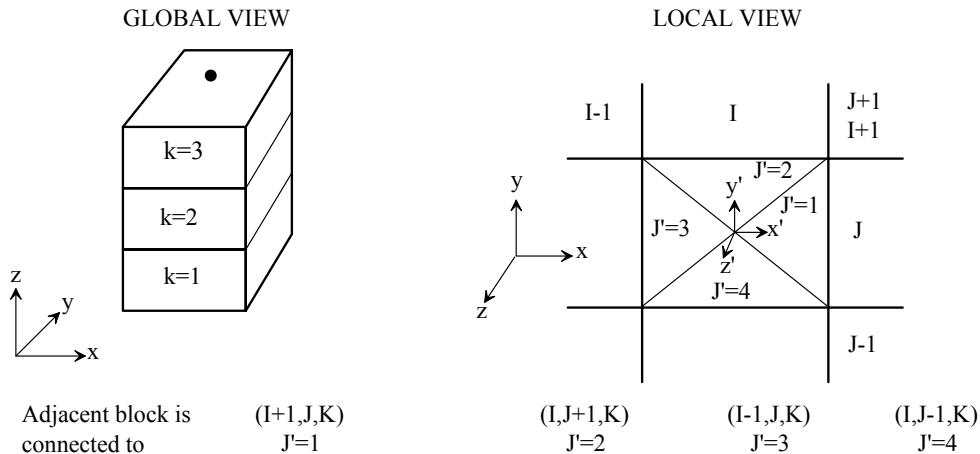
Well in I Direction (\*IDIR):  $x = xo + z'$ ,  $y = yo + y'$ ,  $z = zo - x'$



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 (constant and pressure-dependent)
- block area modifiers
- dispersion

and output

- block sizes
- permeabilities
- transmissibility multipliers (constant and pressure-dependent)
- transmissibilities
- conduction geometry factors
- block area modifiers
- dispersion.

Note that connection-based quantities such as transmissibility have an explicit Radial and Angular direction printout, but the Axial direction values are found through this direction correspondence.

### Block Sizes

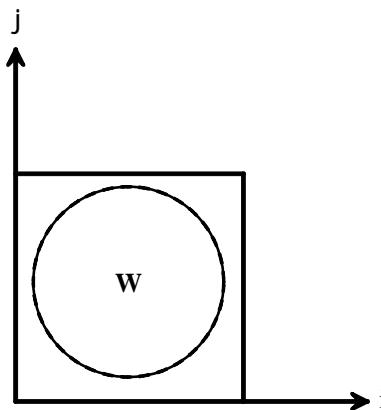
The block sizes reported for the hybrid grid blocks are similar to those found for a cylindrical grid system. The radial and axial block sizes have standard definitions. The angular direction block size is the midpoint arc length, so that the product of the block sizes in the three directions is equal to the block volume (without volume modifier).

The only exception to this is 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. The radial block size is an average value which gives 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

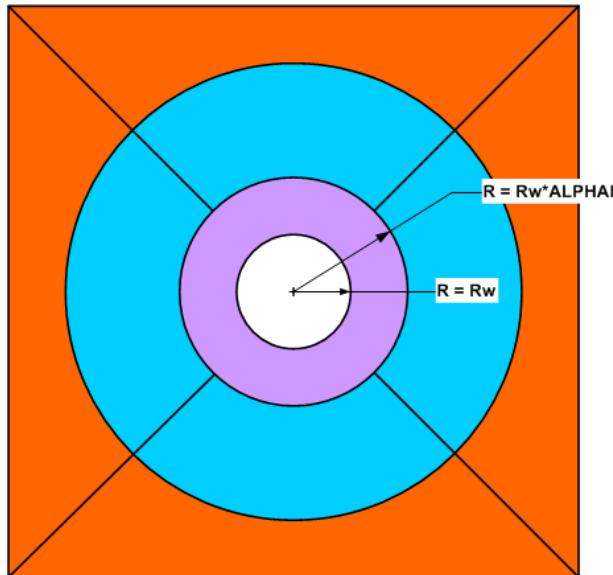
```
*REFINE 1 1 1 *INTO 3 4 1 *HYBRID *IDIR
```

# 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.



## Changing Grid Refinement at Well Changes and Timesteps

Grid refinement and de-refinement can be done for various times in recurrent data. See **Recurrent Grid Control** in the Well and Recurrent Data chapter.

Grid refinement and de-refinement can be done dynamically between timesteps according to user-specified criteria. See **Dynamic Grid Control** in the Well and Recurrent Data chapter.

---

## Block Geometry Modifiers (Optional)

\*VAMOD, \*VATYPE

### PURPOSE:

Describes modification to grid block volumes and face areas.

### FORMAT:

\*VAMOD key v ai aj ak (ai- aj- ak-) (\*9P aij+ aij-)

### ARRAY:

\*VATYPE

### DEFINITIONS:

key

Integer key associated with this geometry type, to be used with \*VATYPE. You do not need to define a key for the unmodified type or null-block type. It is suggested that you define your own modifier types using key = 2 and up, leaving predefined key = 0 for null blocks and predefined key = 1 for unmodified (whole) blocks.

v

Block volume modification factor, equal to (desired gross volume) / (product of block sizes \*DI, \*DJ and \*DK). It is needed even for zero-porosity blocks to correctly account for energy in rock. A value of zero denotes a true null block, with no pore volume and no rock volume.

ai

Area modifier factor in the I direction, equal to (desired area) / (area from block sizes \*DJ and \*DK). A zero value will result in no flow.

aj

Area modifier factor in the J direction, equal to (desired area) / (area from block sizes \*DI and \*DK). A zero value will result in no flow.

ak

Area modifier factor in the K direction, equal to (desired area) / (area from block sizes \*DI and \*DJ). A zero value will result in no flow.

ai-

Area modifier factor in the -I direction, used in situations where ai varies along the I direction.

aj-

Area modifier factor in the -J direction, used in situations where aj varies along the J direction.

ak-	Area modifier factor in the -K direction, used in situations where ak varies along the K direction.
aij+	Area modifier factor in the I+J+ direction for *NINEPOINT *IJ and in the I+K+ direction for *NINEPOINT *IK. This is needed only along the diagonal boundaries of symmetry patterns.
aij-	Area modifier factor in the I+J- direction for *NINEPOINT *IJ and in the I+K- direction for *NINEPOINT *IK. This is needed only along the diagonal boundaries of symmetry patterns.
VATYPE	Assign a modifier type key to the grid, including refined blocks. A key value of 0 denotes a null block. A key not defined using *VAMOD refers to the unmodified type. See 'key', above.

#### **DEFAULTS:**

If \*VAMOD and \*VATYPE are absent, all blocks are active and their full volumes and areas are used.

If ai-, aj- and ak- are absent then ai- = ai, aj- = aj and ak- = ak. This is appropriate when the factor does not vary along its direction. When it does, the + and - face of a block will have a different factor, and both ai and ai- must be given values.

#### **EXPLANATION:**

##### **Typical Uses of Geometry Modifiers**

Typical uses for block geometry modifiers are:

1. Place centres of outer blocks on the reservoir boundary,
2. Model symmetry elements of repeating patterns (see Appendix E.5), 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.

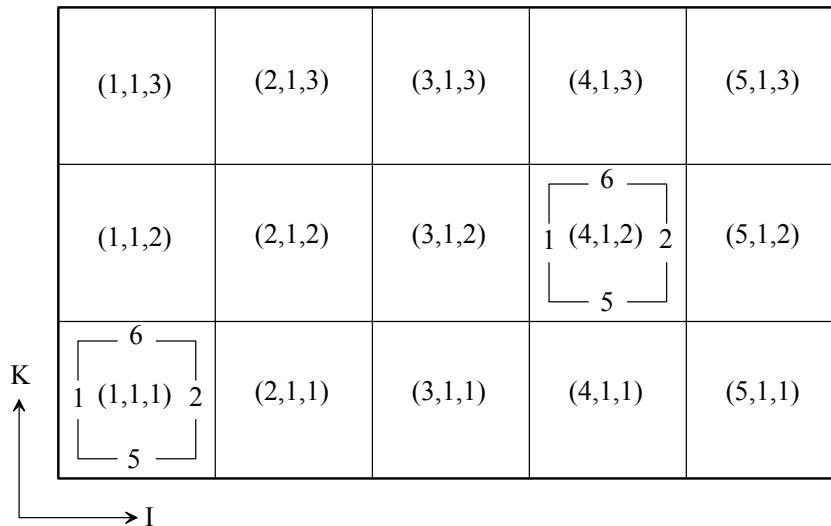
## Referencing Grid Block Faces

An area modifier applies to the interface between the current block and the adjacent block in the "plus" coordinate axis direction. The "plus" 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 5 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 "minus" directions are rarely needed since value for those interfaces default to the value of the "plus" 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 "minus" 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 5. Suppose each block interface in the I direction has a different volume and area factors V1, A1, V2, A2, etc. Do the following to get the area factors consistent, assuming  $a_k = v$  and  $a_j = 1$  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 5: Referencing Grid Block Faces**

## Definitions of Geometry Factors

Figure 6 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 6. 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

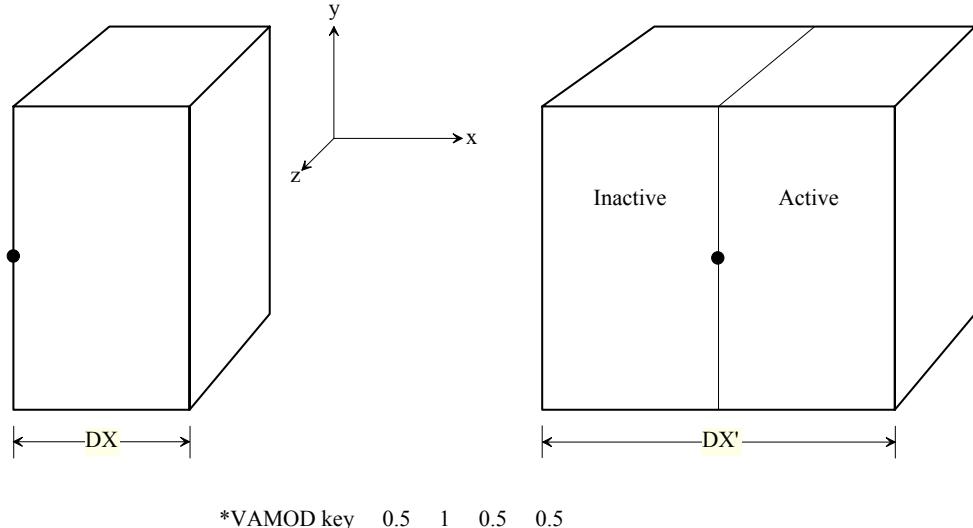
$$\begin{aligned}
 v &= [\text{desired volume}] / [\text{volume from block sizes}] \\
 &= [DX * DY * DZ] / [DX' * DY * DZ] \\
 &= 0.5 \\
 ai &= [\text{desired area}] / [\text{area from block sizes}] \\
 &= [DY * DZ] / [DY * DZ] \\
 &= 1 \\
 aj &= [\text{desired area}] / [\text{area from block sizes}] \\
 &= [DX * DZ] / [DX' * DZ] \\
 &= 0.5 \\
 ak &= [\text{desired area}] / [\text{area from block sizes}] \\
 &= [DX * DY] / [DX' * DY] \\
 &= 0.5
 \end{aligned}$$

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

```



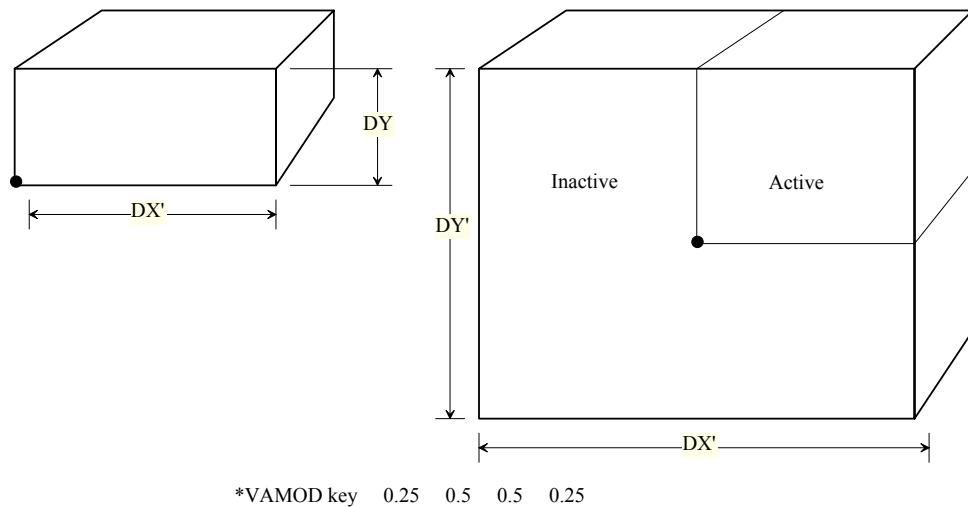
**Figure 6: Grid Node on a Side Block Boundary**

Figure 7 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' . . .
*DJ *JVAR DY' . . .
*VAMOD key 0.25 1 0.5 0.25
          ** Assign DX' to boundary block
          ** Assign DY' to boundary block
          ** Split block in half in X
          ** direction
```



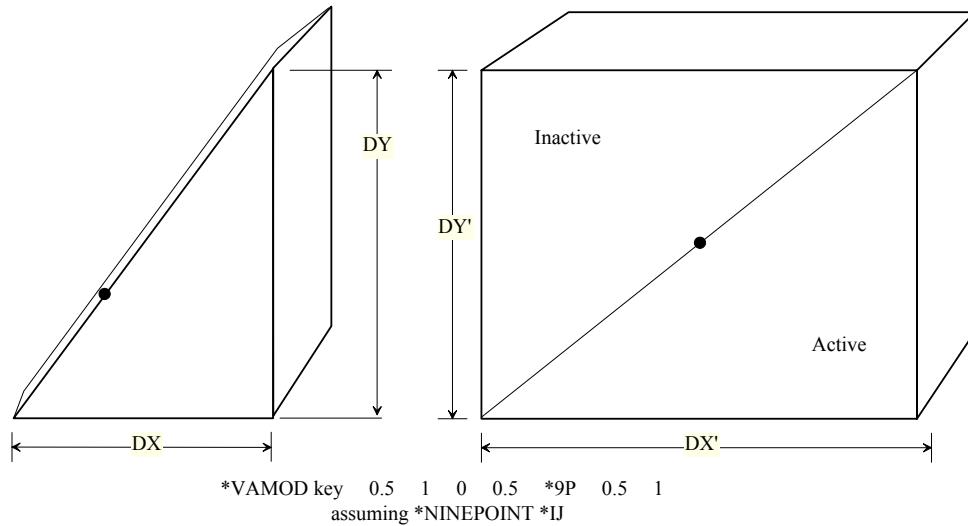
**Figure 7: Grid Node in a Block Corner**

Figure 8 shows how to place a block node on a diagonal boundary. Both the X and Y directions are extended, but the geometry factors come more from inspection. Here, the nine-point option is illustrated. The meaning of the geometry factors is

$$\begin{aligned} v &= [\text{desired volume}] / [\text{volume from block sizes}] = 0.5 \text{ by inspection} \\ ai &= 1 \text{ since this whole face is on the active part of the block,} \\ aj &= 0 \text{ since this face is on the inactive part of the block, and} \\ ak &= v = 0.5, \\ aij+ &= [\text{desired diagonal "area"}] / [\text{existing diagonal "area"}] = 0.5 \text{ by inspection} \\ aij- &= [\text{desired diagonal "area"}] / [\text{existing diagonal "area"}] = 1 \text{ by inspection} \end{aligned}$$

and the keywords are

```
*NINEPOINT *IJ
*DI *IVAR DX' . . .
*DJ *JVAR DY' . . .
*VAMOD key 0.5 1 0 0.5 *9P 0.5 1
                           ** Assign DX' to boundary block
                           ** Assign DY' to boundary block
                           ** Split block in half in X dir
```



*Figure 8: Grid Node on a Diagonal Boundary*

### Null Blocks

You can use \*VATYPE to specify null blocks instead of \*NULL. In fact, this is preferred if any geometry modifiers will be assigned via \*VAMOD. Use key value 0 for null blocks, just as for \*NULL.

There is no interblock connection to null blocks or beyond the reservoir boundary, and area modifiers corresponding to such connections are not needed internally. When a number is required to satisfy the syntax of the \*VAMOD keyword, but you know it will not be used, enter 0.

Example: One-eighth of a 5-spot Symmetry Element

Apply the 9-point option to a 9x5 grid of square blocks, and then trim to 1/8 of a 5-spot pattern.

```
*GRID *CART 9 5 1 *NINEPOINT *IJ
*DI *CON 10
*DJ *EQUALSI
**   key   v     ai    aj    ak      aij+    aij-
*VAMOD 2  0.5   0.5   1.0   0.5          ** like Fig 6
*VAMOD 3  0.5   1.0   1.0   0.5   *9P  0.5   1.0   ** like Fig 8
*VAMOD 4  0.5   1.0   1.0   0.5   *9P  1.0   0.5   ** like Fig 8
*VAMOD 5  0.125 0.5   1.0   0.125  *9P  0.5   1.0
*VAMOD 6  0.25  1.0   1.0   0.25   *9P  1.0   0.5
```

```

*VATYPE *ALL 5 2 2 2 2 2 2 2 2 5
  0 3 1 1 1 1 1 4 0    ** ---- i
  0 0 3 1 1 1 4 0 0   **
  0 0 0 3 1 4 0 0 0   **
  0 0 0 0 6 0 0 0 0   ** j

```

The only difference between keys 3 and 4 is the \*9P values; these keys could be merged if \*NINEPOINT were not used. Array-reading option \*ALL was used with \*VATYPE so that the data itself can make a picture of the grid when arranged in rows of ni, columns of nj and planes of nk.

### Refined Grid

By default, all blocks of a refined grid have the same values for a quantity or property (except block size) as the grid's parent block. This applies also to block modifiers. Geometry modifiers may be entered for specific refined blocks using the subkeyword \*RG.

Suppose that the block in Figure 6 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 6, 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 a unrefined block in the "plus" 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 (see section **Well Fraction** in the introductory summary of chapter **Well and Recurrent Data**). 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 6, 7 and 8. 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

```

### Pseudo-Infinite Blocks

The use of very large block volume modifier to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for \*CONVERGE.

---

## Null Block Indicator (Optional)

\*NULL

### PURPOSE:

\*NULL indicates the input of an array of null block indicators.

### ARRAY:

\*NULL

### DEFAULTS:

Optional keyword. Default: all blocks are active.

### 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:

Any of the array reading options may be used to designate the location of null blocks within the given grid configuration.

0 = null block

1 = active block.

If the keyword \*NULL is used to designate a null block and a nonzero porosity is assigned to that block with the \*POR keyword, the \*NULL designation overrides the porosity value.

Since block geometry modifier array \*VATYPE also can be used to indicate null blocks, it is recommended that both \*NULL and \*VATYPE not appear in the same data set. If both partial and null blocks are being specified, use only \*VATYPE.

If a dual porosity model is being used, selective participation of the two porosities can be controlled with the \*NULL keyword. Using \*NULL with no \*MATRIX or \*FRACTURE qualifier nulls the block (including both porosities) and makes it a barrier to flow. Nulling the block once with one of \*MATRIX or \*FRACTURE, and then again with the other qualifier has the same effect. Using \*NULL \*MATRIX and setting each of \*DIFRAC, \*DJFRAC, and \*DKFRAC equal to 0 for a block accomplishes the same task. Using just \*NULL \*MATRIX or \*NULL \*FRACTURE makes only one of the porosities non-participating. Flow can occur to the other porosity as required.

Note that a pinched out status set by the \*PINCHOUTARRAY keyword over-rides \*NULL settings. See the description of \*PINCHOUTARRAY following. \*NULL settings over-ride pinch out setting generated by use of the \*PINCHOUT-TOL keyword, or zero thickness situations.

---

## Dual Porosity (Optional)

\*DUALPOR

### PURPOSE:

\*DUALPOR indicates the use of a dual porosity 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, \*RANGE, \*NULL and \*POR keywords.

Only one of \*DUALPOR, \*DUALPERM, \*SUBDOMAIN or \*MINC may be specified.

\*DUALPOR option cannot be used with the \*NINEPOINT or \*WELLBORE options.

\*DUALPOR may be used with local grid refinement \*REFINE.

### EXPLANATION:

This keyword indicates that a dual porosity option will 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 qualifier while fracture properties are described using the \*FRACTURE qualifier.

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 is calculated when \*DUALPOR is specified. These flows are governed either by the matrix or matrix-fracture properties depending on the choice of the shape factor calculation (see also \*SHAPE keyword).

Thus, \*DUALPOR allows one matrix porosity and one fracture porosity per grid block, where the matrix is connected only to the fracture in the same grid block. Fracture porosities are connected to other neighboring 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). Property definition for \*DUALPOR systems usually requires the use of pairs of definitions for most items, one carrying a \*MATRIX qualifier and the other a \*FRACTURE qualifier. Further details are explained in the descriptions for the individual properties.

Inter-block heat conduction uses the same connectivity as fluid flow, so there is no inter-matrix heat conduction. If you need a natural fracture grid type and inter-matrix heat conduction is important, use the \*DUALPERM option instead.

See J. R. Gilman and H. Kazemi, "Improvements in Simulation of Naturally Fractured Reservoirs", SPE10511 for further details. See also Appendix E.8.

---

## Dual Permeability (Optional)

\*DUALPERM

### PURPOSE:

\*DUALPERM indicates the use of a dual porosity 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, \*RANGE, \*NULL and \*POR keywords.

Only one of \*DUALPOR, \*DUALPERM, \*SUBDOMAIN or \*MINC may be specified.

\*DUALPERM option cannot be used with the \*NINEPOINT or \*WELLBORE options.

\*DUALPERM may be used with local grid refinement \*REFINE.

### EXPLANATION:

The 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 (convection and conduction) 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 neighboring fracture porosities, and the same holds true for neighboring matrix porosities.

Property definition for \*DUALPERM systems usually requires the use of pairs of definitions for most items, one carrying a \*MATRIX qualifier and the other a \*FRACTURE qualifier. Further details are explained in the descriptions for the individual properties.

Dual permeability 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). See also Appendix E.8.

---

## Dual Porosity Subdomain Method (Optional)

\*SUBDOMAIN

### PURPOSE:

\*SUBDOMAIN indicates the use of a dual porosity model using the subdomain method.

### FORMAT:

```
*SUBDOMAIN idiv
  ( *FRACVOL vol(1) ... vol(idiv) )
  ( *SGMETHOD (*AVERAGE | *ELEMENT ) )
```

### DEFINITIONS:

idiv

Subdivide each matrix block into idiv subblocks. Typical values for idiv range from 2 to 5.

vol(i)

Volume fraction of matrix subblock i within the matrix volume of the grid block. These volume fractions must sum to 1. Subblock i=1 is the bottommost one for \*KDIR \*UP and the topmost one for \*KDIR \*DOWN.

\*SGMETHOD

Specify the length scale used to calculate phase segregation and pressure adjustment in the fracture block.

\*AVERAGE – Length scale is the undivided matrix block, which is vertical block size (specified via \*DK) minus fracture width. This was the method always used by STARS before \*SGMETHOD became available.

\*ELEMENT – Length scale is the elemental matrix portion, which is vertical fracture spacing (specified via \*DKFRAC) minus fracture width. This was the method always used by IMEX and GEM before \*SGMETHOD became available.

### DEFAULTS:

Optional keyword. No default.

If \*SUBDOMAIN is present but \*FRACVOL is absent, vol(i) = 1/idiv for all subblocks i.

If \*SUBDOMAIN is present but \*SGMETHOD is absent, the result depends upon keyword \*TRANSD (which enables the SUBDOMAIN-DK option):

- if \*TRANSD is absent, \*SGMETHOD \*AVERAGE is assumed;
- if \*TRANSD is present, \*SGMETHOD \*ELEMENT is assumed.

## **CONDITIONS:**

This keyword must be located in the RESERVOIR DESCRIPTION keyword group, before the \*NULL and \*POR keywords.

Only one of \*DUALPOR, \*DUALPERM, \*SUBDOMAIN or \*MINC may be specified.

\*SUBDOMAIN may not be used together with the \*NINEPOINT, \*WELLBORE or \*REFINE options.

## **EXPLANATION:**

The description given above for the \*DUALPOR keyword should be studied first, as this option is closely related.

This option allows each simulator block to have up to two porosity systems, one called the matrix porosity and the other called the fracture porosity. Moreover, the \*SUBDOMAIN option splits up the matrix porosity vertically into "idiv" segments with a thickness depending on \*FRACVOL values. Inter-block fracture to fracture, as well as matrix to fracture and matrix to matrix flows within a block are calculated. The \*SUBDOMAIN method models vertical gradients (pressure, temperature, etc.) within the matrix porosity of a block.

Each subblock has a porosity value and permeabilities, as well as other distinct properties. Matrix properties are described using the \*MATRIX qualifier while fracture properties are described using the \*FRACTURE qualifier. Further details are explained in the descriptions for the various properties.

For details of the \*SUBDOMAIN method, 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. See also Appendix E.8.

### **\*FRACVOL for SUBDOMAIN-DK option**

Specifying non-zero values for \*TRANSD (see below) enables the SUBDOMAIN-DK feature which connects the subdomain matrix stacks of vertically adjacent blocks. Also, non-zero values for \*SD\_REINF (Rock-Fluid Data section) enable the re-infiltration option for those interblock connections. In this case the most correct capillary holdup is obtained by specifying the smallest possible fractional volume for the bottommost subblock, on the order of the fracture width. Practically, that subblock size should not be so small that CPU time increases significantly.

---

## SUBDOMAIN-DK Transmissibility Multiplier (Optional)

\*TRANSD

### PURPOSE:

Enable the SUBDOMAIN-DK option and specify inter-subdomain transmissibility multipliers.

### ARRAY:

\*TRANSD

### DEFAULTS:

If \*TRANSD is absent, all inter-subdomain transmissibility multipliers are assumed to be zero, effectively disabling the SUBDOMAIN-DK option.

### CONDITIONS:

Keyword \*TRANSD is meaningful and effective only when used together with keyword \*SUBDOMAIN.

Array qualifiers \*MATRIX and \*FRACTURE are not allowed.

### EXPLANATION:

Keyword \*TRANSD enables the **SUBDOMAIN-DK** model, adding to the \*SUBDOMAIN model an additional interblock connection vertically across the horizontal fracture plane. Keyword \*TRANSD also allows you to control the contact effectiveness of that new connection via fluid transmissibility multiplier.

Values entered for \*TRANSD are dimensionless and must be non-negative. A \*TRANSD value of one results in a transmissibility similar to a matrix-matrix connection within a local matrix stack. A value of zero will remove that newly created connection completely.

A \*TRANSD value entered for block (i,j,k) will be applied to the new connection between the matrix subblock stacks in blocks (i,j,k) and (i,j,k+1).

### SUBDOMAIN-DK Model

In the standard **SUBDOMAIN** natural fracture model, the fracture and matrix are modelled as two separate grid systems. For a single spatial block, the fracture is modelled using a single block while the matrix is modelled using a vertical stack of subblocks. Flow between matrix subblocks occurs only within the local stack and not between adjacent block stacks.

Keyword \*TRANSD enables the **SUBDOMAIN-DK** model, creating an additional interblock connection vertically across the horizontal fracture plane. Specifically, for a vertically adjacent pair of matrix subblock stacks, the lowest subblock in the upper stack is connected to the highest subblock in the lower stack. This allows fluid flow (e.g., gravity drainage) to occur directly from matrix to matrix across a horizontal fracture. The “DK” part of this model’s name comes from the Dual Permeability (DK) model which also connects matrix blocks across a fracture plane.

Generally for the SUBDOMAIN-DK option there is full capillary continuity between matrix grids (across the horizontal fracture plane). Use keyword \*SD\_REINF in the Rock-Fluid Data section to control the continuity level of capillary pressure, from full continuity to complete discontinuity.

The SUBDOMAIN-DK option uses \*SGMETHOD \*ELEMENT, independent of the user input choice of \*SGMETHOD.

Use keyword \*SD\_HCMULT in the Other Reservoir Properties section to adjust the thermal conductive flow across the SUBDOMAIN-DK connection.

For further details of the SUBDOMAIN-DK model, see "Simulation Gravity Drainage and Reinfiltration with a Subdomain-Dual Permeability Hybrid Fracture Model", SPE 106191 by Barry Rubin.

#### **Examples:**

The following provides two examples of \*TRANSD usage:

```
** Enable direct gravity drainage between grids for all columns
*SUBDOMAIN 4
*TRANSD *CON 1.0

** Enable direction connection only between (2,3,6) and (2,3,7)
*SUBDOMAIN 3
*TRANSD *IJK 2 3 6 1.0
```

---

## 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
  *FRACVOL vol(1)....vol(idiv)
```

### DEFINITION:

idiv

Number of subdivisions for each matrix blocks, typically chosen from the range of 2 to 5.

vol(i)

Volume fraction of matrix element i within the matrix volume of the grid block. These volume fractions should sum to 1. The fraction i=1 corresponds to the innermost element.

### DEFAULT:

Optional keyword. No default.

### CONDITIONS:

This keyword must be located in the RESERVOIR DESCRIPTION keyword group, before the \*REFINE, \*RANGE, \*NULL and \*POR keywords.

Only one of \*DUALPOR, \*DUALPERM, \*SUBDOMAIN or \*MINC may be specified.

\*MINC option cannot be used with the \*NINEPOINT, \*WELLBORE or \*REFINE options.

### EXPLANATION:

The description given above for the \*DUALPOR keyword should be studied first, as this option is closely related.

This option allows each simulator block to have up to two porosity systems, one called the matrix porosity and the other called the fracture porosity. Moreover, the \*MINC option splits up the matrix porosity into "idiv" nested rings according to \*FRACVOL values. Inter-block fracture to fracture, and matrix to fracture flows within a block, are calculated. Also, matrix to matrix flows 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 block has a porosity value and its own permeabilities, as well as other distinct properties. Matrix properties are described using the \*MATRIX qualifier while fracture properties are described using the \*FRACTURE qualifier. 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. See also Appendix E.8.

## **Shape Factor Calculation (Conditional)**

**\*SHAPE**

### **PURPOSE:**

\*SHAPE describes the method (which type of shape factor) is to be used in calculating matrix-fracture flow within a naturally fractured block.

### **FORMAT:**

\*SHAPE ( \*GK | \*K-HARMONIC | \*PI2 )

### **DEFINITION:**

#### **\*GK**

This sub-keyword indicates the use of a Gilman and Kazemi style formulation for the shape factor.

#### **\*K-HARMONIC**

This sub-keyword indicates the use of a harmonic fracture and matrix permeability average in the shape factor calculation.

#### **\*PI2**

This sub-keyword specifies the use of  $\pi^2$  (9.87) instead of 4 in the \*GK style formula for shape factor. See references below.

### **DEFAULT:**

Conditional keyword. Default: \*SHAPE \*GK

### **CONDITIONS:**

These keywords must be in the RESERVOIR DESCRIPTION keyword group before the \*NULL and \*POR keywords.

### **EXPLANATION:**

As a general rule, fluid flow between small porous regions is proportional to transmissibility. The inverse of transmissibility is the sum of fluid resistance and associated geometry in each direction. When the element size determined by \*DIFRAC, \*DJFRAC and \*DKFRAC is not the same as a grid block size then the fracture-matrix transfer term must be scaled up/down by multiplying it with a ratio of block and element volume. There are different formulas in the literature for calculation of this fracture-matrix transfer term. All of them are derived from the formulas for flow in the porous media (mentioned above) but have different assumptions about the effect of fracture and matrix permeability.

The Gilman and Kazemi formulation is:

$$G \& K = 4V_b \sum_i \frac{k_{mi}}{L_i^2}$$

$L_i$  is the fracture spacing in x, y and z direction

$k_{mi}$  is the effective matrix permeability in all directions

$V_b$  is a block volume

The K-Harmonic formulation is:

$$K - H = 4V_p \sum_i \left[ \frac{1}{L_i} \left( \frac{k_{f_i} k_{m_i}}{L_{f_i} k_{m_i} A_{f_i} / A_{m_i} + L_{m_i} k_{f_i}} \right) \right]$$

$L_{fi}$  is the fracture width in all directions

$L_{mi}$  is the matrix size in all directions

$k_{fi}$  is the effective fracture permeability in all directions

$A_{fi}$  and  $A_{mi}$  is the fracture/matrix area perpendicular to the flow

Further details and especially the discussion about effective and intrinsic values are in the references noted before.

The K- Harmonic calculation is more general because it does not assume that the fracture permeability is much higher than the matrix permeability. Both of these calculations incorporate various anisotropies. When fracture spacing is equal to zero in certain direction then fracture width is equal to zero and flow between fracture and matrix is zero for that direction.

Note that the basic transmissibility formulae are used even when the fracture spacing exceeds the grid block size. These cases correspond to dividing up the matrix to fracture flow over several grid blocks and modelling the usual matrix to fracture flows over the individual blocks. Also, the basic formulae are used even when the grid block containing the matrix regions is itself not cubic in form (such as for corner point, radial or hybrid refined grids).

### References for \*PI2 Option

Mora, C.A., Wattenbarger, R.R., "Analysis and Verification of Dual Porosity and CBM Shape Factors", JCPT, February 2009, Vol. 48, No. 2, pp. 17-21.

Van Heel, A.P.G., Boerrigter, P.M., van Dorp, J.J., "Thermal and Hydraulic Matrix-Fracture Interaction in Dual-Permeability Simulation," SPE-REE, August 2008, Vol. 11, No., 4, pp. 735-749.

## **Fracture Spacing (Conditional)**

**\*DIFRAC, \*DJFRAC, \*DKFRAC**

### **PURPOSE:**

\*DIFRAC indicates the input of the fracture spacing in the I direction.

\*DJFRAC indicates the input of the fracture spacing in the J direction.

\*DKFRAC indicates the input of the fracture spacing in the K direction.

### **ARRAY:**

\*DIFRAC  
\*DJFRAC  
\*DKFRAC

### **DEFAULTS:**

Absence of the keyword implies that all grid blocks have zero fracture spacing (no fracture) in that direction. If some blocks have fractures and some don't, enter zero for the unfractured blocks. Setting values to 0 corresponds to an infinite, and hence ineffective, spacing in that direction.

Fracture spacing corresponding to block size in a specified direction will be assigned when \*DIFRAC, \*DJFRAC or \*DKFRAC have negative value.

### **CONDITIONS:**

These keywords must be in the RESERVOIR DESCRIPTION keyword group before the \*NULL and \*POR keywords.

Array qualifiers \*MATRIX and \*FRACTURE must not be used with these keywords.

These keywords are used in conjunction with the natural fracture options \*DUALPOR, \*DUALPERM, \*MINC and \*SUBDOMAIN.

Setting one of the fracture spacings to 0 indicates that there is no fracture plane perpendicular to that axis. 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 together with fracture volume are used to calculate fracture and matrix sizes in each direction. These values are used calculate the matrix to fracture transfer coefficient as well as other geometry parameters (e.g. matrix and fracture block volume). See detailed description in Appendix E.8.

Fracture spacings should be measured from center line to center line in the appropriate direction. The unit is (m | ft | cm). The basic transmissibility formulas (see Appendix E.8) are applied even when the fracture spacings exceed the grid block size. It means that all the blocks contained in the fractured element will be fractured. The specified fracture volume will be distributed among all blocks. If this is a concern, then reservoirs with fracture spacing spanning several blocks should be modeled as a single porosity problem with implicit fracture blocks. \*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.

### **Zero Fracture Spacing and Child Grid Inheritance**

Generally each cell in a naturally fractured grid is split into separate matrix and fracture cells. When a cell has zero fracture spacing in all three directions, the corresponding fracture cell is deemed to be null. If that same cell is locally refined into child cells, each corresponding child fracture cell is deemed to be null as well and fracture spacing specified for it is ignored.

### **Converting IMEX keywords \*SHAPE and \*TRANSFER**

STARS does not support the IMEX keywords \*SHAPE \*WR and \*TRANSFER. If \*SUBDOMAIN is in effect then \*TRANSFER 1 is assumed; otherwise, \*TRANSFER 0 is assumed.

The acceptable range of values for fracture spacing is:

	<b>SI m</b>	<b>Field ft</b>	<b>Lab cm</b>
min	0.0	0.0	0.0
max	1.0E+4	32,808.0	1.0E+6

Fracture spacing has a large effect on the values of effective porosity and permeability to enter for the fracture.

---

## Fracture Definition (Conditional)

\*FRFRAC, \*FORMINFRAC

### PURPOSE:

Assign the fracture volume fraction and the rock-in-fracture fraction.

### ARRAY:

\*FRFRAC  
\*FORMINFRAC

### DEFINITIONS:

#### \*FRFRAC

Specify the fracture volume in an element as a fraction of the gross volume. When a fracture does not contain rock then this value is equal to the effective fracture porosity.

#### \*FORMINFRAC

Specify what fraction of the fracture volume is rock (formation). If this value is zero then the fracture consists entirely of open void space and the intrinsic fracture porosity is 1. When this value is non-zero the intrinsic fracture porosity is less than 1 and the rock (formation) intrinsic porosity is specified via \*POR \*FRACTURE.

### DEFAULTS:

If keyword \*FRFRAC is absent then the fracture does not contain rock, in which case the effective fracture porosity must be specified via \*POR \*FRACTURE.

If keyword \*FORMINFRAC is absent then the fracture contains no rock (formation).

### CONDITIONS:

These keywords must be in the RESERVOIR DESCRIPTION keyword group before the \*NULL and \*POR keywords.

Array qualifiers \*MATRIX and \*FRACTURE must not be used with these keywords.

\*FORMINFRAC may not be used without the \*FRFRAC keyword.

### EXPLANATION:

The natural fracture options \*DUALPOR, etc., split a cell gross volume into two distinct regions: (1) a fracture cell, and (2) a matrix cell or cell group. For isothermal applications it is natural to identify the fracture cell only with the open void fracture space. In this case the fracture cell intrinsic porosity (fracture void volume over fracture cell volume) is always one. Therefore, early implementations used the normal fracture porosity input facility (\*POR \*FRACTURE) to specify “effective” fracture porosity defined as the fracture void volume over the sum of the matrix and fracture cell volumes. This definition leads to small values of effective fracture porosity, e.g., 0.01 or 0.001.

In thermal applications the formation immediately adjacent to the fracture takes heat from the fluid in the fracture via conduction, at a field time scale. The modeling of this effect can be very important in predicting the propagation of heat fronts, e.g., steam breakthrough. Therefore, in thermal applications it is common to include some part of the formation with the fracture cell, assuming that some of the formation will have a temperature close to the fracture fluid value. However, this is not possible with the “effective” fracture porosity technique described above.

Instead, keyword \*FRFRAC lets you specify what fraction of the original gross volume should be defined as the fracture region, and keyword \*FORMINFRAC lets you specify how much of each fracture region is actually formation. In this case, \*POR \*FRACTURE specifies the intrinsic porosity of the formation portion of a fracture cell. Also in this case, the value reported for intrinsic fracture porosity (fracture void volume over fracture cell volume) is

$$\phi_f = 1 - F_{fr} \cdot (1 - \phi_{fr})$$

where  $F_{fr}$  is specified via \*FORMINFRAC and  $\phi_{fr}$  is the formation porosity specified via \*POR \*FRACTURE. For example, if  $F_{fr} = 0.3$  and  $\phi_{fr} = 0.25$  then the reported intrinsic fracture porosity will be  $\phi_f = 0.775$ . This intrinsic value is then used in calculation of block heat capacities and thermal conductivities.

See detailed description in Appendix E.8.

### Varying \*FORMINFRAC

For a thermal application it is common to use \*FORMINFRAC as a matching or sensitivity parameter. You can vary \*FORMINFRAC and \*FRFRAC in a way that retains both the formation pore volume and the fracture void volume. Let  $\phi_r$  be the intrinsic formation porosity specified by \*POR for both the matrix and fracture regions, let  $F_f$  be the \*FRFRAC value, let  $F_{fr}$  be the \*FORMINFRAC value and let  $V$  be the gross volume (matrix plus fracture), all of one block. The fracture void volume is

$$V \cdot F_f \cdot (1 - F_{fr})$$

and the total (matrix plus fracture) formation pore volume is

$$V \cdot (1 - F_f) \cdot \phi_r + V \cdot F_f \cdot F_{fr} \cdot \phi_r = V \cdot \phi_r \cdot [1 - F_f \cdot (1 - F_{fr})]$$

Each of these quantities contains the quantity  $F_f \cdot (1 - F_{fr})$ . Therefore, you can change  $F_f$  or  $F_{fr}$  and preserve those pore volumes as long as  $F_f \cdot (1 - F_{fr})$  is unchanged. If matrix and fracture cells have the same conditions (e.g., at initial conditions) then component amounts and heat content are preserved as well. For example, when  $F_f = 0.003$  and  $F_{fr} = 0$  these pore volumes are  $0.003V$  and  $0.997\phi_r V$ . Suppose you wish to compare this with a similar case in which you include on each side of the fracture void space an amount of formation corresponding to twice the fracture width, that is, 80% of the fracture region is formation. To increase  $F_{fr}$  to 0.8 while preserving these pore volumes, change  $F_f$  to  $0.003 / (1 - 0.8) = 0.015$ .

---

## **Discretized Wellbore (Conditional)**

\*WELBORE, \*RELROUGH,

\*LAMINAR, \*TRANSIENT, \*CIRCWELL, \*WELLINFO, \*REGIME, \*WELLWALL,  
\*TUBINSUL, \*ANNULUSWAL, \*CASING, \*FILM\_COND, \*RANGE, \*WBZ, \*WBZADJ

### **PURPOSE:**

Define wells which are to be discretized. Discretized wellbore may be specified also in recurrent data via keyword \*WELBORE-REC.

### **FORMAT:**

```
*WELBORE rw (*RELROUGH relrof)
*LAMINAR
*TRANSIENT (*ON | *OFF)
*CIRCWELL ra i j k nwbwt (*RELROUGH relrof)
*WELLINFO
*REGIME

*WELLWALL      rwo      hcww
*TUBINSUL      rins     hcins    nwbwin
*ANNULUSWAL    rao      hcaw
*CASING        rcas     hecas   nwbwca
*FILM_COND

*RANGE          i1(:i2)  j1(:j2)  k1(:k2)
                ( i1(:i2)  j1(:j2)  k1(:k2) )
*WBZ            z(1) ... z(nlayer)
-or-
*WBZADJ        dz(1) ... dz(nlayer)
```

### **DEFINITIONS:**

#### **\*WELBORE rw**

Indicates that a discretized well will be defined. Each discretized wellbore needs its own \*WELBORE keyword. Quantity “rw” is the inside well radius (m | ft | cm), or inside tubing radius when well is circulating.

#### **\*RELROUGH relrof**

Relative roughness values for a well (tubing).

#### **\*LAMINAR**

Forces the wellbore flow to be in laminar mode so that the flow correlations are not used. Use this keyword for vertical or deviated wells, or runs where counter-current flow is present.

**\*TRANSIENT**

\*ON: Indicates that the transient behavior in a wellbore will be simulated.

\*OFF: Wellbore will be initialized to pseudo-steady state.

This keyword may be also used in WELL DATA Section.

**\*CIRCWELL**

Indicates additional information for a circulating well. If \*CIRCWELL is used, it must appear before \*RANGE.

ra

Annulus inside radius (m | ft | cm), which must be greater than the tubing radius rw.

i j k

I-J-K address of the grid block which defines the downhole end of the well (toe). This block must be at one end of the well structure defined by \*RANGE.

nwbwt

Number of sections (blocks) in the circulating well that do not contain tubing. This, together with the I-J-K address of the downhole end, indicates which wellbore sections will not contain tubing.

**\*RELROUGH relrof**

Relative roughness values for an annulus.

**\*WELLINFO**

Flags printing of detailed wellbore information.

**\*REGIME**

This keyword indicates that another method for friction pressure drop calculation will be used. It first evaluates the flow regime and then calculates friction pressure drop and liquid holdup accordingly.

**\*WELLWALL**

This keyword indicates that parameters for tubing (wellbore) wall will be defined. A non-zero wall thermal conductivity may be specified using either (1) \*WELLWALL or (2) *thconr* with \*ROCKTYPE and \*THTYPE \*WELBORE, but not both.

rwo

Tubing (wellbore) outside radius (m | ft | cm), which must not be less than tubing (wellbore) inner radius rw.

hcww	Tubing (wellbore) wall heat conductivity (J/m-day-C   Btu/ft-day-F   J/cm-min-C).
<b>*TUBINSUL</b>	This keyword indicates that parameters for tubing insulation will be entered.
rins	Tubing insulation outer radius (m   ft   cm), which must not be less than tubing outer radius rwo.
hcins	Heat conductivity of tubing insulation (J/m-day-C   Btu/ft-day-F   J/cm-min-C).
nwbwin	Number of tubing grid blocks in a discretized well without insulation (partial tubing insulation). When tubing is shorter than annulus, indicate only the number of tubing blocks that are not insulated.
<b>*ANNULUSWAL</b>	This keyword indicates that parameters for annulus wall will be entered.
rao	Annulus wall outer radius (m   ft   cm), which must not be less than the annulus wall inner radius ra.
hcaw	Heat conductivity of annulus wall (J/m-day-C   Btu/ft-day-F   J/cm-min-C).
<b>*CASING</b>	This keyword indicates that parameters for casing will be entered.
rcas	Casing outer radius (m   ft   cm), which must not be less than the annulus wall outer radius ra.
hccas	Heat conductivity of a casing (J/m-day-C   Btu/ft-day-F   J/cm-min-C).
nwbwca	Number of grid blocks in a discretized well without casing.
<b>*FILM_COND</b>	Indicates that heat transfer through a fluid film will be calculated. This parameter together with heat conduction through walls, insulation, etc. is used in calculation of the overall heat transfer coefficient. This heat transfer coefficient does not include heat transfer by radiation.

NOTE: Dimensionless parameters such as Reynolds, Prandtl, Nusselt and Grashof number are used in evaluation of heat transfer through the fluid film. Therefore, input values for heat capacities, viscosities and heat conductivities must be correct for each component and phase. Specifically, for heat conductivities do not use a single average value for water, oil and gas phases.

**\*RANGE**

Indicates the addresses of grid blocks through which the wellbore penetrates. All discretized wellbores require the first address line; a deviated well requires the second line as well. Each address line must indicate a range in exactly one direction. The total number of blocks penetrated must not exceed the dimension limit for well layers.

This keyword defines only the blocks which contain the discretized wellbore. For a horizontal wellbore the end that is connected to the surface will be determined by the perforation keywords in the well data section.

For a deviated wellbore the two ranges defined by the two \*RANGE lines must have exactly one block in common, which also must be at one end of each range.

If \*CIRCWELL is present, \*RANGE must appear after \*CIRCWELL.

**i1(:i2)**

I direction index or range for well location.

**j1(:j2)**

J direction index or range for well location.

**k1(:k2)**

K direction index or range for well location.

**\*WBZ**

Indicates that wellbore depth will be redefined. This option is useful when the grid depth varies (that is, you used \*DTOP) and you want the wellbore depth to be constant or nearly constant. See explanation 'Depth Adjustments', below.

**z(i)**

Block centre depth for wellbore interval i (m | ft | cm). Enter a value for each block addressed by \*RANGE, in the order given by \*RANGE. The wellbore block centre depth must not be different from the parent block centre depth by more than one half of the block size in the vertical direction, so that assumptions required by the transmissibility (well index) calculation still apply.

**\*WBZADJ**

Indicates that wellbore depth will be adjusted. This option is useful when the grid depth is constant (that is, you did not use \*DTOP) and you want the wellbore depth to vary along its length.

dz(i)

Block centre depth adjustment for wellbore interval i (m | ft | cm). Enter a value for each block addressed by \*RANGE, in the order given by \*RANGE. dz must not exceed one half of the block size in the vertical direction, so that assumptions required by the transmissibility (well index) calculation still apply.

### DEFAULTS:

If \*WELLBORE is absent, then no discretized wells will be defined.

If \*RELROUGH is absent, then the relative roughness is set to 0.0001.

If \*LAMINAR is absent the Reynolds number is calculated every timestep, and when flow becomes turbulent the appropriate slip between liquid and gas phase as well as friction pressure drop is applied.

If \*TRANSIENT is absent, wellbore is set to 'pseudo steady-state conditions' initially or at every true well change if not specified differently. If \*TRANSIENT appears without \*ON or \*OFF, then \*ON is assumed.

If \*CIRCWELL is absent after \*WELLBORE, the well will not contain a tubing string.

If \*WELLINFO is absent, wellbore parameters are not printed out.

If \*REGIME is absent, friction pressure drop is calculated using the Dukler correlation and liquid holdup is evaluated from Bankoff's correlation. **NOTE:** Keyword \*LAMINAR overrides \*REGIME so that friction pressure drop and liquid holdup are not calculated.

If \*WELLWALL is absent, the tubing (wellbore) outer radius is equal to the tubing inner radius. The result is a tubing wall with zero thickness that provides no additional resistance to heat flow.

If \*TUBINSUL is absent, the tubing insulation outer radius is equal to the tubing outer radius. The result is a tubing insulation with zero thickness that provides no additional resistance to heat flow.

If \*ANNULUSWAL is absent, the annulus outer radius is equal to the annulus inner radius. The result is an annulus wall with zero thickness that provides no additional resistance to heat flow.

If \*CASING is absent, the casing outer radius is equal to the annulus outer radius. The result is a casing wall with zero thickness that provides no additional resistance to heat flow.

If \*FILM\_COND is absent, heat transfer does not account for the presence of fluid film.

If \*WBZ and \*WBZADJ are absent, a wellbore block has the same depth as the centre of the grid block that contains it.

### CONDITIONS:

If \*WELLBORE is present, then \*RANGE must be present also.

If \*CIRCWELL is present, it must appear before \*RANGE.

The discretized wellbore option may not be used in a naturally fractured grid (keywords \*DUALPOR, \*DUALPERM, \*SUBDOMAIN and \*MINC).

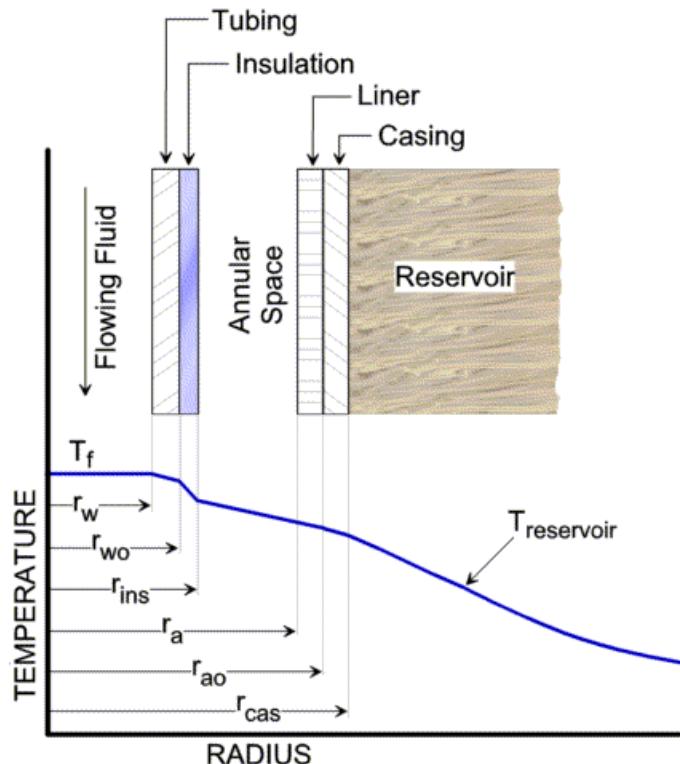
A non-zero wall thermal conductivity may be specified using either (1) \*WELLWALL or (2) *thconr* with \*ROCKTYPE and \*THTYPE \*WELLBORE, but not both.

The discretized wellbore option may not be used in blocks that are locally refined without the \*HYBRID option of \*REFINE.

When defining a discretized wellbore inside a hybrid grid, the hybrid grid must be defined first. See explanation in the option summary at the beginning of this chapter.

The radii described above must have values in the following increasing sequence. The defaults for *rwo*, *rins*, *rao* and *rcas* automatically satisfy this sequence.

$$rw \leq rwo \leq rins < ra \leq rao \leq rcas$$



## EXPLANATION:

### Aspects of the Discretized Wellbore

For some detailed discussion of this option, see Appendix A.7.

The method of modelling well flow more accurately by discretizing the wellbore separately can be viewed as both a well option and an advanced grid option. The part of the well modelled with grid blocks or "discretized" is defined via the above keywords; block volumes and inter block transmissibilities are calculated as for any other block.

The part of the well not "discretized" is defined in the well data section via \*WELL in the same manner as before, with only one 'layer' which is connected to one end of the discretized part of the well. This source/sink well provides an entrance/exit for the discretized wellbore blocks, and controls several aspects (e.g., initialization) of the wellbore. Several source/sink wells may be attached to a discretized wellbore at various times (e.g., to switch from injection to production) but only one may be attached at a time.

It is common that grid properties and initial conditions are different in the wellbore. When the keyword \*TRANSIENT is not used the simulator will calculate 'pseudo steady-state' conditions for the well-bore. This will improve the numerical performance. However, in some cases the transient behavior in the wellbore is of interest (injection or flow of viscous macromolecules) and therefore the wellbore proper-ties should be set appropriately by the user in the INITIAL CONDITIONS section. See the array input options \*WELLBORE or \*RG. In the output, conditions in the wellbore penetrating a block will have a 'WB' appended, e.g., 1,1,1 + WB or 1,1,1 / 1,1,1.

Example: Horizontal producer as well #1, attached to the surface at block (1,1,1).

```
** Reservoir Definition Section
wellbore 0.15
    range 1:4 1 1

** Recurrent and Well Section
well 1 'Producer 1'
producer 1
operate bhp 154
operate max liquid 80000
perf 1 ** i j k wi
    1 1 1 wb 50
```

Two different methods are used to calculate the friction pressure drop and liquid holdup in the wellbore. The first method uses Bankoff's correlation to evaluate liquid holdup and Dukler's correlation to calculate friction pressure drop. These correlations are valid only for co-current vertical upward or horizontal flow. This method was the only one in versions before 98.00, and currently is the default. A more detailed description can be found in "Aspects of Discretized Wellbore Modelling Coupled to Compositional/Thermal Simulation", V. Oballa, D.A. Coombe, W.L. Buchanan, JCPT, April 1997, Volume 36, No. 4, page 45.

The second method (invoked with keyword \*REGIME) calculates friction pressure drop and liquid holdup according to a flow regime existing in the wellbore. These correlations are valid only for co-current flow. This method is based on "A Comprehensive Mechanistic Model for Two-Phase Flow in Pipelines", J.J. Xiao, O. Shoham, J.P. Brill, Proceedings from 65th Annual Technical Conference of SPE, September 23-26, 1990, New Orleans, USA, SPE 20631.

### Circulating Well

A circulating well is just a discretized wellbore with a second independent flow string in it. The injection from surface is attached to one end of the tubing, and the injection stream flows into the open wellbore at the end of the tubing. The resulting annulus stream flows into the reservoir through the perforations, and excess annulus stream is produced to surface. Each discretized wellbore stream requires that a source/sink well be attached to it in the well data section.

Properties and initial conditions can be assigned to the annulus and tubing alone using the array input qualifiers \*ANNULUS and \*TUBING; the qualifier \*WELLBORE refers to both annulus and tubing. In the output, conditions in the annulus and tubing are denoted with 'WB' and 'TU', respectively, e.g., 1,1,1 WB and 1,1,1 TU.

Example: Circulating well #1 and #2, attached to block (1,1,1); full length tubing; high initial temperature in tubing.

```
** Reservoir Definition Section
wellbore 0.15
    circwell 0.4 4 1 2 0
    range 1:4 1 2
** Recurrent and Well Section
well 1 'TUBING'
injector mobweight 1
operate bhp 155
operate max water 80000
tinjw 355 qual .7
perf 1 ** i j k wi
    1 1 2 tu 241.3
well 2 'ANNULUS'
producer 2
operate bhp 154
operate max water 80000
perf 2 ** i j k wi
    1 1 2 wb 241.3
```

### Depth Adjustments

The depth adjustment options \*WBZ and \*WBZADJ allow the modelling of undulating (varying depth) wellbore in a constant-depth grid. As well, it allows the modelling of wellbore depth different from the variable grid depth defined by the \*DTOP and \*DK keywords.

\*WBZADJ requires only relative adjustments, and so is easy to use. However, \*WBZ requires the absolute depth which may not be apparent from your data. Whenever \*WELLBORE is used in a run, the block centre depths 'Block Centre from Ref plane' will be printed. It is suggested that you first run the simulator initialization with no \*WBZ keywords and examine the printed depths in the grid blocks of interest. By default, the wellbore depth is the same as the block centre. Then, enter wellbore depths that are different (within half of block thickness) from the block depths. Check the printout for confirmation.

For a circulating well, the annulus and tubing have the same depth.

### Cautionary Note

Each well in a simulation does not necessarily need to be discretized. This option should be used with care, and only when considered necessary for an adequate representation of the process, such as a detailed study of a horizontal well application.

## Wellbore Initialization and Transient Behavior

The initial conditions in the wellbore (tubing, annulus) dictate the length of a transient state. When initial pressure, temperature and composition differ considerably from conditions at which fluid is injected or produced the period of transient behavior may be extended to several days. Depending on the problem this may affect the final physical results (production, pressure, temperature, saturations, etc.).

In addition, attempts to simulate the transient period will change the overall numerical performance in comparison with the sink/source approach where pseudo-steady state is assumed. High pressure, temperature or saturation changes occur due to small wellbore volume. Even in an implicit simulator the timestep size will be fairly small (10e-3 to 10e-4 days, probably smaller for high rates). For example, the worst scenario is to inject steam into a wellbore containing cold oil, which may be the case after primary production. Thus, the well type may be changed instantaneously, but the condition in the discretized part of the well will take time to change.

If one is not interested in the wellbore's transient behavior, the initial conditions should be a pseudo-steady state to avoid a lengthy equilibration period. This is achieved by omitting the keyword \*TRANSIENT.

## Discretized Wellbore in Hybrid Grid

A normal discretized wellbore is connected to the block containing it with only one connection. This is enough for most cases, where the flow is mostly one way (wellbore-to-block or block-to-wellbore) at any one time. The wellbore depth can be adjusted to be different from the block depth so that a fluid head potential difference can be modelled, but it is only one connection and therefore works for flow in only one direction at a time.

In order to model effectively the single-well SAGD (Steam Assisted Gravity Drainage) process, the wellbore needs to be connected directly to blocks above it and below it, allowing steam to rise and liquid to migrate in from the bottom at the same time. This is accomplished by embedding the discretized wellbore inside a hybrid refined grid.

When a fundamental block contains both a discretized wellbore and a hybrid grid, the wellbore completely replaces the innermost hybrid grid block. The discretized wellbore/annulus block connects directly to the next outer hybrid block(s) in the hybrid grid's radial direction.

This option is invoked by defining a hybrid grid in a fundamental block, and then defining a discretized wellbore of the same orientation in the same block. You may have a hybrid grid surrounding any of the wellbore's sections except the corner of a deviated wellbore. If the hybrid grid is refined into multiple blocks in the axial direction, then there will be one discretized wellbore section in each.

You may refer to both the hybrid blocks as well as the wellbore blocks separately and individually. For arrays, use the \*RG array qualifier for hybrid blocks, and array qualifiers \*WELLBORE, \*ANNULUS and \*TUBING for the wellbore blocks. For example, if block (I,J,K) contains a discretized wellbore in a hybrid grid, then use \*RG I J K to refer to the hybrid grid and use \*WELLBORE I J K to refer to the wellbore blocks. For example, to assign relative permeability type #1 to the main grid, #2 to the near-well region in block (3,4,5) and #3 to its embedded wellbore separately, use

```

*KRTYPE *CON 1           ** entire grid
*KRTYPE *RG 3 4 5 *CON 2   ** near-well region (hybrid grid)
*KRTYPE *WELLBORE 3 4 5 *CON 3 ** wellbore or tubing/annulus

```

To attach a source/sink well to one end of a discretized wellbore embedded in a hybrid grid, use \*PERF and the wellbore block's UBA. Using the above example,

```

*PERF *GEO 'Producer 1'      ** attach s/s well to wellbore block
            3 4 5 / 1 1 1 / 1 1 1

```

The following I-J-K block labels will appear in the output (symbols i1, j1, k1, i2, j2 and k3 are integers):

Fundamental block:

i1,j1,k1

Block in hybrid grid:

i1,j1,k1 / i2,j2,k2

Discretized wellbore in fundamental block:

```

i1,j1,k1 / 1,1,1 WB (non-circulating wellbore)
i1,j1,k1 / 1,1,1 TU (tubing in circulating wellbore)
i1,j1,k1 / 2,1,1 WB (annulus in circulating wellbore)

```

Discretized wellbore in innermost block of hybrid grid:

```

i1,j1,k1 / 1,1,k2 / 1,1,1 WB (non-circulating wellbore)
i1,j1,k1 / 1,1,k2 / 1,1,1 TU (tubing in circulating wellbore)
i1,j1,k1 / 1,1,k2 / 2,1,1 WB (annulus in circulating wellbore)

```

Note that the innermost hybrid block always has  $i_2 = j_2 = 1$ .

Depth adjustments entered via \*WBZ and \*WBZADJ will be ignored because the wellbore fits exactly inside the hybrid grid inner block.

### Reporting of Layer Flow Performance

Well layer reports are made available to RESULTS by the \*LAYER option of \*OUTSRF \*WELL and to the .out file by the \*LAYPHASE option of \*OUTPRN \*WELL. In these reports flow to and from a discretized wellbore (DW) block is treated much like a source/sink well layer defined by \*PERF. The following are important points unique to DW layers for reporting purposes.

1. Flow involving a DW block is given a unique label for reporting purposes. Flow between a DW block and its surrounding parent block  $(i,j,k)$  is labeled with the UBA of the DW block, that is, " $i,j,k/n,1,1$ " where  $n = 1$  for non-circulating DW and  $n = 2$  for circulating DW (annulus). An exception is DW in the centre of a hybrid grid with angular divisions, where that flow is labeled with the UBA of each of the 4 hybrid blocks surrounding the DW, that is, " $i,j,k/2,m,1/n,1,1$ " where  $m = 1,2,3,4$  for the angular sections and  $n$  is as above. Flow between tubing and annulus in a circulating DW is labeled with the tubing block UBA, that is " $i,j,k/1,1,1$ " or " $i,j,k/1,1,1/1,1,1$ ", and reported with the tubing well.
2. The sign used in DW reports is such that injection (DW to reservoir) is (+) and production (reservoir to DW) is (-), similar to source/sink wells.

3. Each DW report includes an extra layer that corresponds to the source/sink well attached to it. In RESULTS this layer is labeled with “-S/S-” instead of a UBA. In the .out file this layer is labeled with the UBA of the source/sink well and is indicated as a “Reference Layer” with “\*”.
4. A DW may be attached to one or more source/sink wells which may be any combination of active and shut-in.
5. Each active source/sink well attached to the same DW will include in its layer report the same DW layer performance. This is the case even if two source/sink wells are injecting different phases into a DW.
6. When a DW is not attached to an active source/sink well, that is, all attached wells are shut in, the DW may experience fluid and heat exchange with reservoir blocks and may report these non-zero rates in its layer performance.
7. When a DW is attached to both active and shut-in wells at the same time, each shut-in well will report zero rates for its DW layer performance while each active well will report the actual DW layer performance.

---

## **Porosity (Required)**

**\*POR**

### **PURPOSE:**

\*POR indicates input of porosities.

### **ARRAY:**

\*POR

### **DEFAULTS:**

Required keyword. No defaults.

### **CONDITIONS:**

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

### **EXPLANATION:**

Units are in fractions, dimensionless.

Porosity for wellbore and tubing blocks will be calculated automatically and reported along with the matrix values.

See **Zero-Porosity Blocks** in the introductory section of this chapter. There are two ways to specify a zero-porosity block: (1) specify zero porosity or (2) specify zero permeability in all three directions.

### **Zero Porosity, Isothermal and Child Grid Inheritance**

When a cell has zero porosity in an isothermal setting, the cell is deemed to be null. If that same cell is locally refined into child cells, each corresponding child cell is deemed to be null as well and porosity specified for it is ignored.

### **Reference Porosity**

Porosities entered via \*POR are interpreted as either reference or initial values. See the EXPLANATION for keyword \*PORINTERP in the Other Reservoir Properties data section.

### **Natural Fracture Systems**

There are two scenarios for defining the fracture in a natural fracture system (\*DUALPOR, etc.), depending upon whether or not keyword \*FRFRAC is used. See the EXPLANATION for keywords \*FRFRAC and \*FORMINFRAC. See also Appendix E.8, section “Fracture and Matrix Properties”.

1. \*FRFRAC is used. Keywords \*FRFRAC and \*FORMINFRAC allow you to specify a fracture volume that contains some formation along with the fracture void space. This is required in thermal applications where heat conduction between fluid in the fracture and rock adjacent to the fracture is fast on field time scales. This option allows you to specify normal intrinsic properties for the formation, e.g., porosity and heat capacity. In this case, \*POR \*FRACTURE specifies intrinsic porosity of formation in the fracture cell. Note that the “Fracture Porosity” reported as output is the fracture cell intrinsic porosity (see \*FORMINFRAC). Use of \*FRFRAC is recommended even if there is no formation in fracture, to facilitate the possible conversion to thermal.

2. **\*FRFRAC** is not used. Early implementations of natural fracture options assumed that there is no rock (formation) associated with the fracture cell. This implies that the intrinsic fracture porosity is always 1. Instead, \*POR \*FRACTURE is assumed to be “effective” fracture porosity, that is, fracture volume over the sum of matrix and fracture cell volumes. This data entry option is considered obsolete for thermal applications but is retained for compatibility with pre-existing data sets.

\*POR \*MATRIX is the intrinsic formation porosity, that is, fraction of void space in a piece of un-fractured matrix material examined independently of any fractures.

If \*FRFRAC is absent, a value zero for \*POR \*FRACTURE indicates that the block is not fractured. A value of zero for \*POR \*MATRIX indicates a zero matrix porosity (no pore space) with no fracture.

### Example #1

A simple 5 x 3 x 1 grid is used to model a naturally fractured system. Only blocks with j = 2 are fractured. Block (1,1,1) has no pore volume. There is no formation in fracture.

```
*POR *MATRIX *CON 0.16
  *MOD 1 1 1 = 0.0    ** No pore space, no fracture
*POR *FRACTURE *CON 0  ** No fracture
  *MOD 1:5 2 1 = 0.01 ** Only blocks j=2 have fracture
```

## Permeabilities (Required)

\*PERMI, \*PERMJ, \*PERMK

### PURPOSE:

- \*PERMI indicates input of I direction permeability.
- \*PERMJ indicates input of J direction permeability.
- \*PERMK indicates input of K direction permeability.

### ARRAY:

\*PERMI  
\*PERMJ  
\*PERMK

### DEFAULTS:

Required keywords. No defaults.

### CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group after the \*NULL and \*POR keywords.

### EXPLANATION:

See **Zero-Porosity Blocks** in the introductory section of this chapter. Note that a zero value for porosity or permeability may indicate a zero-porosity block. For permeability to indicate a zero-porosity block, the permeability in all resolved directions must be zero. A resolved direction is one in which there is more than one block.

Grid block permeabilities in each direction must be entered for all grid blocks.

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 and fracture-matrix flow when \*SHAPE \*K-HARMONIC is used.

\*MATRIX permeabilities should be the values measured from a piece of un-fractured matrix material (intrinsic). On the other hand, \*FRACTURE permeabilities should be entered as effective fracture permeabilities; that is, the permeability of the fracture system with respect to a ratio of fracture and element areas in a plane perpendicular to the flow. 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 fracture porosity.

```
*PERMJ and *PERMK may be specified using the *EQUALSI array input
option. *PERMI must first be input. For example, for a dual
porosity/dual permeability system:
  ** Specify horizontal permeability
  *PERMI *FRACTURE *ALL
  2500. 2200. 2150. 2300. 2200.
  ...
  ...
```

```

*PERMI *MATRIX *ALL
 340.   315.   280.   260.   240.
...
** J direction permeabilities are equal to I direction values.
*PERMJ *MATRIX *EQUALSI
*PERMJ *FRACTURE *EQUALSI
** Vertical permeability is one tenth of the horizontal permeability.
*PERMK *MATRIX *EQUALSI * 0.10
*PERMK *FRACTURE *EQUALSI * 0.10

```

The example demonstrates a scenario in which you specify only the values for the I direction and then, if necessary, alter the values in the other directions or equate the I direction values to the remaining directions.

### **Matrix/Fracture and \*EQUALSI Operators**

Keywords \*PERMJ and \*PERMK are able to use the \*EQUALSI facility for entering grid array data, even for the \*MATRIX and \*FRACTURE portions of the array. However, use of \*EQUALSI with \*MATRIX and \*FRACTURE has this additional restriction: the \*MATRIX and \*FRACTURE instances of the keyword must use the same numerical operator if an operator is used. For example, the following data fragment will not work as expected:

```

*PERMI *MATRIX *CON 100
*PERMI *FRACTURE *CON 200
*PERMJ *FRACTURE *EQUALSI + 111 ** Operator "+"
*PERMJ *MATRIX *EQUALSI * 2      ** Operator "*" (not allowed)

```

The acceptable range of values for permeability is:

	<b>SI m</b>	<b>Field md</b>	<b>Lab md</b>
min	0.0	0.0	0.0
max	1.0E+13	1.0E+13	1.0E+13

---

## Bulk Volume Modifiers (Optional)

\*VOLMOD

### PURPOSE:

\*VOLMOD indicates input of an array of bulk volume modifiers.

### ARRAY:

\*VOLMOD

### DEFAULTS:

Optional keyword. Default: 1.0

### CONDITIONS:

Values for bulk volume modifiers must be non-negative, and they may exceed 1.

### EXPLANATION:

Keyword \*VOLMOD applies a multiplier to a block's bulk volume, that is, the rock as well as the pore volume. This allows for correct accounting of heat in the rock in proportion to the pore space. The CMG isothermal simulators may refer to \*VOLMOD as a *pore* volume multiplier since they are not concerned with rock volume. However, such \*VOLMOD data may be transferred directly to STARS running in either isothermal or thermal mode.

The \*VAMOD facility is recommended for STARS instead of \*VOLMOD for repeated patterns like one-eighth nine spot, since the additional area factors can increase significantly the accuracy of the pattern representation. \*VOLMOD may be more convenient to use when per-block volume modifier data comes from other software such as mapping packages. In any case it is recommended that volume modifiers be used instead of porosity adjustments, to avoid unusual or unphysical porosity values. This is especially true when some quantity depends on porosity, e.g., variable permeability and chemical reactions.

\*VOLMOD modifiers accumulate as data lines are read, so avoid repeated specifications for a block. For example, if one data line specifies a multiplier of  $A$  for a block and then another data line specifies a multiplier of  $B$  for that same block, the resulting multiplier for that block will be  $A \cdot B$ . This often happens when values are applied to overlapping rectangular areas.

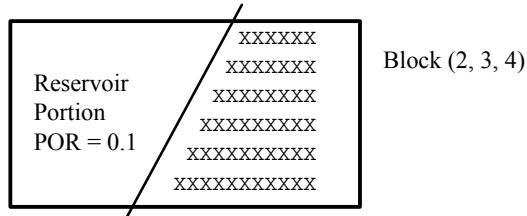
Note that multipliers larger than 1 may be used to associate volume external to the reservoir with a grid block. For instance, total well bore volume for a gas well can be associated with the grid block through which the well penetrates by using a multiplier for that block. This practice is not recommended for thermal simulation since the increased block heat capacity may result in unrealistically low temperatures near the well. See **Pseudo-Infinite Blocks**, below.

### Natural Fracture Grids

For natural fracture grids, bulk volume modifiers can be applied to the matrix and fracture cells 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. If \*VOLMOD must be used instead of \*VAMOD, it is recommended that \*VOLMOD \*MATRIX and \*VOLMOD \*FRACTURE be used together, or not at all, for dual porosity models, and that each pair of co-located matrix and fracture blocks be given the same multiplier.

**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.

**Pseudo-Infinite Blocks**

The use of very large block volume modifier to model a constant-pressure boundary may have subtle negative side effects. See **Pseudo-Infinite Blocks** in the manual entry for **\*CONVERGE**.

## **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) | permi is replaced by | permi | *ntg |
| (c) | permj is replaced by | permj | *ntg |

where "por" denotes the grid block's porosity as set using the \*POR keyword, "permi" and "permj" are the block's permeabilities as set using the \*PERMI and \*PERMJ keywords, and "ntg" is the net- to-gross ratio as derived from the net pay value assigned by 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.

A net-to-gross value of zero has cell-type consequences. See section **Zero-Porosity Blocks** at the beginning of this chapter.

The acceptable range of values for any derived net-to-gross ratio is:

	<b>SI</b>	<b>Field</b>	<b>Lab</b>
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.

### **ARRAY:**

\*NETGROSS

### **DEFAULTS:**

Optional keyword. The default net-to-gross multiplier is 1.0.

In dual porosity models, net-to-gross multipliers can be applied to the matrix and fracture pore volumes separately by use of the \*MATRIX and \*FRACTURE qualifiers. However, the default for fracture multipliers is still 1.0, even if \*NETGROSS \*MATRIX (or just \*NETGROSS) appeared, and the default for matrix multipliers is still 1.0, even if \*NETGROSS \*FRACTURE appeared.

It is recommended that \*NETGROSS \*MATRIX and \*NETGROSS \*FRACTURE be used together, or not at all, for dual porosity models.

### **CONDITIONS:**

This keyword should not be used with \*NETPAY.

### **EXPLANATION:**

The keyword \*NETGROSS allows input of net-to-gross which are used to modify the porosities and permeabilities in the I and J directions. The net-to-gross ratios are used as multiplication modifiers as follows:

- |     |                      |       |      |
|-----|----------------------|-------|------|
| (a) | por is replaced by   | por   | *ntg |
| (b) | permi is replaced by | permi | *ntg |
| (c) | permj is replaced by | permj | *ntg |

where "por" denotes the grid block's porosity as set using the \*POR keyword, "permi" and "permj" are the block's permeabilities as set using the \*PERMI and \*PERMJ keywords, and "ntg" is the incoming net-to-gross ratio for the block. 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.

A net-to-gross value of zero has cell-type consequences. See section **Zero-Porosity Blocks** at the beginning of this chapter.

The acceptable range of values is:

	<b>SI</b>	<b>Field</b>	<b>Lab</b>
min	0.0	0.0	0.0
max	1.0E+4	1.0E+4	1.0E+4

---

## **Transmissibility Multipliers (Optional)** \*TRANSI, \*TRANSJ, \*TRANSK

### **PURPOSE:**

Specify transmissibility multipliers in the various positive directions.

### **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 default 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.

For an isothermal run (see \*ISOTHERMAL in the Numerical Methods data section), specifying a multiplier of 0.0 before the \*END-GRID keyword will remove the interblock connection completely. This matches the behavior of the other isothermal CMG simulators. To start a connection with a value of 0.0, specify the 0.0 value in the first segment of recurrent data instead of the Reservoir Description data.

For a thermal run, specifying a multiplier of 0.0 before the \*END-GRID keyword will not remove the interblock connection. Such a connection will have heat conduction between the grid blocks but there will be no fluid flow, at least until the multiplier is changed optionally to non-zero in recurrent data.

### **EXPLANATION:**

A transmissibility multiplier is a factor that is applied to both convective and dispersive flow. Therefore, it affects fluid phase flow involving relative permeabilities and viscosities, convective heat flow and flow of components caused by dispersion. This factor is deemed to apply to the pore space only, and is not applied to conductive heat flow.

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 this transmissibility is used, it is multiplied by the multiplier specified via keywords \*TRANSI, \*TRANSJ and \*TRANSK.

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 any interblock connection involves two grid blocks, a method is required for indicating to which connection the multiplier is assigned when using the grid-block-based \*TRANSI, etc.

#### **Inter-block flow between blocks on a single grid:**

This rule applies whether the grid is the (main) fundamental grid (grid 1), or any refined grid. If flow between a pair of blocks is considered, and they both lie on the same grid, then it is the block with the lowest I index for an I direction pair, or the lowest J index for a J direction pair, or the lowest K index for a K direction pair, that supplies the multiplier; that is, a directional multiplier applies to a block's interface with its neighbour with the higher index in that direction. These rules apply even when faults are present (see \*FAULT following). Note that fault considerations only affect lateral (I and J direction) calculations.

This rule is altered for \*GRID \*RADIAL and \*GRID \*HYBRID when connecting block nj to block 1 in (angular) direction J when nj exceeds 1; that is, when a subdivided ring is being closed. In this case, the multiplier from block nj is used for the closure. Also, flow perpendicular to the wellbore in \*HYBRID grids uses a multiplier averaged over the two directions perpendicular to the well.

Note that refined grids inherit the multipliers from their parent block, unless special values are read for the refined grid directly (\*RG qualifier).

Note that except for the special case of zero transmissibility multipliers, all refined blocks in a locally refined block inherit the multipliers from their parent block. Interior blocks and those at a refined grid – refined grid interface would inherit the parent's multipliers.

In the special case of a zero multiplier, the multiplier is inherited only by those refined blocks on the appropriate interface.

If the user needs to modify transmissibilities of an interface to a value other than zero when refined grids are involved, the use of the \*RG keyword is required to explicitly refer to refined blocks at the interface.

#### **Flow between a refined grid and the fundamental:**

Basically the same rules apply as for fundamental blocks, except when determining the I, J, or K index of a refined block at a refined block fundamental block interface, refer to its parent's I, J, or K index. If the refine blocks parent has the lowest I (J, or K) index then the multiplier 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 dual porosity models and \*SHAPE \*GK, except for \*DUALPERM (when such multipliers exist). The \*SHAPE \*K-HARMONIC does not use any \*MATRIX transmissibility multiplier for the fracture-matrix flow. \*MATRIX multipliers are applied to matrix-to-matrix flows within a block for \*SUBDOMAIN and \*MINC. Use \*TRANSMF to modify matrix-fracture flows.

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 non-zero. 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 (1,1,1) contains a 3 x 2 x 1 refined grid. Then I-direction transmissibility multipliers can be applied to flows going in or out of the refined grid as follows:

```
*TRANSI *RG 1 1 1 *ALL
.8 1 .8 .8 1 .8
```

Transmissibility multipliers may be applied to flow along the wellbore of a discretized well. Use the keyword corresponding to the local wellbore axial direction (which for a deviated wellbore may change along the well's length). For example, use the following to reduce by half the flow in a horizontal well completed in the I-direction in block (1,1,1).

```
*TRANSI *RG 1 1 1 CON 0.5
```

For \*TRANSI, etc., that appear before \*END-GRID use \*RG to refer to the wellbore instead of the surrounding parent block; in recurrent data you can use array qualifiers \*WELLBORE, \*TUBING, \*ANNULUS and \*RG.

The acceptable range of values for transmissibility multipliers is:

	<b>SI</b>	<b>Field</b>	<b>Lab</b>
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.

### **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 this transmissibility is used, a 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 acceptable range of values for transmissibility multipliers is:

	<b>SI</b>	<b>Field</b>	<b>Lab</b>
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.

---

## **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**.)

### **ARRAY:**

**\*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 ni \* nj \* nk values. The array values should consist of 0's to indicate blocks that are pinched out and 1's to indicate blocks that are not pinched out. (This keyword is similar to the **\*NULL** keyword in that 1's are used to indicate active blocks and 0's are used to indicate special blocks that do not fully participate in the simulation.)

Note that if a vertical stack of one or more corner point cells are pinched out, and they are surrounded above and below by active corner point cells, then the blocks above and below will not connect to each other UNLESS the pinched out cells form an 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 near 0.0 or matching, or nearly matching, top and bottom corner points) since the transmissibility calculations between cells above and below the pinched out stack do not use the intervening stack's thickness.

A pinched out status set with \*PINCHOUTARRAY over-rides an inactive setting using \*NULL. This means that if a block has been designated in a \*PINCHOUTARRAY list, fluid will pass through it regardless of it having been also designated inactive in a \*NULL list. However, a \*NULL setting overrides zero thickness; that is, a zero thickness block will not allow vertical fluid passage if it is flagged inactive using the \*NULL keyword. Note that zero thickness overrides zero pore volume; that is, a block that is pinched out due to having zero thickness will allow fluid passage regardless of whether it was assigned non-zero porosity values or not.

The simulator uses the following hierarchy when determining whether a block is pinched out (allows vertical fluid passage) or is completely inactive. Note that Rule (1) overrides Rule (2), which overrides Rule (3), which overrides Rule (4).

1. The block has been flagged as pinched out using the \*PINCHOUTARRAY keyword (a 0 value was assigned). This block will always be pinched out and this state will not be overridden.
2. The block has been flagged as inactive using the \*NULL keyword (a 0 value was assigned). This block will not allow vertical fluid passage, unless overridden by Rule (1).
3. A zero thickness block will be pinched out (allows vertical fluid passage) if not overridden by Rules (1-2).
4. A zero pore volume block will be inactive and not allow any fluid passage unless overridden by Rules (1-3).

Example:

To pinch out the second layer of a 100 X 100 X 9 model use the following:

```
*PINCHOUTARRAY *IJK 1:100 1:100 2:2 0
```

Note that the remaining blocks in the model need not be referred to and their state remains unaltered.

## Pinchout Tolerance (Optional)

\*PINCHOUT-TOL

### PURPOSE:

\*PINCHOUT-TOL controls the minimal thickness required to initiate an automatic pinched out connection (see also \*PINCHOUTARRAY).

### FORMAT:

\*PINCHOUT-TOL pnctol

### DEFINITIONS:

pnctol

Minimal thickness required under which a block is removed from the simulation and the block above it is connected directly to the block below. Dimensions are (m | ft).

### DEFAULTS:

Optional keyword. The defaults are:

- 0.0010 (m | ft) for Corner Point grids on the fundamental grid only, if \*PINCHOUTARRAY does not appear;
- 0.0002 (m | ft) for non-Corner Point grids on the fundamental grid only, if \*PINCHOUTARRAY does not appear;
- (m | ft) otherwise.

Values at computer round-off levels are considered to be identically 0.0.

### CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group.

\*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:**

Blocks whose thickness are less than pnctol are considered to be pinched out. When this occurs, blocks above and below the pinched out blocks are connected as if the pinched out blocks did not exist. The minimum thickness “pnctol” at which this occurs is controlled using \*PINCHOUT-TOL.

Pinched out blocks are used to remove layers from the simulator's calculations in certain regions on a grid. Such layers may be required to model geological strata that exist in other portions of the grid but are not present in the pinched out region. The \*PINCHOUT-TOL and \*PINCHOUTARRAY keywords correspond to true geological pinch outs.

Blocks that are designated as pinched out allow fluid to pass through them vertically, but not laterally, and fluid can pass through a stack of one or more pinched out blocks on a grid. Pinched out blocks can also lie between active blocks on a grid and a refined grid region, allowing fluid to pass through between the grids. Two \*HYBRID grids can even be connected vertically through intervening pinched out layers on their parent grid.

\*NULL settings override \*PINCHOUT-TOL-generated pinch outs; that is, a small thickness block will not allow vertical fluid passage if it is flagged inactive using the \*NULL keyword. Note that small thickness overrides zero pore volume; that is, a block that is pinched out due to having small thickness will allow fluid passage regardless of whether it was assigned non-zero porosity values or not.

---

## 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).

In the simulator, throws provide modifications to depth data given earlier through use of the \*DEPTH, \*DTOP or \*PAYDEPTH 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, i1, i2, j1, and j2 locate grid block columns whose first index (I index) lies between i1 and i2 inclusive, whose second index (J index) lies between j1 and j2 inclusive, and whose third index (K index) lies between 1 and nk inclusive.

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:

\*FAULT should not be used with \*GRID \*CORNER. (Fault data can be entered directly for corner point grids.)

### 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 and \*PAYDEPTH keywords) will apply. If a full and correct depth array was introduced earlier (using the \*PAYDEPTH option, for instance), the throw values can be set to 0.0. (If a grid block is assigned to more than one fault block, throws 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 are:

	<b>SI</b> <b>m</b>	<b>Field</b> <b>ft</b>	<b>Lab</b> <b>cm</b>
min	1.0E-3	.00328	0.1
max	1.0E+3	3,280.0	1.0E+5

---

## 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, njhigh where i refers to the i direction and j refers to the j direction. Low refers to flow between block i (or j) and i-1 (or j-1). High refers to flow between block i (or j) and block i+1 (j+1).

The \*FAULTARRAY binary integer flag uses the following convention:

nilow, nihigh, njlow, njhigh = 0 if the connection is a standard connection

nilow, nihigh, njlow, njhigh = 1 if the connection is a fault connection

The value of \*FAULTARRAY for a block is:

$$\text{IVAL} = \text{nilow} + 2*\text{nihigh} + 4*\text{njlw} + 8*\text{njhig}$$

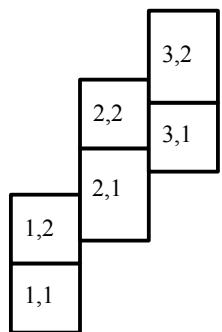
Thus if all connections are standard IVAL = 0, and if all connections take into account block depths (are fault connections), IVAL = 15.

Use:

```
**all connections are really are fault connections  
*FAULTARRAY *CON 15  
**all i connections are fault connections,  
**all J connections are standard  
*FAULTARRAY *CON 3
```

Example:

<b>Standard Connections:</b> <b>i Connections</b>	<b>Fault Connections:</b> <b>j Connections</b>
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 →

---

## Special Connections (Optional)

\*SCONNECT

### PURPOSE:

Specify flow connection between cells that are not connected otherwise.

### FORMAT:

\*SCONNECT { *uba1 uba2 trans* }

### DEFINITIONS:

\*SCONNECT { ... }

\*SCONNECT may be followed by one or more sets of data, with one set per line.

*uba1 uba2*

User Block Addresses of two cells between which the special flow connection is created. Each cell must be active, that is, not null, not pinched out and not the parent cell of a locally refined grid. Index ranges are not allowed. UBA qualifiers MT and FR are allowed for naturally fractured grids.

*trans*

Numerical value of the fluid flow transmissibility (md-m | md-ft | md-cm). The value is expected to be the result of taking a cross-sectional area for the flow, multiplying it by some absolute permeability, and dividing it by a centre-to-centre inter-block distance.

### DEFAULTS:

Optional keyword. No defaults.

### EXPLANATION:

#### Special Connection

A regular connection is one that is generated automatically between adjacent active blocks. If *uba1* and *uba2* specify active cells for which a regular connection is NOT made, a special connection is created and added to the existing connection list. A special connection is used like a regular one in a flow equation, with the following exceptions:

1. For a thermal run there is no conductive heat flow (keywords \*THCONR, etc.).
2. Fluid-flow transmissibility is not recalculated if permeability varies (keywords \*PERMCK, etc.).
3. There is no electrical current flow in an electrical heating run (keywords \*ELECHEAT, etc.)
4. There is no mechanical dispersion (keywords \*MDSPI\_WAT, etc.) since velocity is not known.
5. There is no contribution to capillary number dependence (keywords \*DTRAPW, etc.) since velocity is not known.

These exceptions exist because the fluid transmissibility supplied by \*SCONNECT is insufficient information for the treatment of the above features. A regular connection has separate geometrical factors for its two cells, which are used to calculate flux and velocity for flow in series.

Note that these exceptions apply only to a special connection and not to any regular connection in the same run. For each of these restrictions a message is issued when a run contains at least one special connection that suffers from the restriction.

### Overwriting Transmissibility

If *uba1* and *uba2* specify an existing regular connection, that connection's internally generated fluid transmissibility is replaced by *trans*. This can be useful for setting transmissibilities to specific values in a way that avoids printing out lists of inter-block connections and using multipliers to adjust transmissibility values.

#### Examples:

Connect cells on opposite sides of a 2-dimensional grid:

```
...
*GRID *CARTESIAN 10 1 5
...
*SCONNECT  1 1 1  10 1 3  200.  ** md-ft
            1 1 2  10 1 4  230.
            1 1 3  10 1 5  310.
...
...
```

## Fault Transmissibilities (Optional)

\*TRANSF

### PURPOSE:

Adjusts transmissibilities on a fault basis.

### FORMAT:

*TRANSF	'Fault_Name'	fault_trans_mult
<pair or single>	<pair or single>	...

### DEFINITIONS:

\*TRANSF

Keyword introducing the fault name, multiplier and fault description.

'Fault\_Name'

A quoted name for this fault.

fault\_trans\_mult

Transmissibility multiplier for the connections across this fault. Multipliers of this type apply cumulatively to previously applied multipliers.

<pair or single>

Identifiers for the connections that are to make up this fault. The identifier can either be of the “pair” or “single” variety. A “pair” identifier looks like:

i1 j1 k1 [\*IDIR or \*JDIR or \*KDIR] i2 j2 k2

which refers to an existing connection between cells, while a “single” identifier looks like:

i1 j1 k1 [\*IDIR- or \*IDIR+ or \*JDIR- or \*JDIR+ or \*KDIR- or \*KDIR+]

which refers to all connections on a certain cell face. Multiple mixed “pair” or “single” identifiers can follow a \*TRANSF line.

### DEFAULTS:

Optional keyword.

### CONDITIONS:

This keyword, if present, must be in the RESERVOIR DESCRIPTION keyword group. “Pair”-type identifiers should refer to pairs of cells that actually are connected in the grid direction specified by the given direction identifier (\*IDIR, \*JDIR or \*KDIR). Refined grid cells cannot be referenced in the description of a “pair” or “single” identifier.

## **EXPLANATION:**

This keyword allows the imposition of a single multiplier to a group of connections. The group of connections can be thought of as corresponding to a single fault, and the multiplier as a way of adjusting the sealing properties of this fault. If the descriptor is of “pair” type, the multiplier will be applied to an (existing) connection. If the descriptor is of “single” type, the multiplier will be applied to all connections that the cell has to other cells on a specified face. The face for the latter is identified using \*IDIR-/+ or \*JDIR-/+ or \*KDIR-/+ descriptors. The “-” identifier refers to the face crossed by moving from the cell to its neighbour along the grid direction corresponding to decreasing the appropriate I, J or K index, and the “+” identifier to increasing the appropriate I, J or K index. The multiplier can be 0 if desired, which will eliminate connections.

For non-isothermal simulations, this multiplier applies only to fluid flow, not to heat flow.

## **Duplicate Assignments to a Cell Face**

If multiple instances of \*TRANSF attempt to assign a factor to the same cell face more than once, only one instance of the assignment will be accepted, that is, the factor is applied only once. However, no check for duplicate assignments is done while processing a single instance of \*TRANSF with a list of cell faces. Therefore, data entry must be done carefully. Consider the following examples in which a multiplier of  $10^5$  is assigned to the face between blocks (5,1,1) and (6,1,1).

```
** The following data will assign the factor only once
** since duplicate assignment is detected
*TRANSF 'Fault1' 1e5
    5 1 1 *IDIR 6 1 1
*TRANSF 'Fault1' 1e5
    5 1 1 *IDIR+
*TRANSF 'Fault1' 1e5
    6 1 1 *IDIR-

** The following data will assign the factor twice
*TRANSF 'Fault1' 1e5
    5 1 1 *IDIR+
    6 1 1 *IDIR-
```

---

## Aquifer Model

\*AQUIFER, \*AQMETHOD, \*AQPROP, \*AQVISC, \*AQCOMP,  
\*AQLEAK, \*HFPROP, \*AQGEOM

### PURPOSE:

Define aquifer model for reservoir boundary water influx and heat conduction calculations.

### FORMAT:

```
*AQUIFER      (*BOTTOM | *BOUNDARY |
                  { *REGION i1(:i2) j1(:j2) k1(:k2) (direction) } )
*AQMETHOD    (*CARTER-TRACY | *FETKOVITCH
                  | *SEMI-ANALYTICAL)
*AQPROP       Thickness Porosity Permeability Radius Angle (R-Ratio)
*AQVISC        Aqvisc
*AQCOMP        Aqcomp
*AQLEAK        (*ON | *OFF )
*HFPROP        ( aqrcap aqrrend )
*AQGEOM        (*RECTANG | *RADIAL )(*INFINITE | *FINITE )
```

### DEFINITIONS:

#### \*AQUIFER

Specifies the aquifer location, via one of three methods:

Use \*BOTTOM to connect aquifer to the bottom of the reservoir.

Use \*BOUNDARY to connect aquifer to all boundary blocks in the sides of the reservoir.

Use \*REGION to connect aquifer to an arbitrary list of fundamental grid blocks via I-J-K address ranges i1(:i2) j1(:j2) k1(:k2). The \*REGION keyword and the data following it may appear multiple times after \*AQUIFER if necessary to describe a complex geometry. Use optional *direction* (\*IDIR, \*JDIR or \*KDIR) for connection to the block face on the exterior reservoir boundary in the indicated direction. Interior block faces are ignored. For example, for \*IDIR the connection is to the -I face when I = 1 and to the +I face when I = NI.

#### \*AQMETHOD

Specifies the method used to calculate water influx from the aquifer. The choices available are \*CARTER-TRACY, \*FETKOVITCH and \*SEMI-ANALYTICAL. See **Water Influx Models** in EXPLANATION below for discussions of these methods.

#### \*AQVISC aqvisc

Aquifer water viscosity (cp). Use this keyword only to over-ride the default.

#### \*AQCOMP aqcomp

Total aquifer compressibility (1/kPa | 1/psi). Use this keyword only to over-ride the default.

## \*AQPROP

Specifies the following aquifer properties:

Thickness      Vertical dimension (m | ft | cm) of the aquifer when connected to the bottom of the reservoir. For aquifers connected to the sides of the reservoir, it defines the vertical extent for Carter-Tracy and Fetkovitch method, but lateral extent for the semi-analytical method. Enter 0 when using the \*INFINITE option of \*AQGEOM.

Porosity      Aquifer porosity.

Permeability      Aquifer permeability (md | md | md).

Radius      Effective reservoir radius (m | ft | cm).

Angle      Angle of influence (expressed as a fraction of a circle).

R-Ratio      Optional ratio of the aquifer's external radius to the effective reservoir radius. Used only for \*FETKOVITCH.

Radius and Angle are used only used by \*CARTER-TRACY and \*FETKOVITCH, so enter zero when using the semi-analytical method.

## \*AQUEAK ( \*ON | \*OFF )

Specifies whether water is allowed to leak from the reservoir into the aquifer where the block pressure exceeds the adjacent aquifer pressure. Aquifer behavior is modeled more accurately with \*ON, that is, leakage is allowed. For \*OFF no leakage is allowed.

## \*HFPROP ( aqrcap aqrcnd )

Specifies that heat conduction calculations are performed between aquifer and reservoir. For a thermal run (keyword \*ISOTHERMAL absent), heat transferred with convective flow is always accounted for.

Optionally, the following aquifer thermal properties may be specified in order to over-ride the default.

aqrcap      Volumetric heat capacity of rock in the aquifer (J/m<sup>3</sup>-C | Btu/ft<sup>3</sup>-F). A non-positive value triggers the default.

aqrcnd      Thermal conductivity of rock in the aquifer (J/m-day-C | Btu/ft-day-F). A non-positive value triggers the default.

## \*AQGEOM ( \*RECTANG | \*RADIAL ) ( \*INFINITE | \*FINITE )

Specify the aquifer geometry used in calculating heat conduction to and from the aquifer. For the semi-analytical method, this geometry is used also to calculate water influx. See **Geometry Options** below. Either \*RECTANG or \*RADIAL must appear after \*AQGEOM.

## DEFUALTS:

Absent	Action
*AQUIFER	No aquifer calculations.
*AQMETHOD	*CARTER-TRACY using default from *AQFUNC.
*AQPROP	<p>Thickness: Average reservoir thickness for aquifer attached to reservoir sides for *CARTER-TRACY and *FETKOVITCH, or *INFINITE for *SEMI-ANALYTICAL; square root of the contact area for aquifer connected to reservoir bottom.</p> <p>Porosity = reservoir average porosity.</p> <p>Permeability = reservoir average permeability, in aquifer flow direction.</p> <p>Radius: radius of 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 <math>\pi</math>, for *BOTTOM; or the square root of the contact area divided by <math>\pi</math>, for *REGION.</p> <p>Angle: Full circle for all grids except radial, when the sum of the outer ring angular extents is used (after division by <math>360^\circ</math>), for *BOUNDARY and *REGION; or angular extent of the bottom of the reservoir (after division by <math>360^\circ</math>), for *BOTTOM.</p> <p>Note: A zero value entered for any individual item after *AQPROP will be replaced internally with its default.</p>
R-Ratio	R-Ratio = 100
*AQVISC	aqvisc = viscosity of component specified by *AQFRCOMP (usually water) at the average initial temperature of adjacent blocks.
*AQCOMP	aqcomp = cmpf + cmpr, where cmpf is the liquid compressibility of the component specified by *AQFRCOMP (usually water), and cmpr is the formation compressibility of rock type #1 (see *CPOR).
*AQLEAK	*AQLEAK *OFF.
*HFPROP	no conductive heat flow. For a thermal run (keyword *ISOTHERMAL absent), heat transferred with convective flow is always accounted for.
aqrcap	aqrcap = value from adjacent reservoir rock.
aqrcnd	aqrcnd = value from adjacent reservoir rock.
*AQGEOM	*RECTANG *INFINITE for heat conduction calculation (if *HFPROP is present) and water influx for the semi-analytical method. If *AQGEOM and one of *RECTANG   *RADIAL is present, but both *INFINITE and *FINITE are absent, then *INFINITE is assumed.

In the above, ‘average’ refers to a pore volume weighted average taken over aquifer connecting 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.

## **CONDITIONS:**

These keywords must be in the Reservoir Description keyword group.

The composition of "water" in the aquifer is specified via keyword \*AQFRCOMP in the COMPONENT PROPERTIES chapter.

The minimum required keyword to enable water influx is \*AQUIFER followed by either \*BOTTOM, \*BOUNDARY, or a \*REGION definition. To enable heat conduction calculations, the minimum additional required keyword is \*HFPROP.

Multiple aquifers maybe specified, that is, keyword \*AQUIFER may appear more than once. Defaults are applied separately for each \*AQUIFER definition.

## **EXPLANATION:**

The aquifer models described here allow water and heat influx (and outflow for \*AQLEAK \*ON) to a reservoir from one or more aquifers. Use of these aquifer models can be more economical for simulation purposes than using many grid blocks filled with water. However, if great accuracy is required in modeling aquifers then water filled blocks should be used.

### **Water Influx Models**

The \*CARTER-TRACY water influx calculation option is a Carter-Tracy approximation. For more information, refer to R. D. Carter and G. W. Tracy, "An Improved Method for Calculating Water Influx", Trans., AIME, Vol. 219, (1960), 415-417. This method uses a dimensionless pressure influence function P(td), expressed as a function of dimensionless time td. The function is defined using a table (see keyword \*AQFUNC), 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. See Appendix D.19 for further discussion of the Carter-Tracy and Fetkovitch options.

The \*FETKOVITCH water influx calculation option is based on work by Fetkovitch (see M. J. Fetkovitch, "A Simplified Approach to Water Influx Calculations - Finite Aquifer Systems", JPT, July 1971, 814-828). This approach is able to model finite aquifers via parameter R-Ratio and does not need dimensionless pressure function \*AQFUNC.

The \*SEMI-ANALYTICAL water influx calculation option is based on an extension of the work by Vinsome and Westerveld. See "A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators", Vinsome, P.K.W. & Westerveld, J.D., JCPT, July-September 1980, Volume 19, No. 3). With this method, the water influx from an adjacent aquifer region is predicted using a semi-analytical pressure profile based on a one-dimensional single-phase flow assumption. See Appendix D.12 for further discussion.

### **Heat Conduction Model**

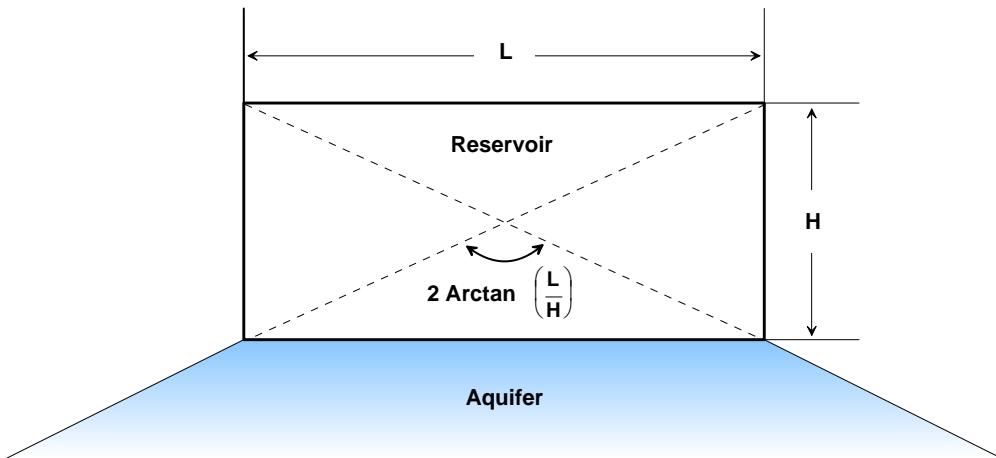
To model heat conduction to and from an aquifer, the keyword \*HFPROP must be specified. The method for heat conduction calculation is a semi-analytical formulation similar to the \*SEMI-ANALYTICAL water influx method. Heat conduction can be calculated for any of the water influx methods. For a thermal run (keyword \*ISOTHERMAL absent), heat transferred with convective flow is always accounted for, independent of conduction.

## Geometry Options

Flow of water and heat in the aquifer may be either linear or radial. Linear flow is appropriate for situations such as bottom water where the areal confinement of the aquifer is similar to that of the reservoir. Radial flow is useful for single-well problems where there is a surrounding aquifer in the horizontal direction. Specify which geometry to use via \*AQGEOM. The aquifer geometry type for heat conduction is the same as the one for water influx if \*SEMI-ANALYTICAL is used, and will take the one assigned by \*AQGEOM or by default for \*CARTER-TRACY and \*FETKOVITCH.

\*AQPROP specifies the aquifer properties for calculations of water influx from the aquifer. For a boundary aquifer (\*BOUNDARY) with \*CARTER-TRACY and \*FETKOVITCH, \*AQPROP defaults (Thickness, Angle and Radius) 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 from the reservoir defined.

For a bottom aquifer (\*BOTTOM) with \*CARTER-TRACY and \*FETKOVITCH, \*AQPROP defaults envision a square contact area with a 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 model assumes the aquifer is 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 aquifer angle (Angle) is taken from the wedge angle and equals to  $2\text{Arctan}(L/H)/360$ , aquifer thickness (Thickness) taken as L, and the effective reservoir radius (Radius) taken as the square root of  $(L^2H/\pi)$ .



For a region aquifer (\*REGION) with \*CARTER-TRACY and \*FETKOVITCH, \*AQPROP defaults are much like those for \*BOUNDARY except that the effective reservoir radius (Radius) is taken to be the square root of the contact area divided by  $\pi$ .

## **Limitations of Aquifer Models**

The aquifer models of this section are intended mainly for drawdown pressure maintenance. Therefore, care must be exercised when using these models to simulate outflow from the reservoir into the aquifer. Where flow reversal is significant in duration and extent, it is recommended that aquifers be modelled at least partially using water-filled grid blocks. This case may occur when the reservoir pressure is expected to increase substantially during the course of the simulation.

There are four restrictions applied to aquifer calculations when water is flowing into or out of an aquifer.

1. Initial pressures in the reservoir must be at or close to vertical equilibrium, since the initial pressure in each aquifer region is taken from the initial block values. This is enforced by disallowing a uniform pressure for a grid in which there is non-uniform block depth. The easiest way to satisfy this is to initialize the reservoir by performing gravity equilibrium calculations (\*VERTICAL).
2. To maintain a good mobility continuity, the endpoint of the water relative permeability  $K_{rw}$ (at  $S_w=1$ ) should be 1 in a block adjacent to an aquifer and should not decrease with temperature.
3. If flow potential is from reservoir to aquifer during the simulation and aquifer leakage is allowed (\*AQLEAK \*ON), the reservoir  $K_{rw}$  must be at least 1.0e-5 for the simulation to continue.
4. If flow potential is from reservoir to aquifer during the simulation and aquifer leakage is allowed (\*AQLEAK \*ON), the reservoir water must be below the steam temperature and must consist entirely of the aquifer component specified by keyword \*AQFRCOMP or its default.

## **Advanced Grid Options**

An aquifer may be attached to most types of grid blocks generated by the advanced grid options described earlier in this chapter. Of particular note are the following.

- a) Null blocks are skipped silently.
- b) Zero-porosity blocks will not render any water influx, but might gain or loss heat through conduction.
- c) For dual porosity blocks, the aquifer is attached to the fracture part of the block and not the matrix part.

## **Initial Temperature in Aquifer**

The initial temperature in the aquifer segment attached to a block is uniformly equal to the temperature in its associated block. It is not possible to specify initial aquifer temperatures or potentials that are different from the associated blocks. These aquifer models are designed to start at equilibrium conditions. Therefore, no matter what the initial temperature distribution in the reservoir, there is no initial conduction of heat (or flow of water) to or from any aquifer segments.

## Detailed Output

Use the subkeywords \*AQWATCUM, \*AQWATRATE, \*AQHEATCUM and \*AQHEATRATE of \*OUTPRN \*GRID, \*OUTSRF \*GRID and \*OUTSRF \*SPECIAL \*BLOCKVAR to see instantaneous rates and net accumulations for the aquifer regions attached to each grid block. Since aquifer regions normally are attached to select boundary blocks, the full grid output will be mostly zeros; also, the quantities are proportioned to the block face areas and so may not be useful to compare between blocks. The most useful output may be via the special history \*BLOCKVAR in which histories of selected quantities for selected blocks may be chosen.

Another useful printout is the \*AQSTAT option, which reports the net and rate quantities \*AQWATCUM, \*AQWATRATE, \*AQHEATCUM and \*AQHEATRATE in column format for the active aquifer regions only. See the \*OUTPRN manual page.

The total net flow of water and energy is reported along with the usual material balance statistics in the test output file.

These quantities are relative to the aquifer, so a positive value indicates a gain by the aquifer and therefore a loss by the reservoir grid block.

## EXAMPLES:

### Multiple Aquifers

To model water and heat influx into all boundary blocks in the sides and at the bottom of the reservoir, use the following:

```
*AQUIFER    *BOUNDARY
*AQMETHOD   *FETKOVITCH
*AQGEOM     *RECTANG *FINITE
*AQPROP     240.0  0.3  100.0  802.41  1.00  1.5
*HFPROP     35.0   24.0

*AQUIFER    *BOTTOM
*AQMETHOD   *FETKOVITCH
*AQGEOM     *RECTANG *FINITE
*AQPROP     102.04 0.25  250.0  0.0   0.0   0.0
*HFPROP     35.0   24.0
```

In this example, the numerical method employed for water influx calculations of both aquifers is the Fekovitch formulation and both aquifers retain a rectangular, finite geometry type for heat flux calculations. For the bottom aquifer zeroes indicate default values for the effective reservoir radius (Radius), angle of influence (Angle) and R-Ratio.

### Multiple \*REGION Aquifers

Multiple use of \*REGION following \*AQUIFER keyword is allowed to define complex aquifer connections,

```
*AQUIFER  *REGION 1:10  1      1  *JDIR
        *REGION 1      2:10  1  *IDIR
        *REGION 2:9    2:9  1  *KDIR
```

However, if multiple aquifer connections are intended for a block, such as the blocks at the corner of a reservoir, multiple aquifer definition should be used. For a block in a corner (e.g., i=1, j=1 and k=1), aquifers attach in all three directions, the correct input should be:

```
*AQUIFER *REGION 1 1 1 *IDIR  
. .  
*AQUIFER *REGION 1 1 1 *JDIR  
. .  
*AQUIFER *REGION 1 1 1 *KDIR
```

### Multiple Aqueous Components

These analytical aquifer models assume that the fluid in the aquifer pore space is solely water phase that consists entirely of one aqueous component. When there are multiple aqueous components you must choose which one is the aquifer component. See the manual entry for keyword \*AQFRCOMP in the COMPONENT PROPERTIES chapter.

### Converting Data from Versions Before 2002

Between versions 2001 (and before) and version 2002 the aquifer data changed location and syntax. The purpose of this move was to make available to STARS the industry-standard aquifer options used by the other CMG simulators. This data change is done automatically by the version 2002 data pre-processor.

The following instructions show how to do the change manually. These instructions assume that you wish to reproduce the aquifer behavior given by versions before 2002, and are not using newly available options. Of course, after the data is converted you may try the new options (e.g., remove \*AQMETHOD to try default method \*CARTER-TRACY).

**Note:** Even if data is converted simply to the new location and syntax, the aquifer behavior may not be the same as given by pre-2002 versions. This is because the semi-analytical calculations in pre-2002 versions had errors that have been corrected in version 2002.

1. Move all the aquifer data from its old location after \*END-GRID to a point before \*END-GRID (immediately before is best).
2. Keyword \*AQUIFER is unchanged, except that sub-keyword \*WELLBORE is not supported (use \*PHWELLBORE for wellbore heatloss).
3. Add keyword \*AQMETHOD \*SEMI-ANALYTICAL.
4. If keyword \*AQH is present, keyword \*FINITE must appear after \*AQGEOM. In this case, if \*AQGEOM is present then append \*FINITE after \*RECTANG or \*RADIAL; otherwise, add keyword \*AQGEOM \*RECTANG \*FINITE.
5. Add keyword \*AQPROP followed by five zeroes. For optional keywords \*AQH, \*AQPOR and \*AQPERM present, remove the keyword and move its value to the corresponding position after \*AQPROP.
6. If keyword \*ISOTHERMAL is absent, add \*HFPROP followed by two zeroes.
7. For optional keywords \*AQRCAP and \*AQRCND present, remove the keyword and move its value to the corresponding position after \*HFPROP.
8. Add \*AQLEAK \*ON.
9. Move keyword \*AQSTAT to the I/O Control data section.

---

## **Pressure Influence Function (Conditional)**

**\*AQFUNC**

### **PURPOSE:**

Define dimensionless pressure influence function for the Carter-Tracy water influx method.

### **TABLE:**

**\*AQFUNC { td P(td) }**

### **DEFINITIONS:**

**td**

Dimensionless time.

**P(td)**

Dimensionless pressure influence function.

### **DEFAULTS:**

See EXPLANATION for a discussion of the default table.

### **CONDITIONS:**

This keyword may be used only with the Carter-Tracy option of \*AQUIFER.

### **EXPLANATION:**

If water influx from (to) the aquifer is calculated using the Carter-Tracy approximation, a dimensionless pressure influence function  $P(td)$  as a function of dimensionless time  $td$  is required.

The default dimensionless pressure function is for a constant terminal-rate solution and an infinite radial aquifer, 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.

Influence functions for limited extent aquifers also can be found in this reference.

Additional tables may be found in Appendix D.19.

## Pore Volume Cut-Off Threshold (Optional)

\*PVCUTOFF

### PURPOSE:

\*PVCUTOFF controls the level at which a cell's pore volume is small enough to be considered zero.

### FORMAT:

\*PVCUTOFF *pvcut*

### DEFINITIONS:

*pvcut*

If a cell's raw pore volume is less than or equal to *pvcut*, its pore volume is considered to be zero. A zero pore volume has cell-type consequences. See section **Zero-Porosity Blocks** at the beginning of this chapter. The unit of *pvcut* normally is (m<sup>3</sup> | ft<sup>3</sup> | cm<sup>3</sup>); see **Advanced Unit Usage** below.

A cell's raw pore volume is its bulk volume multiplied by porosity (keyword \*PORI) multiplied by net-to-gross ratio (keyword \*NETPAY or \*NETGROSS).

### DEFAULTS:

If \*PVCUTOFF is absent, the pore-volume cut-off criterion is not used to change a cell's type. Note that other keywords (e.g., \*NULL, \*PINCHOUTARRAY) and criteria (e.g., \*PINCHOUT-TOL) may change a cell's type independent of \*PVCUTOFF.

### 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 fluid flow simulation. Such small pore volume blocks can hinder convergence.

#### Null versus Pinched Out

A cell that is changed to type Null by \*PVCUTOFF is not pinched out, that is, there is no vertical inter-cell connection through it. Use keyword \*PINCHOUTARRAY or \*PINCHOUT-TOL to force a vertical connection through a cell.

#### Advanced Unit Usage

If normal unit systems are used for \*INUNIT without exceptions, then the unit of *pvcut* corresponds to volume. However, the unit of *pvcut* is actually (*length*)<sup>3</sup> instead of *volume*, where *length* and *volume* are basic unit types defined in tables found in the EXPLANATION for \*INUNIT. For normal unit systems (*length*)<sup>3</sup> is effectively the same as *volume*. However, use of the \*EXCEPT option may result in a case where (*length*)<sup>3</sup> is not the same as *volume*. For example, for \*INUNIT \*FIELD *length* is ft and *volume* is ft<sup>3</sup>. If “\*EXCEPT 4 0” is added then *length* is m instead of ft, so that the unit of *pvcut* is m<sup>3</sup> even though *volume* is still ft<sup>3</sup>.

---

## Sectors (Optional)

\*SECTOR

### PURPOSE:

\*SECTOR controls the definitions of sectors, which are used to summarize regional reservoir activity.

### FORMAT:

*SECTOR 'Sector_Name'	i1:i2	j1:j2	k1:k2
	:	:	:

### DEFINITIONS:

'Sector\_Name'

Sector identification name (16 characters maximum), enclosed in quotes. The name 'Entire Field' is reserved for internal use (see DEFAULTS, below).

i1:i2

Indicates the I beginning and ending indices of the grid region where the sector is to be located.

j1:j2

Indicates the J beginning and ending indices of the grid region where the sector is to be located.

k1:k2

Indicates the K beginning and ending indices of the grid region where the sector is to be located.

### DEFAULTS:

The first sector is defined internally as all grid blocks in the field, and is called 'Entire Field'.

If \*SECTOR is absent, no additional sectors are defined.

### CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group.

Use keywords \*WPRN and \*WSRF to enable writing of sector statistics to the output file and SR2.

### EXPLANATION:

Sectors are collections of grid blocks. Various simulation results are available by sector in both the text output and the graphical output, making sectors useful for obtaining regional summaries. A grid block may belong to any number of sectors.

Example: Consider the following 7 x 6 x 1 grid with three sectors:

J = 6	S1	S1	S1	S1/S3	S3	S3	S3
J = 5	S1	S1	S1	S3	S3	S3	S2/S3
J = 4	S1	S1	S1			S2	S2
J = 3					S2	S2	S2
J = 2				S2	S2	S2	S2
J = 1			S2	S2	S2	S2	
	I = 1	2	3	4	5	6	7

The following input is needed to define these 3 sectors:

```
*SECTOR 'S1' 1:3 4:6 1
*SECTOR 'S1' 4 6 1
*SECTOR 'S2' 3:6 1 1
  'S2' 4:7 2 1
  'S2' 5:7 3 1
    6:7 4 1
    7 5 1
*SECTOR 'S3' 4:7 5:6 1
```

There is considerable flexibility in the way the \*SECTOR keyword can be used. For an alternative method see the \*SECTORARRAY keyword.

See keyword \*WPRN for the list of sector statistics.

---

## Sector Array (Optional)

\*SECTORARRAY

### PURPOSE:

\*SECTORARRAY defines Sectors (see \*SECTOR) using an input format. Sectors permit output to be printed on a regional basis.

### ARRAY:

\*SECTORARRAY 'Sector\_Name'

### DEFINITIONS:

'Sector\_Name'

Same as for \*SECTOR.

### DEFAULTS:

Comments for keyword \*SECTOR regarding default sectors apply here as well.

If SECTORARRAY is absent, then no additional sectors are defined.

### CONDITIONS:

This keyword must be in the RESERVOIR DESCRIPTION keyword group. All array qualifiers and array reading options are allowed for specifying the required ni \* nj \* nk values. The qualifiers and array values should come after 'Sector\_Name'. The array values should consist of either 0 (no sector membership for that cell) or 1 (indicating sector membership for that cell).

### EXPLANATION:

Sectors are collections of grid blocks that the simulator uses to summarize various quantities. A grid block can belong to different sectors. See the description of \*SECTOR above for further information about sectors.

The \*SECTORARRAY keyword provides an array-based alternative to the \*SECTOR range-based input format.

#### Example:

To set sector membership in a sector named 'Sector-1' for a few cells in the reservoir, use the following

```
*SECTORARRAY 'Sector-1' *IJK 1:5 1:5 1:1 0
              4   4   1   1
              2   3   1   1
```

where it is assumed the grid is dimensioned 5x5x1. Note that the string "1:5 1:5 1:1 0" is not actually required as "no membership" is the default state.

---

## Sector Names and Locations (Optional)

\*SECTORNAMES, \*ISECTOR

### PURPOSE:

Define sectors by a list of sector names and corresponding sector numbers.

\*ISECTOR assigns these sector numbers to cells using the standard array concepts.

### FORMAT:

\*SECTORNAMES 'Sector\_Name\_1' i1 'Sector\_Name\_2' i2 ...

### ARRAY:

\*ISECTOR

### DEFINITIONS:

'Sector\_Name\_1' i1 'Sector\_Name\_2' i2 ...

A series of sector names (16 characters maximum) and their associated numbers.

### DEFAULTS:

Comments for keyword \*SECTOR regarding default sectors apply here as well.

### CONDITIONS:

\*SECTORNAMES must appear before \*ISECTOR. All array qualifiers and array reading options are allowed for specifying the values for \*ISECTOR. Any value assigned by \*ISECTOR must appear in the \*SECTORNAMES list.

\*SECTORNAMES must appear at most once in the data, but \*ISECTOR may appear more than once.

### EXPLANATION:

See \*SECTOR for a discussion of sectors.

The \*ISECTOR keyword provides an array-based alternative to assigning sector numbers that have been defined using \*SECTORNAMES. Several instances of \*ISECTOR can appear in the data set.

Example:

To set sector membership in two sectors named 'LAYER-1' and 'LAYER-2', use the following:

```
*SECTORNAMES 'LAYER-1' 1 'LAYER-2' 2  
*ISECTOR *IJK 1:5 1:5 1:1 1  
          1:5 1:5 2:2 2
```

where it is assumed that the grid is dimensioned 5x5x2. Provided that this not a dual porosity problem, then three sectors will be defined in the simulation: 'LAYER-1', 'LAYER-2' and the default sectors.

# Other Reservoir Properties

## Summary of Other Reservoir Properties

This section contains data describing other reservoir properties. These data can be classified into the following groups:

1. Formation Compressibility
2. Reservoir Rock Thermal Properties
3. Overburden Heat Loss Options

### Critical Keyword Ordering

The critical keyword ordering is:

\*END-GRID  
Other keywords

It is recommended to follow the order in which keywords appear in this manual, when appropriate.

### Rock Properties

The rock property keywords \*ROCKTYPE and \*THTYPE are used to assign multiple rock types to the reservoir for the following groups of properties:

Formation compressibility

\*PRPOR, \*CPOR, \*CTPOR, \*CPTPOR, \*CPORPD, \*PORMAX,  
\*DILATION, \*EPCOMPACT, \*COMPACT\_VAR

Rock thermal properties

\*ROCKCP, \*THCONR, \*THCONS, \*THCONW, \*THCONO, \*THCONG,  
\*THCONTAB, \*THCOMMIX, \*ROCKCP\_SHL, \*THCONR\_SHL

Overburden heat loss

\*HLOSSPROP, \*HLOSST, \*HLOSSTDIF

Variable Permeability

\*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMEXP, \*PERMULI/J/K,  
\*PERMSLD

Keyword \*DILATION accesses a dilation-recompaction option that was developed for cyclic steam stimulation but is applicable for other processes and scenarios with appropriate parameter values. Keyword \*EPCOMPACT enables a compaction-rebounding model with elastic-plastic deformations. Unlike \*DILATION and \*EPCOMPACT model which use constant (or piecewise constant) pore volume compressibility and thermal expansion coefficient, the approach activated by keyword \*COMPACT\_VAR utilizes pressure history dependent property values. Currently, the three empirical reservoir deformation models (\*DILATION, \*COMPACT\_VAR and \*EPCOMPACT) are mutually exclusive for a simulation.

### Overburden Heat Loss Option

The heat loss directions and over/underburden thermal properties for the semi-analytical infinite-overburden heat loss model is specified by the following keywords:

\*HLOSSPROP \*HLOSST \*HLOSSTDIF

### Electrical Heating

The electrical heating option is enabled by keyword \*ELECHEAT, and static properties like electrical conductivity are specified.

### Natural Fracture Changes in v2007

The natural fracture options triggered by \*DUALPOR, etc., have been enhanced significantly and a number of important bugs have been fixed. Previously "effective" fracture porosity was entered and used (incorrectly) for calculation of quantities (e.g., heat capacity, thermal conductivity and reaction rates) that required "intrinsic" porosity. Starting with v2007 fracture and matrix porosities are treated consistently and pseudo values of these properties are not needed. Also, numerous improvements have been made to internal natural-fracture calculations for both fluid flow and heat conduction, and the User's Guide was corrected.

Existing data sets will need some modification of the properties organized under \*ROCKTYPE. Previously these properties were given pseudo values calculated from intrinsic (unfractured matrix) values using formulas found in the section "Fracture and Matrix Properties" of Appendix E.8. Now, these input parameters should have their intrinsic values. Consequently there is no longer a need for separate \*ROCKTYPE types for matrix and fracture if they have the same intrinsic formation properties.

Take as an example template "sttst28.dat" which has no formation/rock in the fracture blocks. First consider the previous data, before v2007. The matrix \*ROCKTYPE data was

```
*CPOR 3E-6
*ROCKCP 35
*THCONR 24
*THCONW 24
*THCONO 24
*THCONG 24
*HLOSSPROP OVERBUR 35 24 UNDERBUR 35 24
```

and the fracture \*ROCKTYPE data was

```
*ROCKCP 0
*THCONR 0
*THCONW 16
*THCONO 16
*THCONG 16
*HLOSSPROP OVERBUR 35 24 UNDERBUR 35 24
```

In the fracture data note the value of 0 for \*ROCKCP and \*THCONR, as well as the fluid thermal conductivity values that are 2/3 the matrix values. These were pseudo values calculated to obtain the desired end result from the previous treatment of fracture blocks.

Starting with v2007 the matrix (intrinsic) rock type data noted above is applied to both matrix and fracture blocks, so that only one \*ROCKTYPE is needed. Rock properties \*ROCKCP and \*THCONR are not used in these fracture blocks which contain no rock. The fracture fluid phase thermal conductivities now require intrinsic values 24 instead of the previous pseudo value of 16 (2/3 of 24).

For the matrix properties it appears that the values are unchanged but this is not strictly true. The previous matrix "effective" values usually were close to the intrinsic values so it was a common practice to enter the intrinsic values instead. Now, use of the intrinsic values is strictly correct.

New keywords \*FRFRAC and \*FORMINFRAC let you specify that a fracture cell contains some formation, in a way that is consistent and correct with regard to porosity treatment. Specification of pseudo properties from complex formula in Appendix E.8 is no longer needed, greatly reducing the task of data preparation for the rock-in-fracture modelling technique. For example, see templates "sttst29.dat" and "sttst31.dat".

New keyword \*SHAPE lets you control which type of shape factor is used in calculating matrix-fracture flow in natural fracture grid systems: \*GK (Gilman-Kazemi, the default) or \*K-HARMONIC. Previously the shape factor always used was \*GK instead of what the User's Guide indicated. See new templates "stgro041.dat", stgro042.dat" and "stgro043.dat".

Generally natural fracture results generated by v2007 are close to previous results after the required data conversion. Because of numerous bug fixes and improvements to low-level calculations in the natural fracture treatment, only the simplest cases (uniform, isotropic, isothermal) will obtain exactly the same results. Most of these improvements affect aspects of the result that usually are of only second order importance. However, the improved consistency shows more in cases where some specific aspect becomes of primary importance (e.g., mimic natural fracture grid with suitable "single-porosity" grid type).

Generated result differences will be more pronounced in rock-in-fracture cases, which can depend largely on the pseudo values used for "fracture" rock properties. The formulas for these pseudo values went through several revision stages, the last of which appeared only in interim releases after v2006.10. Before v2007 the basis of the natural fracture feature was not really intended for rock-in-fracture usage, so a number of small inconsistencies were present even if the rock property pseudo values were correct by the latest formula. Starting with v2007 the rock-in-fracture feature is treated consistently, so some result differences are expected. In addition, data entry is much simpler since only intrinsic rock and thermal properties are needed.

---

## **Indicate End of Grid Definition (Required)**

**\*END-GRID**

### **PURPOSE:**

\*END-GRID flags the beginning of the data that defines the other reservoir properties.

### **FORMAT:**

\*END-GRID

### **CONDITIONS:**

This keyword must occur after all the grid definition keywords in chapter Reservoir Description and before keywords in this chapter Other Reservoir Properties.

### **EXPLANATION:**

This keyword signals the Grid Module to stop reading and processing data, and passes control back to STARS.

This chapter consists largely of keywords that are unique to STARS, and so not found in the other CMG simulators and hence are not in the Grid Module.

---

## **Rock Type**

**\*ROCKTYPE, \*THTYPE**

### **PURPOSE:**

Define and assign multiple rock property types.

### **FORMAT:**

\*ROCKTYPE key (COPY old\_key)

### **ARRAY:**

\*THTYPE

### **DEFINITIONS:**

key

Rock property type key. All rock/formation properties listed below are assigned to this rock type key until another \*ROCKTYPE is encountered.

\*COPY old\_key

Initialize the set corresponding to 'key' with values from the set corresponding to 'old\_key'. This is useful when you want two rock types that are the same except for a few properties.

\*THTYPE

Enter a rock type key for each grid block. Only 1 and key values that have been defined are allowed.

### **DEFAULTS:**

The default rock type key value is 1. \*ROCKTYPE is needed only to define multiple rock types.

The default key assigned to each block is 1. \*THTYPE is needed only to assign multiple rock type keys to the grid.

Unless you have multiple rock types, you do not need \*ROCKTYPE or \*THTYPE.

### **CONDITIONS:**

This keyword must be in the Other Reservoir Properties keyword group.

### **EXPLANATION:**

The following rock properties may be assigned values for multiple rock types:

Rock compressibility	- *PRPOR, *CPOR, *CTPOR, *CPTPOR, *CPORPD, *PORMAX, *DILATION, *EPCOMPACT, *COMPACT_VAR
Rock thermal properties	- *ROCKCP, *THCONR, *THCONS, *THCONW, *THCONO, *THCONG, *THCONTAB, *THCONMIX, *ROCKCP_SHL, *THCONR_SHL
Overburden heat loss	- *HLOSSPROP, *HLOSSST, *HLOSSTDIF
Variable permeability	- *PERMCK, *PERMTAB, *PERMTABLOG, *PERMEXP, *PERMSLD

---

## **Formation Compressibility (Optional)**

\*PRPOR, \*CPOR, \*CTPOR,  
\*CPTPOR, \*CPORPD, \*PORMAX, \*PORINTERP, \*PORFORM, \*VOLCONST

### **PURPOSE:**

- \*PRPOR signals the input of a reference pressure for the formation compressibility.
- \*CPOR signals the input of formation compressibility.
- \*CTPOR signals the input of formation thermal expansion.
- \*CPORPD signals the input of pressure-dependent formation compressibility.
- \*PORINTERP specifies the interpretation of input porosity.
- \*PORFORM specifies the form of p and T variation of porosity.
- \*VOLCONST specifies the volume constraint type.

### **FORMAT:**

*PRPOR	<i>prpor</i>
*CPOR	<i>cpor</i>
*CTPOR	<i>ctpor</i>
*CPTPOR	<i>cptpor</i>
*CPORPD	<i>cpor_p2 ppr1 ppr2</i>
*PORMAX	<i>pormax</i>
*PORINTERP	( *REF   *INIT )
*PORFORM	( *EXP   *LINEAR )
*VOLCONST	( *ROCK   *BULK )

### **DEFINITIONS:**

#### *prpor*

Reference pressure (kPa | psi | kPa). The suggested range is from 100 kPa (14.504 psi) to 1.0e6 kPa (1.45e5 psi); *prpor* must be non-negative.

#### *cpor*

Effective formation compressibility, that is, of the formation's pore space (1/kPa | 1/psi | 1/kPa). The lower limit is 0, and the suggested upper limit is 0.01 1/kPa (0.069 1/psi).

#### *ctpor*

Effective thermal expansion coefficient of the formation (1/C | 1/F | 1/C). The lower limit is 0, and the suggested upper limit is 0.01 1/C (0.0056 1/F).

#### *cptpor*

Pressure-temperature cross-term coefficient of the formation effective porosity (1/kPa-C | 1/psi-F | 1/kPa-C).

#### *cpor\_p2*

Effective formation compressibility near ppr2 (1/kPa | 1/psi | 1/kPa). The lower limit is 0, and the suggested upper limit is 0.01 1/kPa (0.069 1/psi).

### *ppr1, ppr2*

Lower (*ppr1*) and upper (*ppr2*) reference pressures for pressure-dependent formation compressibility (kPa | psi | kPa). At *ppr1* the compressibility is nearly *cpor*, and at *ppr2* the compressibility is nearly *cpor\_p2*.

*ppr1* must be non-negative, and *ppr2* must be greater than *ppr1*. The suggested lower limit of *ppr1* is 100 kPa (14.504 psi), and the suggested upper limit of *ppr2* is 1.0e6 kPa (1.45e5 psi).

### *pormax*

Maximum allowed fractional increase in porosity due to pressure. One aspect of sand dilation can be modelled very simply by using a large compressibility, i.e., greater than 0.0001 1/psi. Unphysical porosity increases are avoided by enforcing a maximum porosity fractional increase *pormax*. The value of *pormax* must be greater than zero and less than one. A typical value is 0.10 to 0.20. The default value of 10 effectively disables this limit. This option is considered obsolete and has been replaced by \*DILATION.

### \*PORINTERP ( \*REF | \*INIT )

Per-block porosities specified by keyword \*POR can be interpreted in one of two ways:

- \*REF: reference porosity, at reference pressure \*PRPOR and temperature \*TEMR (if thermal), or
- \*INIT: initial porosity, at initial pressure given by \*PRES or \*VERTICAL and initial temperature given by \*TEMP (if thermal). See **Reference versus Initial Porosity**, below.

### \*PORFORM ( \*EXP | \*LINEAR )

Specify the form with p and T dependence of porosity. Let  $c(p,T)$  be the fractional change in porosity as a function of p and T. For example, for the linear elastic case  $c(p,T) = \min[pormax, cpor(p-prpor)] - ctpor(T-Temr)$ . The porosity formula is:

- \*EXP:  $\phi_v(p,T) = \phi_{vr} \cdot \exp\{c(p,T)\}$
- \*LINEAR:  $\phi_v(p,T) = \phi_{vr} \cdot \{1 + c(p,T)\}$

The \*EXP option is recommended since it cannot give a negative porosity and is more correct according to the definition of compressibility. The historic \*LINEAR option is a good approximation to \*EXP for small  $c(p,T)$ .

### \*VOLCONST ( \*ROCK | \*BULK )

Specify the cell volume constraint type. The bulk volume  $V_b$  of each grid cell is the sum of the rock (grain) volume  $V_r$  and the pore volume  $V_p$ . Since  $V_p$  changes (p, T, dilation), only one of  $V_b$  and  $V_r$  can remain constant.

**\*ROCK:** Rock volume  $V_r$  is constant and bulk volume changes according to  $V_b = V_p + V_r$ . This preserves the cell's rock mass and hence rock heat capacity at a given T.

**\*BULK:** Bulk volume  $V_b$  is constant and the rock volume changes according to  $V_r = V_b - V_p$ . This changes the rock mass and hence rock heat capacity, which can result in T changes due solely to external forces like dilation.

### DEFUALTS:

If \*PRPOR is absent for all rock types, *prpor* of each rock type is equal to the initial pressure in the first (in natural ordering) active block in the rock type. If \*PRPOR is present for at least one rock type, the first *prpor* is assigned to all rock types and each subsequent *prpor* is assigned to its associated rock type (overwriting the first *prpor*). For example, if a data set has multiple rock types define by \*ROCKTYPE but \*PRPOR appears only once, all rock types will use the *prpor* value specified by \*PRPOR.

If \*CPOR is absent, the formation compressibility is zero.

If \*CTPOR is absent, the formation thermal expansion coefficient is zero.

If \*CPTPOR is absent, *cptpor* = 0 is assumed.

If \*PORMAX is absent, the corresponding option is disabled.

If \*CPORPD is absent, the corresponding option is disabled.

If \*PORINTERP is absent then option \*REF is assumed.

If \*PORFORM is absent then \*LINEAR is assumed.

If \*VOLCONST is absent then \*BULK is assumed.

### CONDITIONS:

This keyword must be in the Other Reservoir Properties keyword group.

Keywords \*CPTPOR and \*CPORPD may not be used together.

### EXPLANATION:

Fluid porosity  $\phi_f$  contains the fluid phases but not the solid phase and is calculated as

$$\phi_f(p, T, C_i) = \phi_v(p, T) * (1 - \sum C_i / \rho_{si})$$

$\phi_v$	- void porosity at p and T,
p	- fluid pressure,
T	- temperature,
$C_i$	- component solid concentration in the pore space, and
$\rho_{si}$	- component solid density from *SOLID_DEN.

There are several ways to calculate void porosity  $\phi_v$  from pressure and temperature. The porosity is shown for \*PORFORM \*EXP but is available for \*PORFORM \*LINEAR, too.

1. **Linear Elastic:** Use \*CPOR for pressure dependence:

$$\phi_v(p,T) = \phi_{vr} \cdot \exp \{ \min[ pormax, cpor(p-prpor) ] - ctpor(T-Temr) \}$$

$\phi_{vr}$  - void porosity at reference  $prpor$  and  $Temr$  (see \*POR)  
 $p$  - fluid pressure,  
 $T$  - temperature, and  
 $Temr$  - reference temperature from \*TEMR.

2. **Nonlinear Elastic:** Use \*CPOR and \*CPORPD for pressure dependence:

$$\phi_v(p,T) = \phi_{vr} \cdot \exp \{ \min[ pormax, cpor(p-prpor)+cpord ] - ctpor(T-Temr) \}$$

$\phi_{vr}$  - void porosity at reference  $prpor$  and  $Temr$  (see \*POR)  
 $p$  - fluid pressure,  
 $T$  - temperature, and  
 $Temr$  - reference temperature from \*TEMR.  
 $cpord$  -  $A * [ D * (p - prpor) + \ln( B / C ) ]$   
 $A$  -  $(cpor\_p2 - cpor) / D$   
 $B$  -  $1 + \exp [ D * (pav - p) ]$   
 $C$  -  $1 + \exp [ D * (pav - prpor) ]$   
 $D$  -  $10 / (ppr2 - ppr1)$   
 $pav$  -  $(ppr1 + ppr2) / 2$

Example: \*CPOR 0 \*PRPOR 5000 \*CPORPD 1.0e-5 5000 9000

The fractional contribution **cpord** is as follows. Above  $p = ppr2$  the compressibility is equal to  $cpor\_p2$ , and below  $ppr1$  it is nearly 0.

<b>p</b>	<b>cpord</b>	<b>Note</b>
1000	-2.68e-5	Low pressure
5000	0	$P = PRORP = PPR1$
7000	2.75e-3	$P = Pav$
9000	2.00e-2	$P = PPR2$
15000	8.00e-2	High pressure

3. **P-T Cross Term:** Use \*CPOR, \*CTPOR and \*CPTPOR.

$$\phi_v(p,T) = \phi_{vr} \cdot \exp \{ \min[ pormax, cpor(p-prpor) + cptpor(p-prpor) \cdot (T-Temr) ] - ctpor(T-Temr) \}$$

4. **Dilation-Recompaction:** Use keyword group \*DILATION
5. **Compaction-Rebounding using constant pore compressibility and thermal expansion coefficient:** Use keyword group \*EPCOMPACT
6. **Compaction-Rebounding using pressure history dependent pore compressibility and thermal expansion coefficient:** Use keyword group \*COMPACT\_VAR
7. **Constitutive Geomechanics:** Use keyword group \*GEOMECH for advanced geomechanical effects.

### **Reference versus Initial Porosity**

The default \*PORINTERP option \*REF causes a block value specified by keyword \*POR to be interpreted as  $\phi_{vr}$  which is used directly in the above porosity formulas. With this option the porosities reported at initial conditions may differ from the \*POR values.

\*PORINTERP option \*INIT causes a block value specified by keyword \*POR to be interpreted as  $\phi_v(p_i, T_i)$ , that is, the porosity at initial pressure  $p_i$  and temperature  $T_i$ . In this case each block's reference porosity  $\phi_{vr}$  is back-calculated using the above formulas. This is done for all pressure initialization options and all porosity options. With this option the porosities reported at initial conditions are the \*POR values.

---

## **Reservoir Pore Volume Dilation-Recompaction (Optional)**

**\*DILATION, \*PBASE, \*PDILA, \*PPACT, \*CRD, \*FR, \*PORRATMAX, \*CPEPAC, \*CTD,  
\*CTPPAC**

### **PURPOSE:**

Define a reservoir pore volume dilation-recompaction model.

### **FORMAT:**

**\*DILATION      (\*PBASE *pbase*) (\*PDILA *pdila*) (\*PPACT *ppact*)  
                  (\*CRD *crd*) (\*FR *fr*) (\*PORRATMAX *rat*)  
                  (\*CPEPAC *cpepac*) (\*CTD *ctd*) (\*CTPPAC *ctppac*)**

### **ARRAY:**

**\*PERMULI  
\*PERMULJ  
\*PERMULK**

Grid block permeability multipliers in I, J and K direction. For details, see descriptions in **Variable Permeability** section.

### **DEFINITIONS:**

#### **\*DILATION**

Keyword indicating that the reservoir dilation/recompaction model is enabled, and that other keywords in this group may follow.

#### *pbase*

Reference pressure for elastic curve (kPa | psi | kPa). The suggested range is from 100 kPa (14.504 psi) to 1.0e6 kPa (1.45e5 psi); the value must be non-negative.

#### *pdila*

Pressure at which dilation begins (kPa | psi | kPa). The value must be non-negative and should be larger than the reservoir initial pressure values.

#### *ppact*

Pressure at which recompaction begins (kPa | psi | kPa). The value must be non-negative and should be less than *pdila*.

#### *crd*

Dilation pore volume compressibility (1/kPa | 1/psi | 1/kPa). The value must be non-negative. A zero value will disable the dilation option.

#### *fr*

Residual dilation fraction, i.e., the fraction of total dilation not recovered on recompaction. The allowed range is 0 to 1.

*rat*

Maximum allowed proportional increase in porosity, applied individually to each block's initial reference porosity. The minimum allowed value of *rat* is 1. The maximum recommended value of *rat* is 1.3; much larger values can result in severe convergence problems.

*cpepac*

Pore volume compressibility of the elastic compaction curve (1/kPa | 1/psi | 1/kPa). The value must be non-negative.

*ctd*

Pore volume thermal expansion coefficient for dilation curve (1/C | 1/F | 1/C). The value must be non-negative.

*ctppac*

Pore volume thermal expansion coefficient for recompaction curve (1/C | 1/F | 1/C). The value must be non-negative.

## DEFUALTS:

If \*DILATION is absent, the dilation-recompaction model is disabled and the subkeywords of \*DILATION are disallowed.

If \*PBASE is absent its value is assumed to be equal to that given by \*PRPOR or its default.

Absent	Action
*PDILA	<i>pdila</i> = 0
*PPACT	<i>ppac</i> = 0
*CRD	<i>crd</i> = 0 which will disable the dilation-recompaction model.
*FR	<i>fr</i> = 0
*PORRATMAX	<i>rat</i> = 1
*CPEPAC	<i>cpepac</i> = <i>cpor</i> from *CPOR (initial elastic curve value).
*CTD	<i>ctd</i> = <i>cpor</i> from *CTPOR.
*CTPPAC	<i>ctppac</i> = <i>cpor</i> from *CTPOR.
*PERMULI	All grid permeability multipliers in I-direction are zero, i.e. no corrections for I-direction permeability from porosity change.
*PERMULJ	All grid permeability multipliers in J-direction are zero, i.e. no corrections for J-direction permeability from porosity change.
*PERMULK	All grid permeability multipliers in K-direction are zero, i.e. no corrections for K-direction permeability from porosity change.

## CONDITIONS:

\*PBASE, \*PDILA, \*PPACT, \*CRD, \*FR, \*PORRATMAX, \*CPEPAC, \*CTD and \*CTPPAC are subkeywords of \*DILATION and so must be located immediately after \*DILATION but may appear in any order.

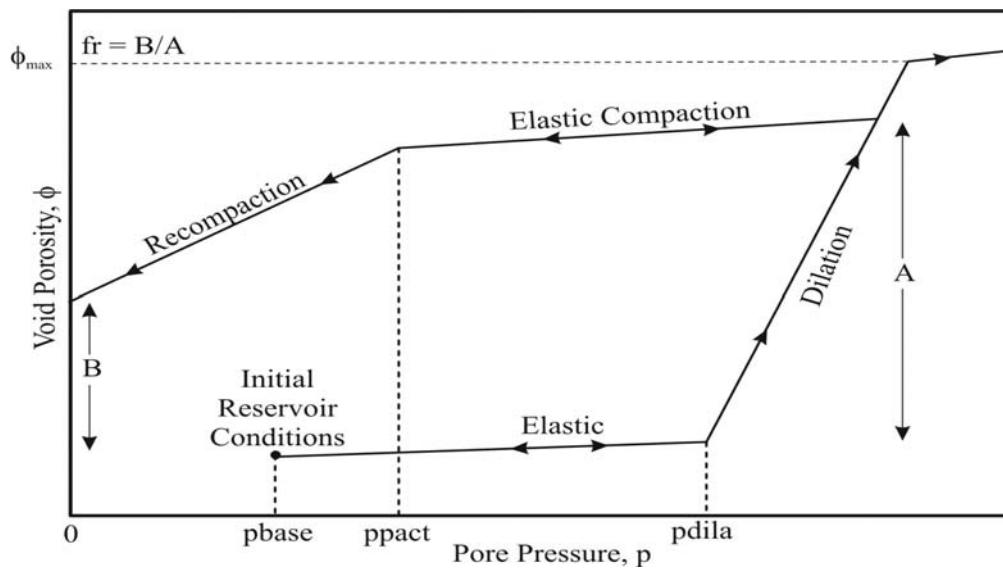
All subkeywords of \*DILATION are indexed by rock type, and their values are assigned to the current rock type number (see keyword \*ROCKTYPE).

These keywords are mutually exclusive for each rock type: \*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMEXP, \*COMPACTION, \*COMPACTION\_VAR and \*EPCOMPACT.

## EXPLANATION:

### Dilation/Recompaction Model

The dilation-recompaction model represents the main features of oil-sand dilation and recompaction occurring during cyclic steam stimulation process. The model is based on the work of Beattie, Boberg and McNab in "Reservoir Simulation of Cyclic Steam Stimulation in the Cold Lake Oil Sands", SPE Reservoir Engineering, May, 1991. In this model, the relationship between grid block void porosity,  $\phi$ , and grid block pressure,  $p$ , is depicted in Figure 13 below and the porosity  $\phi$  at any known pressure  $p$  is calculated using Eq. (D1).



*Figure 13: The Dilation-Recompaction Model for Cyclic Steam Stimulation Process*

$$\phi = \phi_{ref} e^{[c_p(p - p_{ref})]} \quad (D1)$$

In Eq. (D1),  $p_{ref}$  is a reference pressure,  $\phi_{ref}$  is the porosity at  $p_{ref}$  and  $c_p$  is the pore volume compressibility. There is a set of these three grid block quantities for each branch of the deformation curve shown in Figure 13.

As pressure increases due to steam injection from its initial reservoir condition the rock behaves elastically. If pressure continues to increase to exceed  $pdila$ , then porosity follows the irreversible dilation curve until either pressure declines or the assigned maximum porosity is reached. If pressure decreases from a point on the dilation curve, porosity follows a elastic compaction curve initially. As pressure decreases further below the recompaction pressure  $ppact$ , recompaction occurs and the slope of the curve is determined by the specified residual dilation fraction  $fr$ . Another similar dilation/recompaction cycle is started when pressure increases from a point on the recompaction curve, as shown in Figure 13.

### **Temperature Effect on Dilation-Recompaction Model**

To include thermal effect on grid block pore volume, the original formula Eq. (D1) is modified as follows

$$\phi = \phi_{ref} e^{[c_p(p - p_{ref}) - c_T(T - T_{ref})]} \quad (D2)$$

where  $c_T$ , the thermal expansion coefficient, will assume the value of  $ctpor$ ,  $ctd$  or  $ctppac$  depending on the grid block pressure and its history. A general precaution for applying the temperature factor considered here is that its effect on grid porosity should be much less than that of pressure. And therefore the pore volume dilation-recompaction behavior is still dominated by pressure. This requires that the inputted value of  $ctpor$ ,  $ctd$  and  $ctppac$  should be considerably smaller than the corresponding pore volume compressibility.

### **Variable Permeability**

A grid cell's absolute permeability in each direction may depend upon porosity, effectively varying in a manner similar to that shown in Figure 13. Keywords \*PERMULI, \*PERMULJ and \*PERMULK may be used with \*DILATION to specify permeability variation on both a per-block and per-direction basis. See the explanation for keyword \*PERMEXP.

### **Solid Phase Effect on Permeability**

The porosity used in the optional variation of absolute permeability is the fluid porosity, and not the void porosity. Since the fluid volume is the void volume minus the solid phase volume, changes in the amount of material (solid components or adsorbed/trapped fluid components) in the solid phase will have a direct effect on the calculated permeability. For example, the appearance of coke in a combustion process can decrease the fluid porosity and hence the permeability.

### **EXAMPLE:**

The following keywords inputted in the section of Other Reservoir Properties will enable the dilation-recompaction model with no temperature effect on grid block pore volume.

```
*DILATION *PBASE 75.0 *PDILA 400.0 *PPACT 100.0
      *CRD 7e-6 *FR 0.1 *PORRATMAX 1.3
*PERMULI *CON 4.5
*PERMULJ *CON 4.5
*PERMULK *CON 4.5
```

And the example below will enable the dilation-recompaction model with both pressure and temperature effect on the grid block pore volume but not on the absolute permeability.

```
*DILATION *PDILA 400.0 *PPACT 100.0  
*CRD 7e-5 *FR 0.5 *PORRATMAX 1.2  
*CTD 7e-6 *CTPPAC 1e-6
```

Please note that to include the temperature effect on the initial elastic curve and the elastic compaction curve, a non-zero elastic thermal expansion coefficient has to be inputted via the keyword \*CTPOR.

---

## Reservoir Pore Volume Compaction Rebounding (Optional)

\*EPCOMPACT, \*CRP, \*PPLASTIC, \*CTP

### PURPOSE:

Define a reservoir compaction-rebounding model with elastic-plastic deformations via constant compressibility and thermal expansion coefficient.

### FORMAT:

\*EPCOMPACT ( \*CRP  $c_{rp}$  ) ( \*PPLASTIC  $pplastic$  ) ( \*CTP  $c_{Tp}$  )

### ARRAY:

\*PERMULI  
\*PERMULJ  
\*PERMULK

Grid block permeability multipliers in I, J and K direction. For details, see descriptions in **Variable Permeability** section.

### DEFINITIONS:

#### \*EPCOMPACT

Keyword indicating that the elastic-plastic compaction-rebounding model is enabled, and that other keywords in this group will follow.

$c_{rp}$

Formation pore volume compressibility for plastic compaction (1/kPa | 1/psi | 1/kPa). The value must be non-negative.

$c_{Tp}$

Formation pore volume thermal expansion coefficient for plastic compaction (1/C | 1/F | 1/C). The value must be non-negative.

$pplastic$

Threshold pressure at which plastic compaction begins (kPa | psi | kPa). The value must be non-negative and should be smaller than reservoir initial pressure values.

### DEFAULTS:

If \*EPCOMPACT is absent, the elastic-plastic compaction model is disabled and subkeywords \*CRP, \*PPLASTIC and \*CTP are disallowed.

If \*EPCOMPACT is present but \*CRP or \*PPLASTIC is absent, the corresponding data value is zero.

If \*EPCOMPACT is present but \*CTP is absent,  $c_{Tp} = c_{tpor}$  from \*CTPOR (elastic value).

If any of \*PERMULI, PERMULJ or PERMULK is absent, there will be no permeability corrections in that direction due to porosity change.

## CONDITIONS:

\*CRP, \*CTP and \*PPLASTIC are subkeywords of \*EPCOMPACT and so must be located immediately after \*EPCOMPACT but may appear in any order.

All subkeywords of \*EPCOMPACT are indexed by rock type, and their values are assigned to the current rock type number (see keyword \*ROCKTYPE).

The following options are mutually exclusive for each rock type: \*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMEXP, \*DILATION, \*COMPACT\_VAR and \*EPCOMPACT.

## EXPLANATION:

The compaction-rebounding model is primarily intended to simulate the irreversible process of formation shrinkage due to pressure decline in primary depletion and rebound due to pressure rise by a possible subsequent injection period. Under this option, the effect of compaction or rebound on fluid flow is modelled in STARS by the change of pore volume porosity. Figure 14 below schematically shows the behavior of grid block porosity on pressure changes. As the pressure starts to decline from the initial reservoir condition, the rock deforms elastically and the porosity decreases due to the elastic compressibility and thermal expansion (the value from keyword \*CPOR and \*CTPOR). If the pressure reduces further below a threshold pressure ( $p_{plastic}$ ), some unrecoverable compaction will occur and the porosity changes plastically by the plastic values of  $c_{rp}$  and  $c_{Tp}$ . Unlike the elastic period, the plastic compaction is an irreversible process; that is, as the pressure rises the porosity will follow a rebounding curve which is branched out from the plastic compaction, instead of re-traversing the original compaction curve.

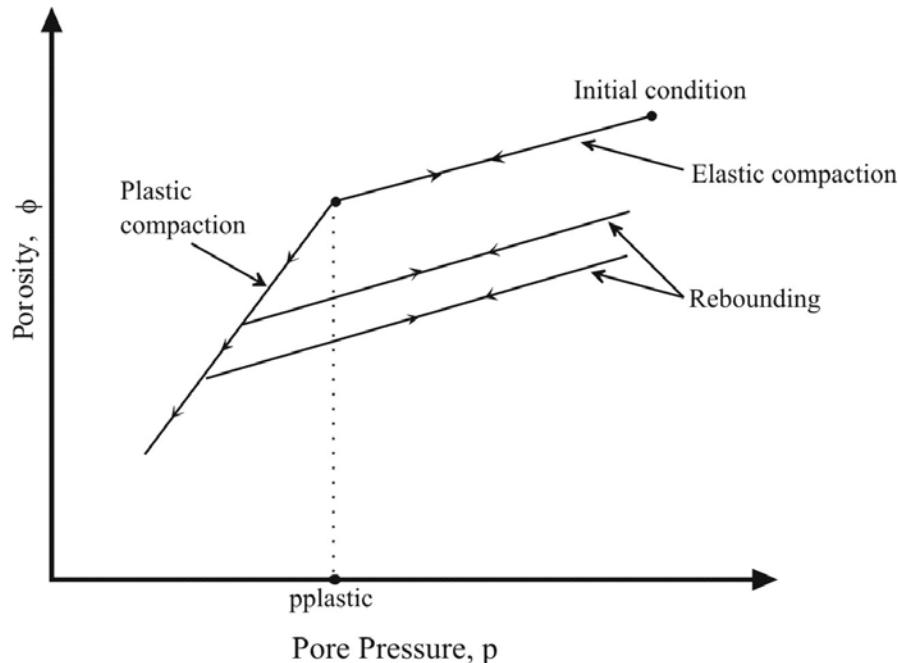


Figure 14: Rock compaction-rebounding model (\*EPCOMPACT)

The functional form used for the porosity-pressure/temperature relationship in the calculation is

$$\phi = \phi_{\text{ref}} e^{[c_p(p - p_{\text{ref}}) - c_T(T - T_{\text{ref}})]}$$

where  $c_p$  represents the compressibility for elastic or plastic compaction,  $c_T$  represents the thermal expansion coefficient for elastic or plastic compaction,  $p_{\text{ref}}$  and  $T_{\text{ref}}$  are the reference pressure and temperature of each curve, and  $\phi_{\text{ref}}$  is the porosity at  $p_{\text{ref}}$  and  $T_{\text{ref}}$ . Please note that the  $p_{\text{ref}}$  and  $T_{\text{ref}}$  vary as the grid blocks move from elastic to plastic, and vice versa.

The formation subsidence is calculated along with the porosity changes. The amount of subsidence can be examined via subkeyword \*SBDZ of \*OUTPRN \*GRID, OUTSRF \*GRID and \*OUTSRF \*SPECIAL \*BLOCKVAR, etc.

### **Temperature Effect on \*EPCOMPACT Model**

A general assumption of the temperature factor considered here is that its effect on grid porosity is much less than that of pressure. So the pore volume compaction-rebound behavior is dominantly controlled by pressure. This requires that the inputted value of  $c_{\text{tp}}$  and  $c_{\text{tp}}$  should be considerably smaller than the corresponding pore volume compressibility  $c_{\text{por}}$  and  $c_{\text{rp}}$ .

### **Variable Permeability**

A grid cell's absolute permeability in each direction may depend upon porosity, effectively varying in a manner similar to that shown in Figure 14. Keywords \*PERMULI, \*PERMULJ and \*PERMULK may be used with \*EPCOMPACT to specify permeability variation on both a per-block and per-direction basis. See the explanation for keyword \*PERMEXP.

### **EXAMPLE:**

The following keywords inputted in the section of Other Reservoir Properties will enable the compaction-rebounding model with no temperature effect on grid block pore volume ( $c_{\text{tp}} = 0$ ).

```
*EPCOMPACT *CRP 1.0e-5 *PPLASTIC 1500  
*PERMULI *CON 10  
*PERMULJ *CON 10  
*PERMULK *CON 10
```

And the example below will enable the compaction-rebounding model with both pressure and temperature effect on the grid block pore volume but not on the absolute permeability

```
*EPCOMPACT *CRP 1.0e-5 *PPLASTIC 1500 *CTP 1.0e-6
```

Please note that to include the temperature effect on the elastic compaction and rebounding, a non-zero elastic thermal expansion coefficient has to be inputted via the keyword \*CTPOR.

---

## **Compaction-Rebounding With Variable cp and cT (Optional)**

**\*COMPACT\_VAR, \*CRB, \*CRM, \*ALFDR, \*FONE, \*FTWO, \*CPERMCOR**

### **PURPOSE:**

Define a pore volume compaction-rebounding model via the pressure history dependent pore compressibility and thermal expansion coefficient.

### **FORMAT:**

**\*COMPACT\_VAR      (\*CRB  $c_{rb}$ )    (\*CRM  $c_{rm}$ )    (\*ALFDR  $a_{dr}$ )    (\*FONE  $f_1$ )    (\*FTWO  $f_2$ )  
                          (\*CPERMCOR  $a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5$ )**

### **DEFINITIONS:**

#### **\*COMPACT\_VAR**

Keyword indicating that the compaction-rebounding model using variable pore compressibility and thermal expansion coefficient is enabled, and that other keywords in this group will follow.

$c_{rb}$

Formation bulk compressibility of the porous structure (1/kPa | 1/psi | 1/kPa). The value must be non-negative.

$c_{rm}$

Formation matrix (grain) compressibility of the porous structure (1/kPa | 1/psi | 1/kPa). The value must be non-negative.

$a_{dr}$

Formation bulk thermal expansion coefficient of the porous structure (1/C | 1/F | 1/C). The value must be non-negative.

$f_1$

Parameter in the compaction formulation. See Eq. (G2) and (G3) below

$f_2$

Parameter in the rebounding formulation. See Eq. (G4) and (G5) below

$a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5$

Coefficients in the permeability-porosity correlation. See Eq. (G6) below

### **DEFAULTS:**

If **\*COMPACT\_VAR** is absent, the compaction-rebound model with variable compressibility and thermal expansion is disabled and the subkeywords of **\*COMPACT\_VAR** are disallowed.

<b>Absent</b>	<b>Action</b>
*CRB	$c_{rb} = 0$
*CRM	$c_{rm} = 0$
*ALFDR	$a_{dr} = 0$
*FONE	$f_1 = 0$
*FTWO	$f_2 = 0$
*CPERMCOR	$a_0 = 1, a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 0, a_5 = 0$

### **CONDITIONS:**

\*CRB, \*CRM, \*ALFDR, \*FONE, \*FTWO and \*CPERMCOR are subkeywords of \*COMPACT\_VAR and so must be located immediately after \*COMPACT\_VAR but may appear in any order.

All subkeywords of \*COMPACT\_VAR are indexed by rock type, and their values are assigned to the current rock type number (see keyword \*ROCKTYPE).

The following options are mutually exclusive for each rock type: \*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMEXP, \*DILATION, \*COMPACTION\_VAR and \*EPCOMPACT.

### **EXPLANATION:**

The compaction-rebound model with variable compressibility and thermal expansion coefficient is based on the work of Zhengming Yang et al. (For details, see SPE 49314 "Method of handling the complexities associated with history matching the performance of a highly stress-sensitive formation", 1998). This method attempted to model the phenomena of sand compaction due to pressure depletion and the formation rebounding thereafter from the steam injection. Based on a geomechanical stress-strain analysis, some simplified relationship is derived for the conventional compressibility and thermal expansion coefficient so that a full coupled solution of reservoir flow equations and the geomechanical stress-strain equations is avoided.

The formation void porosity can generally be expressed as

$$\phi(p, T) = \phi_{ref} [1 + c_p (p - p_{ref}) - c_T (T - T_{ref})] \quad (G1)$$

where  $p$  and  $T$  are grid block pressure and temperature (the subscript "ref" stands for the values at the reference condition).  $\phi$  is the grid block void porosity, and  $c_p$  and  $c_T$  are the pore volume compressibility and thermal expansion coefficient, respectively.

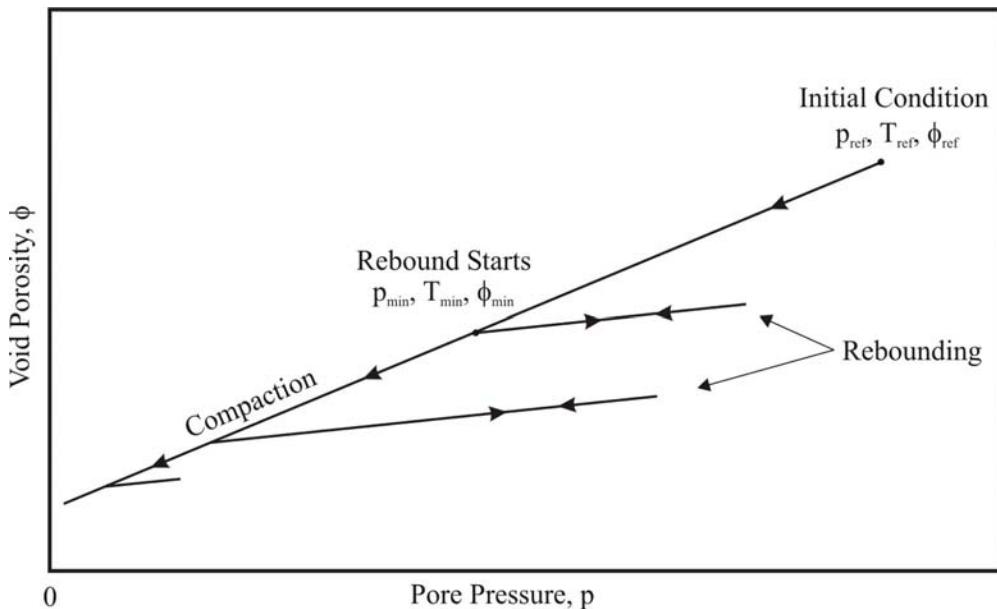
The final formula used for calculating  $c_p$  and  $c_T$  in Eq. (G1) during the reservoir compaction period are

$$c_p = \frac{[c_{rb}(1 - \phi_{ref}) - c_{rm}](1 - f_1)}{\phi_{ref}} + [(1 - f_1)c_{rb} - c_{rm}] + \frac{[c_{rb}(1 - \phi_{ref}) - c_{rm}](1 - f_1)}{\phi_{ref}} + [(1 - f_1)c_{rb} - c_{rm}](p - p_{ref}) \quad (G2)$$

and

$$c_T = a_{dr} \left\{ 1 + \frac{[c_{rb}(1 - \phi_{ref}) - c_{rm}](1 - f_1)(p - p_{ref})}{\phi_{ref}} \right\} \quad (G3)$$

In Eq.(G2) and (G3), the parameter  $f_1$  is a derived variable related to geomechanical parameters (such as Poisson's ratio) and equals to -0.44 approximately for oil sand.



**Figure G1. Compaction-rebounding model using variable compressibility and thermal expansion**

Formation will rebound when grid block pressure increases as shown in Figure G1. Assuming rebounding starts at the pressure  $p_{min}$  and temperature  $T_{min}$  with the corresponding porosity  $\phi_{min}$ , the following formulas are used in computing  $c_p$  and  $c_T$  during the rebounding period

$$\begin{aligned} c_p &= \frac{[c_{rb}(1 - \phi_{min}) - c_{rm}](1 - f_2)}{\phi_{min}} + [(1 - f_2)c_{rb} - c_{rm}] \\ &\quad + \frac{[c_{rb}(1 - \phi_{min}) - c_{rm}](1 - f_2)}{\phi_{min}} + [(1 - f_2)c_{rb} - c_{rm}](p - p_{min}) \end{aligned} \quad (G4)$$

and

$$c_T = a_{dr} \left\{ 1 + \frac{[c_{rb}(1 - \phi_{min}) - c_{rm}](1 - f_2)(p - p_{min})}{\phi_{min}} \right\} \quad (G5)$$

Similar to  $f_1$  for formation compaction,  $f_2$  is a parameter related to the geomechanical properties. For the cases where  $c_{rm} \ll c_{rb}$ ,  $f_2 \approx 0.5$ .

The grid block absolute permeability is correlated to the porosity in a form of polynormal and written as

$$K(p, T) = k_{ref} \left[ a_0 + a_1 \left( \frac{\phi}{\phi_{ref}} \right) + a_2 \left( \frac{\phi}{\phi_{ref}} \right)^2 + a_3 \left( \frac{\phi}{\phi_{ref}} \right)^3 + a_4 \left( \frac{\phi}{\phi_{ref}} \right)^4 + a_5 \left( \frac{\phi}{\phi_{ref}} \right)^5 \right] \quad (G6)$$

In Eq. (G6),  $a_0 - a_5$  are user inputted matching factors and  $K_{ref}$  is the grid block permeability at the reference condition. To satisfy the initial condition, it should have  $a_0 + a_1 + a_2 + a_3 + a_4 + a_5 = 1.0$ .

The formation subsidence is calculated along with the porosity changes. The amount of subsidence can be examine via subkeyword \*SBDZ of \*OUTPRN \*GRID, OUTSRF \*GRID and \*OUTSRF \*SPECIAL \*BLOCKVAR, etc.

#### **EXAMPLE:**

The following keywords inputted in the section of Other Reservoir Properties will enable the compaction-rebounding model with variable compressibility and thermal expansion.

```
*COMPACT_VAR *CRB 1.36E-6 *CRM 1e-7 *ALFDR 1e-7
*FONE -0.44 *FTWO 0.5
*CPERMCOR 1.0 -0.5 0.5 0.0 0.0 0.0
```

---

## Variable Permeability (Optional)

\*PERMCK, \*PERMTAB,

\*PERMTABLOG, \*PERMEXP, \*PERMULI, \*PERMULJ, \*PERMULK, \*PERMSLD

### PURPOSE:

Specify dependence of permeability on fluid porosity.

### FORMAT:

```
*PERMCK ckpower
*PERMTAB
  { $\phi/\phi_o$  K/Ko }
*PERMTABLOG
  { $\phi/\phi_o \ln(K/K_o)$  }
*PERMEXP
*PERMSLD
```

### ARRAY:

```
*PERMULI
*PERMULJ
*PERMULK
```

### DEFINITIONS:

#### \*PERMCK *ckpower*

Permeability is a function of fluid porosity via the Carmen-Kozeny type formula

$$K(\phi) = K_o * [\phi/\phi_o]^{**ckpower*[(1-\phi_o)/(1-\phi)]**2}$$

The lower limit of *ckpower* is 0, and the upper limit is 10.

#### \*PERMTAB

Permeability is a function of fluid porosity via a permeability multiplier obtained from table look-up. Between table entries, K/K<sub>o</sub> varies linearly with  $\phi/\phi_o$ . There must be one table row with  $\phi/\phi_o = 1$  and  $K/K_o = 1$ . In the table,  $\phi/\phi_o$  must be non-negative and increasing, and K/K<sub>o</sub> must be non-negative. The allowed number of rows is 2 to 30. Entries in the first column must be evenly spaced; if not they will be adjusted.

#### \*PERMTABLOG

Permeability is a function of fluid porosity via a permeability multiplier obtained from the exponent of a table look-up. Between table entries,  $\ln(K/K_o)$  varies linearly with  $\phi/\phi_o$ . There must be one table row with  $\phi/\phi_o = 1$  and  $\ln(K/K_o) = 0$ . In the table,  $\phi/\phi_o$  must be non-negative and increasing. The allowed number of rows is 2 to 30. Entries in the first column must be evenly spaced; if not they will be adjusted.

#### \*PERMEXP

Permeability is the following function of fluid porosity:

$$k = k_o * \exp \left[ k_{mul} * \left( \frac{\phi - \phi_o}{1 - \phi_o} \right) \right].$$

Here,  $k_o$  and  $\phi_o$  are the original (initial) permeability and fluid porosity, respectively, and  $k_{mul}$  is a user-defined multiplier factor specified by \*PERMULI, \*PERMULJ or \*PERMULK. This calculation is done on a per-block and per-direction basis. See \*PERMULI below.

#### \*PERMSLD

This keyword is used only for permeability change due to solid in a Discretized Wellbore (DW). Permeability is the following function of fluid porosity:

$$k = k_o * (1 - \text{solid}(\phi))$$

Here,  $k_o$  is initial wellbore permeability. Solid fraction (solid volume/void volume) is a function of solid concentration, void porosity and solid compressibility. **Do not use** any other variable permeability keyword for the DW blocks.

#### \*PERMULI, \*PERMULJ, \*PERMULK

Permeability variation parameters for the I, J and K grid directions, for use in the formula shown for \*PERMEXP, above. Note that these are grid arrays, and so different values may be entered for individual grid blocks. The allowed values are 0 to 1.0e10, but normally values do not exceed about 100 since the factor appears inside the exponential function.

All grid array data assignment options are allowed, including \*EQUALSI. The values for all three directions are initialized internally to 0 which corresponds to no permeability variation with fluid porosity. Only those blocks explicitly given non-zero values, in the each direction separately, will experience permeability variations.

See **Direction Dependent Data** in the description for keyword \*REFINE.  
See also **Multiple Associations of \*PERMULI** below.

#### DEFUALTS:

If any of \*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMSLD and \*PERMEXP is absent, the corresponding model is not used to vary permeability with fluid porosity. Other options may be used to vary permeability (e.g., \*DILATION and \*EPCOMPACT).

\*PERMULI \*CON 0  
 \*PERMULJ \*CON 0  
 \*PERMULK \*CON 0

## **CONDITIONS:**

Keywords \*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMSLD and \*PERMEXP are indexed by rock type, and their values and flags are assigned to the current rock type number (see keyword \*ROCKTYPE). Different rock types may have different variable permeability options in the same run. If more than one of these keywords appears for the same rock type, the last one encountered is used.

Option \*PERMEXP requires keywords \*PERMULI, \*PERMULJ and \*PERMULK to specify the block dependence of the permeability multiplier.

The following options are mutually exclusive for each rock type: \*PERMCK, \*PERMTAB, \*PERMTABLOG, \*PERMEXP, \*COMPACT\_VAR, \*DILATION and \*EPCOMPACT.

## **EXPLANATION:**

In addition to the options described above, the \*DILATION and \*EPCOMPACT options may be used to vary permeability via \*PERMULI, etc. However, the \*PERMCK, \*PERMTAB, \*PERMTABLOG and \*PERMEXP options do not use the complex void porosity model of the \*DILATION and \*EPCOMPACT options, so changes in fluid porosity, and hence permeability, usually are caused largely by changes in the amount of material in the solid/adsorbed/trapped phase.

Initial fluid porosity  $\phi_o$  corresponds to the value for the block calculated at its initial pressure, temperature and solid amounts in place. Initial permeability  $k_o$  is equal to the block's reference permeability entered via keywords \*PERMI, \*PERMJ and \*PERMK.

### **Multiple Associations of \*PERMULI, etc.**

Keywords \*PERMEXP, \*DILATION and \*EPCOMPACT are mutually exclusive for a rock type (in fact, \*DILATION, \*EPCOMPACT and \*COMPACT\_VAR are mutually exclusive for a simulation run), even though all three are associated with keywords \*PERMULI, \*PERMULJ and \*PERMULK. This is possible because keywords \*PERMEXP, \*DILATION and \*EPCOMPACT are "per-rock-type" whereas \*PERMULI, etc., are "per-block" grid arrays.

Consider the following example. A grid system has three K layers belonging to three rock types, two layers assigned the dilation-recompaction model and one layer as the standard elastic compressibility model. At the same time, the variable permeability option is inputted. Note that \*PERMEXP is used to enable permeability variation in rock type 2 with the standard compressibility model.

```
*ROCKTYPE 1 ** Dilation-recompaction model
  *CPOR 1.8e-5 *DILATION ...
*ROCKTYPE 2 ** Standard compressibility model
  *CPOR 2.5e-5 *PERMEXP ...
*ROCKTYPE 3 ** Dilation-recompaction model
  *CPOR 1.8e-5 *DILATION ...
*PERMULI *KVAR 2.5 3.2 2.8
*PERMULJ *EQUALSI
*PERMULK *KVAR 3.5 3.8 2.9
```

## Examples

The Carmen-Kozeny option requires a single number. In the following example, values of  $\phi_o = 0.30$  and  $\phi = 0.32$  give

$$K/K_o = (0.32/0.3)^{2.5} \cdot [(1-0.3)/(1-0.32)]^2 = 1.245.$$

\*PERMCK 2.5

The \*PERMTAB table option has linear interpolation. In the following example, a value of  $\phi/\phi_o = 0.96$  gives

$$K/K_o = 0.08 + (0.96-0.94) \cdot [(0.2-0.08)/(0.97-0.94)] = 0.16.$$

*PERMTAB	** phi/phi0	perm/perm0
	0.91	0.02
	0.94	0.08
	0.97	0.2
	1.00	1.0
	1.03	3.0

The \*PERMTABLOG table option has linear-log interpolation. In the following example, a value of  $\phi/\phi_o = 0.96$  gives

$$\ln(K/K_o) = -1.5 + (0.96-0.94) \cdot [(-0.8-(-1.5))/(0.97-0.94)] = -1.0333$$

$$K/K_o = 0.3558$$

*PERMTABLOG	** phi/phi0	ln(perm/perm0)
	0.91	-1.8
	0.94	-1.5
	0.97	-0.8
	1.00	0.0
	1.03	0.3

The \*PERMEXP option uses \*PERMULI, \*PERMULJ and \*PERMULK. In the following example, for grid layer k=1 values of  $\phi_o = 0.30$  and  $\phi = 0.32$  give

$$K/K_o = \exp[2.5 \cdot (0.32-0.3)/(1-0.3)] = 1.074 \text{ in the I and J directions, and}$$

$$K/K_o = \exp[3.5 \cdot (0.32-0.3)/(1-0.3)] = 1.105 \text{ in the K direction.}$$

\*PERMULI \*KVAR 2.5 3.2 2.8  
\*PERMULJ \*EQUALSI  
\*PERMULK \*KVAR 3.5 3.8 2.9  
\*PERMEXP

## **Rock Heat Capacity (Optional)**

\*ROCKCP, \*ROCKCP\_SHL

### **PURPOSE:**

Specify rock heat capacity.

### **FORMAT:**

\*ROCKCP *rock\_cp1* (*rock\_cp2*)  
\*ROCKCP\_SHL *shale\_cp1* (*shale\_cp2*)

### **DEFINITIONS:**

\*ROCKCP *rock\_cp1* (*rock\_cp2*)

Coefficients in the correlation (*rock\_cp1* + *rock\_cp2*·T<sub>abs</sub>) for volumetric heat capacity of solid formation (rock) in the reservoir, where T<sub>abs</sub> is absolute degrees. If *rock\_cp2* is absent or zero then *rock\_cp1* is the rock heat capacity. The value of the correlation must be positive over the expected T<sub>abs</sub> range.

The unit of *rock\_cp1* is (J/m<sup>3</sup>-C | Btu/ft<sup>3</sup>-F | J/cm<sup>3</sup>-C) and the suggested range is from 0 to 10<sup>8</sup> J/m<sup>3</sup>-C (1491 Btu/ft<sup>3</sup>-F). A typical value is 2.347·10<sup>6</sup> J/m<sup>3</sup>-C (35 Btu/ft<sup>3</sup>-F).

The unit of *rock\_cp2* is (J/m<sup>3</sup>-C-C | Btu/ft<sup>3</sup>-F-F | J/cm<sup>3</sup>-C-C) and the suggested range is ±10<sup>10</sup> J/m<sup>3</sup>-C-C (±8.28·10<sup>8</sup> Btu/ft<sup>3</sup>-F-F).

\*ROCKCP\_SHL *shale\_cp1* (*shale\_cp2*)

Heat capacity of shale when a net-to-gross option is used. See **Shale Properties** in the EXPLANATION. The descriptions of *shale\_cp1* and *shale\_cp2* are similar to those of *rock\_cp1* and *rock\_cp2*, above.

### **DEFAULTS:**

Absent	Action
*ROCKCP	<i>rock_cp1</i> = 2.347·10 <sup>6</sup> J/m <sup>3</sup> -C = 35 Btu/ft <sup>3</sup> -F
<i>rock_cp2</i>	<i>rock_cp2</i> = 0
*ROCKCP_SHL	Shale has same heat capacity as pay rock.
<i>shale_cp2</i>	<i>shale_cp2</i> = 0

### **CONDITIONS:**

These keywords must be in the Other Reservoir Properties keyword group.

### **EXPLANATION:**

The heat (internal energy) contained in a unit volume of rock is the rock heat capacity correlation integrated from reference temperature T<sub>Rabs</sub> (\*TEMR converted to absolute degrees) to temperature T<sub>abs</sub>:

$$rock\_cp1 \cdot (T_{abs} - T_{Rabs}) + rock\_cp2 \cdot (T_{abs}^2 - T_{Rabs}^2)/2$$

Note that this expression is zero when  $T_{abs} = T_{Rabs}$ , which is a consequence of the definition of \*TEMR.

When  $rock\_cp2 = 0$  the rock heat content is linear in temperature, in which case the thermal response of the rock (difference in heat content between two temperatures) becomes independent of \*TEMR. This is another way of saying that the rock heat capacity is constant.

#### **Value of *rock\_cp1* for non-zero *rock\_cp2***

In the rock heat capacity correlation, the value of *rock\_cp1* (and *shale\_cp1*) corresponds to  $T_{abs}$  in absolute degrees, that is,  $T_{abs}$  is in K° when user input temperature is C° or K°, and  $T_{abs}$  is in R° when user input temperature is F° or R°, as determined by keyword \*INUNIT.

Sometimes a two-coefficient correlation from a reference uses a non-absolute temperature unit. In this case the first coefficient must be converted. For example, consider correlation ( $cpf1 + cpf2 \cdot T_f$ ) where  $T_f$  is temperature in F° and  $cpf1$  and  $cpf2$  are in the desired units (e.g., Btu/ft<sup>3</sup>-F). Since  $T_{abs} = T_f + 459.67$  and  $rock\_cp2 = cpf2$ , and we want

$$rock\_cp1 + rock\_cp2 \cdot T_{abs} = cpf1 + cpf2 \cdot T_f$$

then the first correlation coefficient should be

$$rock\_cp1 = cpf1 - 459.67 \cdot rock\_cp2$$

The same comments apply to *shale\_cp1*.

#### **Natural Fracture Data**

For natural fracture options \*DUALPOR, etc., values of rock heat capacity and phase thermal conductivities are specified as intrinsic values for both matrix and fracture blocks. Consequently, fracture and matrix blocks may share the same rock type. When fracture does not contain rock its intrinsic void porosity will be equal to one and therefore only fluid will participate in the block heat capacity and conductivity calculation. STARS will calculate the appropriate block heat capacities and conductivities for each block depending on the current intrinsic porosity values. In fact, a fracture and corresponding matrix block need separate rock types only if (1) the fracture block contains some formation via \*FORMINFRAC and (2) the fracture and matrix blocks have different intrinsic formation properties. Different rock types are specified via \*ROCKTYPE and assigned to blocks via \*THTYPE \*MATRIX and \*THTYPE \*FRACTURE.

A fracture block containing no formation will be assigned internally a void porosity of 1 that will remain constant throughout the run. In this case, porosity parameters like \*CPOR and \*CTPOR, as well as models like \*DILATION, will not be applied to the fracture block. However, those void porosity calculations will be done for a fracture block that does contain some formation and consequently has a void porosity less than 1.

#### **Shale Properties**

The net-to-gross option (\*NETPAY or \*NETGROSS) assumes that a block has horizontal shale streaks which have zero porosity but can retain and conduct heat. Keywords \*POR, \*PERMI and \*PERMJ specify properties for the pay zone, and adjustments are made internally to account for the net-to-gross ratio (see EXPLANATION for \*NETPAY). However, rock property \*ROCKCP is not adjusted this way by default, since the shale is assumed to have the same properties as the rock in the pay zone.

Keyword \*ROCKCP\_SHL lets you specify shale heat capacity that is different from rock in the pay zone. Consider a block with pay zone void porosity  $\phi_v$ , fluid porosity  $\phi_f$  and net-to-gross ratio R. If we let subscripted U denote the per-volume energy of the various parts of the block (r – payzone rock, s – solid phase), then the value for the entire block is

$$U_{\text{block}} = (1-R)U_{\text{shale}} + R[(1-\phi_v)U_r + (\phi_v-\phi_f)U_s + \phi_f(S_wU_w+S_oU_o+S_gU_g)]$$

If keyword \*ROCKCP\_SHL is absent then  $U_{\text{shale}} = U_r$  and the block value reduces to

$$U_{\text{block}} = (1-R\phi_v)U_r + (R\phi_v-R\phi_f)U_s + R\phi_f(S_wU_w+S_oU_o+S_gU_g)$$

This is the more standard form, but with  $\phi$  replaced by “effective” porosity  $R\phi$  as stated in the \*NETPAY explanation.

Note that entering zero for \*ROCKCP\_SHL causes the shale to contribute nothing to the storage of heat in the system. This should be done only if fine-grid calibration results show that the shale in question does not contribute significantly to the thermal process.

---

## **Thermal Conductivities (Optional)** \*THCONMIX, \*THCONR, \*THCONS, \*THCONW, \*THCONO, \*THCONG, \*THCONTAB, \*THCONANTAB, \*THCONR\_SHL

### **PURPOSE:**

Specify thermal conductivities of rock, solid and fluid phases.

### **FORMAT:**

```
*THCONMIX ( *SIMPLE | *COMPLEX | *TEMPER )
*THCONR thconr
*THCONW thconw
*THCONO thcono
*THCONG thcong
*THCONS thcons
OR
*THCONTAB
{ T thconr thconw thcono thcong ( thcons ) }
OR
*THCONANTAB
{ T thconr_i thconr_j thconr_k thconw_i thconw_j thconw_k
  thcono_i thcono_j thcono_k thcong_i thcong_j thcong_k
  ( thcons_i thcons_j thcons_k ) }
*THCONR_SHL thconr
```

### **DEFINITIONS:**

#### **\*THCONMIX**

Specify the rule used to mix thermal conductivities of rock and phases. The choice of mixing rule affects somewhat the meaning of the individual rock/phase thermal conductivity values.

#### **\*SIMPLE**

The \*SIMPLE volume-weighted mixing rule for thermal conductivity is

$$\kappa_{\text{mix}} = \phi_f \cdot ( thconw \cdot S_w + thcono \cdot S_o + thcong \cdot S_g ) \\ + ( 1 - \phi_v ) \cdot thconr + ( \phi_v - \phi_f ) \cdot thcons$$

where  $\phi_v$  is the void porosity (solid plus fluids) and  $\phi_f$  is the fluid porosity (fluids only).

To specify a single constant thermal conductivity (independent of porosity, saturation and temperature), assign the same value to rock and phases via *thconr*, *thconw*, etc. A typical value for water-saturated rock is  $1.496 \cdot 10^5$  J/m-day-K (24 Btu/ft-day-F). This scenario is appropriate where thermal conduction does not contribute significantly to the recovery process, e.g., field-scale cases with modest interblock temperature gradients.

A single constant value is not appropriate where thermal conduction is significant, e.g., wellbore heat-loss and lab-scale models. In this case enter individual values representative of rock and each phase. However, a greater accuracy in rock/phase mixing may require the \*COMPLEX mixing rule.

#### \*COMPLEX

\*COMPLEX specifies mixing of rock and phase thermal conductivities that accounts better for porous media interphase contact. See appendix D.10 for a detailed description. Specify different rock and phase values via *thconr*, etc.

For example, a water-saturated rock with  $\varphi_v = \varphi_f = 0.3$ ,  $S_w = 1$ ,  $thconr = 44$  Btu/ft-day-F and  $thconw = 8.6$  Btu/ft-day-F gives 24.3 Btu/ft-day-F via the complex mixing as opposed to 33.4 Btu/ft-day-F via linear mixing.

The \*COMPLEX option takes special action for small porosities (< 1%). When porosity is zero, the rock value *thconr* alone is used. For positive values less than 1%, the formula is applied with a porosity value of 1%. This action preserves numerical stability since for porosities near zero the mixing formula results in unphysically large thermal conductivity values.

#### \*TEMPER

The \*TEMPER option specifies the \*COMPLEX mixing rule with an additional temperature correction. This temperature dependence option is considered obsolete and is effectively replaced by \*THCONTAB.

##### *thconr*

Thermal conductivity of reservoir rock (J/m-day-C | Btu/ft-day-F | J/cm-min-C). A representative value for rock is  $2.74 \cdot 10^5$  J/m-day-C (44 Btu/ft-day-F). The minimum allowed value is 0, and the maximum suggested value is  $10^7$  J/m-day-C (1605 Btu/ft-day-F). The rock heat conductivity will not be used in a fracture block that contains no rock (see \*FORMINFRAC).

##### *thconw*

Thermal conductivity of the water phase (J/m-day-C | Btu/ft-day-F | J/cm-min-C). A typical value is  $5.35 \cdot 10^4$  J/m-day-K (8.6 Btu/ft-day-F). The minimum allowed value is 0, and the maximum suggested value is  $10^7$  J/m-day-C (1605 Btu/ft-day-F).

##### *thcono*

Thermal conductivity of the oil phase (J/m-day-C | Btu/ft-day-F | J/cm-min-C). A typical value is  $1.15 \cdot 10^4$  J/m-day-K (1.8 Btu/ft-day-F). The minimum allowed value is 0, and the maximum suggested value is  $10^7$  J/m-day-C (1605 Btu/ft-day-F).

*thcong*

Thermal conductivity of the gas phase (J/m-day-C | Btu/ft-day-F | J/cm-min-C). Typical values range from 1800 to 6000 J/m-day-C (0.3 to 1.0 Btu/ft-day-F). The minimum allowed value is 0, and the maximum suggested value is  $10^7$  J/m-day-C (1605 Btu/ft-day-F). Compared to rock and water values, gas values are much lower and so precise values may not be necessary in most cases.

*thcons*

Thermal conductivity of the solid phase ( J/m-day-C | Btu/ft-day-F | J/cm-min-C). For example, solid coke is  $4.5 \cdot 10^5$  J/m-day-K (72 Btu/ft-day-F). The minimum allowed value is 0, and the maximum suggested value is  $10^{20}$  J/m-day-C ( $1.6 \cdot 10^{16}$  Btu/ft-day-F).

**\*THCONTAB**

Temperature-dependent isotropic thermal conductivities of all the phases are defined via a table. After temperature column  $T$  (C|F), columns *thconr*, etc., appear in the order shown. The column for *thcons* is optional; if *thcons* is absent then *thcons* = *thconr* is assumed. Each row must have the same number of columns. The maximum allowed number of rows is 30.

\*THCONMIX \*TEMPER may not be used with this option. If there is only 1 row, thermal conductivity does not vary with temperature.

The minimum allowed  $T$  is -100 C (-148 F).  $T$  entries must be increasing down the column and evenly spaced.  $T$  entries that are not evenly spaced will be adjusted without interpolating the other columns. All thermal conductivity entries must be non-negative, and there is no suggested upper limit.

Keyword \*THCONTAB specifies values for rock and all fluid phases, and so over-rides values specified by keywords \*THCONR, etc.

**\*THCONANTAB**

Temperature-dependent anisotropic thermal conductivities of all phases are defined via a table. After temperature column  $T$  (C|F), columns *thconr\_i*, etc., appear in the order shown. These columns are:

*thconr\_i, thconr\_j, thconr\_k*

rock (same as *thconr* above); I, J and K directions

*thconw\_i, thconw\_j, thconw\_k*

water phase (same as *thconw* above); I, J and K directions

*thcono\_i, thcono\_j, thcono\_k*

oil phase (same as *thcono* above); I, J and K directions

*thcong\_i, thcong\_j, thcong\_k*

gas phase (same as *thcong* above); I, J and K directions

*thcons\_i, thcons\_j, thcons\_k*

solid phase (same as *thcons* above); I, J and K directions

The group of columns for solid phase is optional, so there must be either 13 or 16 columns. The remaining comments from \*THCONTAB apply here as well. Use \*THCONTAB if all thermal conductivities are isotropic.

Keyword \*THCONANTAB specifies values for rock and all phases, and so over-rides values specified by keywords \*THCONR, etc.

#### \*THCONR\_SHL *thconr*

Thermal conductivity of shale when a net-to-gross option is used. See **Shale Properties** in the EXPLANATION. See the description for *thconr*.

#### DEFUALTS:

If \*THCONMIX is absent, \*SIMPLE is assumed.

If \*THCONR\_SHL is absent, shale has the same thermal conductivity as pay rock.

If \*THCONTAB is used and *thcons* is absent then *thcons* = *thconr* is assumed.

If \*THCONANTAB is used and the solid columns are absent then *thcons\_m*= *thconr\_m* is assumed in each direction  $m = i,j,k$ .

If keywords \*THCONTAB and \*THCONANTAB are absent, the following defaults apply:

Absent	Action
*THCONR	<i>thconr</i> = $1.496 \cdot 10^5$ J/m-day-K = 24 Btu/ft-day-F
*THCONW	<i>thconw</i> "
*THCONO	<i>thcono</i> "
*THCONG	<i>thcong</i> "
*THCONS	<i>thcons</i> = <i>thconr</i>

#### CONDITIONS:

These keywords must be in the Other Reservoir Properties keyword group.

No more than one of \*THCONTAB, \*THCONANTAB and \*THCONMIX \*TEMPER may be used together for each rock type.

\*THCONR\_SHL is available only with the anisotropic option \*THCONANTAB.

For each discretized wellbore, a non-zero wall thermal conductivity may be specified using either (1) \*WELLWALL or (2) *thconr* with \*ROCKTYPE and \*THTYPE \*WELLBORE, but not both.

#### EXPLANATION:

##### Organization of Dependencies

Thermal conductivity has three possible types of dependency: rock/phase, temperature and direction. With regard to data entry there is some interaction between these dependencies.

1. Rock and phase dependence is specified through the quantities *thconr*, *thconw*, *thcono*, *thcong* and *thcons*, all of which may appear in either single-keyword context (e.g., \*THCONR) or in the temperature-dependent table keywords \*THCONTAB or \*THCONANTAB.

2. Temperature dependence is specified via keywords \*THCONTAB or \*THCONANTAB, for each of the rock and phase values separately.
3. Direction dependence is specified via keyword \*THCONANTAB for each temperature and for each of the rock and phase values separately.

### **Temperature Dependence**

If the T-dependent table option is used, the conductivities of all the phases are evaluated at the current temperature. If temperature is outside the table range, the nearest table entry is used. If the table is not used, the conductivity values are constant. After the rock/phase conductivities at the current temperature are known, the rock/phase mixing rule is applied.

### **Interblock Weighting**

Conductive heat flow between two adjacent blocks is based on harmonic (series) weighting of the mixed thermal conductivities  $\kappa_{\text{mix}}$  of the two blocks in that direction. This is consistent for all cases of block size, flow directions and conductivity values. For example, if  $\kappa_{\text{mix}} = 0$  in one block then the conductive heat flow between the two blocks is zero.

### **Anisotropic Conductivities**

If keyword \*THCONANTAB is used then thermal conductivities may be different for each of the three directions. The following points apply.

1. Use of single subkeyword \*THCONDUCT of \*OUTSRF \*GRID will generate output values for all three directions, including special histories.
2. For oriented sub-grids the I, J and K directions correspond as described in “Direction Dependent Data” in the explanation for keyword \*REFINE.
3. Full anisotropy is not available for the advanced nine-point option \*NINEPTH; the conductivities in the nine-point plane are taken from direction I.
4. Full anisotropy is not available for heat conduction to and from discretized and semi-analytical wellbores; the conductivities are taken from direction I.

### **Natural Fracture Data**

See **Natural Fracture Data** in the EXPLANATION for keyword \*ROCKCP.

### **Shale Properties**

The net-to-gross option (\*NETPAY or \*NETGROSS) assumes that a block has horizontal shale streaks which have zero porosity but can retain and conduct heat. Keywords \*POR, \*PERMI and \*PERMJ specify properties for the pay zone, and adjustments are made internally to account for the net-to-gross ratio (see EXPLANATION for \*NETPAY). However, rock property \*THCONR is not adjusted this way by default, since the shale is assumed to have the same property as the rock in the pay zone.

Keyword \*THCONR\_SHL lets you specify shale thermal conductivity, in the I and J directions, different from rock in the pay zone. If subscripted  $\kappa$  is the conductivity of the various parts of the block, then the value for the entire block is

$$\kappa_{\text{block}} = (1-R)\kappa_{\text{shale}} + R[(1-\varphi_v)\kappa_r + (\varphi_v - \varphi_f)\kappa_s + \varphi_f(S_w\kappa_w + S_o\kappa_o + S_g\kappa_g)] \quad *{\text{SIMPLE}}$$

$$\kappa_{\text{block}} = (1-R)\kappa_{\text{shale}} + RF(\varphi_v, \varphi_f, S_w, S_o, S_g, \kappa_r, \kappa_s, \kappa_w, \kappa_o, \kappa_g) \quad *{\text{COMPLEX}}$$

If keyword \*THCONR\_SHL is absent, and for K direction, the block value is

$$\kappa_{\text{block}} = (1 - R\varphi_v)\kappa_r + (R\varphi_v - R\varphi_f)\kappa_s + R\varphi_f(S_w\kappa_w + S_o\kappa_o + S_g\kappa_g) \quad *{\text{SIMPLE}}$$

$$\kappa_{\text{block}} = F(R\varphi_v, R\varphi_f, S_w, S_o, S_g, \kappa_r, \kappa_s, \kappa_w, \kappa_o, \kappa_g) \quad *{\text{COMPLEX}}$$

This is the more standard form, but with  $\varphi$  replaced by  $R\varphi$ . Unlike the \*SIMPLE case, the \*COMPLEX expressions are not equivalent when  $\kappa_{\text{shale}} = \kappa_r$ .

Note that entering zero for \*THCONR\_SHL causes the shale to contribute nothing to the conduction of heat in the system. This should be done only if fine-grid calibration results show that the shale in question does not contribute significantly to the thermal process.

See **Shale Properties** in the EXPLANATION for keyword \*ROCKCP.

---

## **SUBDOMAIN-DK Heat Conduction Multiplier (Optional)**

**\*SD\_HCMULT**

### **PURPOSE:**

Specify inter-subdomain multiplier for heat conduction.

### **ARRAY:**

**\*SD\_HCMULT**

### **DEFAULTS:**

If **\*SD\_HCMULT** is absent, all inter-subdomain heat conduction multipliers are assumed to be one.

### **CONDITIONS:**

Keyword **\*SD\_HCMULT** is meaningful and effective only when used together with keywords **\*SUBDOMAIN** and **\*TRANSD**.

Array qualifiers **\*MATRIX** and **\*FRACTURE** are not allowed.

### **EXPLANATION:**

Keyword **\*SD\_HCMULT** allows you to apply a multiplier to the conductive heat flow of the connection created for the **SUBDOMAIN-DK** model by keyword **\*TRANSD**, to control the contact effectiveness of that new connection.

Values entered for **\*SD\_HCMULT** are dimensionless and must be non-negative. A value of one results in heat conduction similar to a matrix-matrix connection within a local matrix stack. A value of zero prevents conductive heat flow across that connection. The value of **\*SD\_HCMULT** does not affect fluid flow across that connection; use **\*TRANSD** for fluid flow adjustment.

A **\*SD\_HCMULT** value entered for block (i,j,k) will be applied to the **SUBDOMAIN-DK** connection between the matrix subblock stacks in blocks (i,j,k) and (i,j,k+1).

Note that a zero value of **\*TRANSD** will remove the connection completely so prevent heat conduction as well. If you want heat conduction but no fluid flow in that connection, use a small but non-zero value for **\*TRANSD**.

### **Examples:**

The following are two examples of **\*SD\_HCMULT** usage:

```
** Enhance heat conduction between matrix blocks across subdomains
** for all grids
*SD_HCMULT *CON 1.5

** Cut off heat conduction between matrix blocks across subdomains
** between grid 2,3,6 and 2,3,7 only
*SD_HCMULT *IJK 2:2 3:3 6:6 0.0
```

The suggested range of values for heat conduction multipliers is:

	<b>SI</b>	<b>Field</b>	<b>Lab</b>
min	0.0	0.0	0.0
max	1000.0	1000.0	1000.0

## Overburden Heat Loss (Optional)

\*HLOSSPROP, \*HOSST,

\*HOSSTDIFF

### PURPOSE:

\*HLOSSPROP defines the heatloss directions and over/underburden thermal properties for the semi-analytical infinite-overburden heat loss model.

\*HOSST and \*HOSSTDIFF control the overburden temperature and critical temperature difference.

### FORMAT:

\*HLOSSPROP ( \*OVERBUR | \*UNDERBUR | \*-I | \*+I  
| \*-J | \*+J | \*-K | \*+K ) *dnurol hconl*

\*HOSST *thf*

\*HOSSTDIFF *dthl*

### DEFINITIONS:

#### \*OVERBUR

Apply these heat loss properties to the outer grid block faces at the reservoir top, equivalent to \*+K for \*KDIR \*UP and \*-K for \*KDIR \*DOWN.

#### \*UNDERBUR

Apply these heat loss properties to the outer grid block faces at the reservoir bottom, equivalent to \*+K for \*KDIR \*DOWN and \*-K for \*KDIR \*UP.

#### \*+I, \*-I, \*+J, \*-J, \*+K, \*-K

Apply these heat loss properties to the outer grid block faces in the indicated direction. \*OVERBUR/\*UNDERBUR may be used together with \*+I, \*-I, \*+J and \*-J but should not be used with \*+K/\*-K.

#### *dnurol*

Volumetric heat capacity of formation adjacent to the reservoir in the indicated direction ( $\text{J/m}^3\text{-C}$  |  $\text{Btu/ft}^3\text{-F}$ ). The lower limit is 0, and the suggested upper limit is  $10^8 \text{ J/m}^3\text{-C}$  ( $1491 \text{ Btu/ft}^3\text{-F}$ ). A value of zero will result in no heat loss. A typical value for wet rock is  $2.347 \cdot 10^6 \text{ J/m}^3\text{-K}$  ( $35 \text{ Btu/ft}^3\text{-F}$ ).

#### *hconl*

Thermal conductivity of formation adjacent to the reservoir in the indicated direction ( $\text{J/m-day-C}$  |  $\text{Btu/ft-day-F}$ ). The lower limit is 0, and the suggested upper limit is  $10^7 \text{ J/m-day-C}$  ( $1605 \text{ Btu/ft-day-F}$ ). A zero value results in no heat loss. A typical value for wet rock is  $1.496 \cdot 10^5 \text{ J/m-day-K}$  ( $24 \text{ Btu/ft-day-F}$ ).

#### *thf*

Initial temperature of formation adjacent to the reservoir, used by the heat loss calculation (deg C | deg F). The value must be non-negative.

*dthl*

Minimum temperature difference needed between block temperature and *thf* to start heat loss calculation (C deg | F deg). The lower limit is 0, and the suggested upper limit is 10 C deg (18 F deg).

### **DEFAULTS:**

If \*HLOSSPROP is absent, there is no heat loss.

If \*HLOSSST is absent, *thf* for each external block face is equal to the initial temperature in the corresponding reservoir block.

If \*HLOSSSTDIF is absent, *dthl* = 0.1 C.

### **EXPLANATION:**

Reference: "A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators", Vinsome, P.K.W. & Westerveld, J.D., JCPT, July-September 1980, Volume 19, No. 3.

Keyword \*OUTPRN \*GRID \*OBHLOSS causes the heat loss rate for each grid block to be dumped to the ".out" file. After that, a summary of heat loss rate and accumulation is written, split between overburden and underburden. Here, "overburden" is the total heat loss for all the blocks with heat loss in the direction opposite gravity; "underburden" is the total heat loss for all the blocks with heat loss in the gravity direction. When heat loss is specified only in those two layers (e.g. using \*OVERBUR and \*UNDERBUR) and those layers are distinct, then the overburden and underburden heat loss will sum to the total shown in the material balance summary.

See Appendix D.11 for further discussion.

### **Sign Convention for Reporting Heat Loss**

Reported heat loss values are negative for heat transfer from reservoir to overburden, and positive from overburden to reservoir.

### **Heat Loss Through Null Blocks**

Overburden heat loss consists of heat flow between a boundary grid block and a semi-infinite portion of formation adjacent to that block, as specified by \*HLOSSPROP directions and possible \*ROCKTYPE and \*THTYPE. Here, "boundary block" in a given direction is an active block that has no other active block between itself and the edge of the grid in that direction. Therefore, for heat loss purposes the boundary of the reservoir is deemed to be the side of such a "boundary block" in that direction. For example, if an I-J grid column has blocks K = 1 & 2 nulled out, heat loss in the -K direction will occur from the -K side of block K = 3.

Blocks can be nulled out via keywords \*NULL, VAMOD, \*PINCHOUTARRAY or \*PINCHOUT-TOL. Note that "direction" and "between" in the above description depend only on I-J-K indices, not X-Y-Z positions in 3D space; the distinction can be significant for corner-point grids. Heat loss is available from a block defined with \*REFINE, as long as there is no active block between it and the grid edge, on any grid refinement level. You can prevent heat loss through null blocks in a direction by not assigning heat loss properties in that direction (e.g., triangular element of areal symmetry in +/-I and +/-J directions).

## **Diagonal Transmissibility Multipliers (Optional)**

**\*TRANSIJ+,**

**\*TRANSIJ-, \*TRANSIK+, \*TRANSIK-**

### **PURPOSE:**

Specify transmissibility multipliers in the diagonal grid directions.

### **ARRAY:**

**\*TRANSIJ+**  
**\*TRANSIJ-**  
**\*TRANSIK+**  
**\*TRANSIK-**

### **DEFAULTS:**

If the keyword does not appear, the corresponding multiplier is assumed to be one. If a keyword appears but a block is not referenced, its corresponding multiplier is one.

### **CONDITIONS:**

Keywords **\*TRANSIJ+**, **\*TRANSIJ-**, **\*TRANSIK+** and **\*TRANSIK-** must be located in the Other Reservoir Properties section (after **\*END-GRID**) or in recurrent data.

**\*TRANSIJ+** and **\*TRANSIJ-** are applicable only with **\*NINEPOINT \*IJ**, and indicate the diagonal directions  $I+J+$  and  $I+J-$ , respectively. Similarly, **\*TRANSIK+** and **\*TRANSIK-** are applicable only for **\*NINEPOINT \*IK** to indicate directions  $I+K+$  and  $I+K-$ , respectively.

### **EXPLANATION:**

See the explanation for **\*TRANSI** for general comments on transmissibility multipliers.

Transmissibility multipliers can be changed at any time in recurrent data.

Diagonal transmissibility multipliers are used only with the nine-point option, to modify the flow for diagonal interblock connections as follows.

<u>This keyword</u>	<u>Refers to this direction</u>	<u>Between This Cell</u>	<u>And This Cell</u>	<u>For *NINEPOINT</u>
<b>*TRANSIJ+</b>	$I+J+$	( $i,j,k$ )	( $i+1,j+1,k$ )	<b>*IJ</b>
<b>*TRANSIJ-</b>	$I+J-$	( $i,j,k$ )	( $i+1,j-1,k$ )	<b>*IJ</b>
<b>*TRANSIK+</b>	$I+K+$	( $i,j,k$ )	( $i+1,j,k+1$ )	<b>*IK</b>
<b>*TRANSIK-</b>	$I+K-$	( $i,j,k$ )	( $i+1,j,k-1$ )	<b>*IK</b>

An interblock connection in a diagonal direction has a transmissibility calculation that is similar to a normal five-point method, but the effective cross-sectional area and internode separation are particular to the nine-point formulation. However, a diagonal connection's multiplier is applied to the corresponding transmissibility in exactly the same way as for normal five-point methods. The key is that this multiplier is applied to the default transmissibility given by the method in use (five-point or nine-point).

## Flow Barrier in Nine-Point Plane

You can use transmissibility multipliers to specify a fluid-flow barrier normal to the nine-point plane, but you must take care with addressing cells for diagonal directions.

For example, suppose that you have a 10x10x5 grid with standard five-point discretization. To specify a fluid-flow barrier between grid planes K=3 and K=4 (over the entire I-J area) you would need only \*TRANSK in the Reservoir Description data section:

```
*TRANSK *IJK 1:10 1:10 3 0.0
```

Note that the multiplier is assigned using cell address K=3 since \*TRANSK refers to the connection between K and K+1.

Now suppose that we apply \*NINEPOINT \*IK to the same grid, that is, nine-point discretization is done in the I-K plane. To implement a fluid-flow barrier between planes K=3 and K=4 you would need additional data:

```
*TRANSK *IJK 1:10 1:10 3 0.0  
*END-GRID  
*TRANSIK+ *IJK 1:10 1:10 3 0.0  
*TRANSIK- *IJK 1:10 1:10 4 0.0
```

Note that the \*TRANSIK+ multiplier is assigned using cell address K=3 since \*TRANSIK+ refers to the connection between K and K+1. On the other hand, the \*TRANSIK- multiplier is assigned using cell address K=4 since \*TRANSIK- refers to the connection between K and K-1.

Similar comments apply to \*TRANSIJ+ and \*TRANSIJ- when constructing a fluid-flow boundary between two J planes when \*NINEPOINT \*IJ is used.

---

## **Electrical Heating Sets (Optional)**

\*ELECHEAT, \*ELECTYPE, \*VOLTOL,  
\*VOLSHF, \*EHEATCYC, \*ELTYPE

### **PURPOSE:**

Enable electrical heating and specify electrical property sets.

### **FORMAT:**

```
*ELECHEAT  
*ELECTYPE set_number ( *COPY old_set_number )  
*VOLTOL Vtol  
*VOLSHF Vshift  
*EHEATCYC ncyc
```

### **ARRAY:**

```
*ELTYPE
```

### **DEFINITIONS:**

#### **\*ELECHEAT**

Enable electrical heating option. See Appendix G.

#### **\*ELECTYPE *set\_number* ( \*COPY *old\_set\_number* )**

Define electrical heating property set *set\_number*, starting at 1 and increasing by 1 for each set.

The following keywords may be specified on a per-set basis: \*ELCONTAB, \*ELWCOMPTAB, \*ELSCOMPTAB, \*ALITHO, \*ATORTU, \*ASATUR and \*TEMMULT.

Use optional \*COPY to load all the property values from previously defined set *old\_set\_number* as a basis for the current set.

#### **\*ELTYPE**

Grid array that assigns set electrical heating property set numbers to grid blocks.

#### **\*VOLTOL *V<sub>tol</sub>***

Specify convergence tolerance for electric potential (volts). Due to the nature of the electrical current equation (no accumulation term), voltage usually converges in one iteration. Therefore, decreasing *V<sub>tol</sub>* rarely has any effect.

#### **\*VOLSHF *V<sub>shift</sub>***

Specify the numerical shift for electric potential (volts). *V<sub>shift</sub>* should be around  $10^{-4}$  of *V<sub>tol</sub>*.

#### **\*EHEATCYC *n<sub>cyc</sub>***

Specify the last Newton cycle number *n<sub>cyc</sub>* at which the heat rate is updated. A value of 1 corresponds to an explicit mode, which is most stable. Values greater than 1 may give more accuracy at the cost of more cycles.

\*CURRENT type boundary constraints may not converge satisfactorily with larger values of \*EHEATCYC.

## DEFAULTS:

Any block that is not explicitly assigned a set number via \*ELTYPE will use set #1.

If \*VOLTOL is absent then  $V_{tol} = 10$  V is assumed.

If \*VOLSHF is absent then  $V_{shift} = 10^{-3}$  V is assumed.

If \*EHEATCYC is absent then  $n_{cyc} = 1$  is assumed.

## CONDITIONS:

### EXPLANATION:

The electric heating option is enabled by keyword \*ELECHEAT in this section. See Appendix G. The other electrical property keywords in this section must follow as a group immediately after \*ELECHEAT. None of the electrical heating keywords in this section is mandatory, but electrical conductivity must be assigned for at least one phase or rock before any electrical potential field can be calculated.

### Keyword Overview

The electrical heating keywords are organized into the following groupings, one grouping per manual page, in the following data sections.

#### Other Reservoir Properties

1. \*ELECHEAT enables the electrical heating option. \*ELECTYPE and \*ELTYPE access the property set option. \*VOLTOL, \*VOLSHF and \*EHEATCYC control convergence.
2. \*ELCONTAB, \*ELWCOMPTAB and \*ELSCOMPTAB specify electrical conductivities that vary with set, temperature, phase and composition.
3. \*ECONDWI, \*ECONDWJ, \*ECONDWK and \*TEMMULT specify electrical conductivities that vary with block and temperature.

#### Well and Recurrent Data

4. \*ELBOUND and \*ELTARGET specify electrical boundary conditions and operating constraints.

#### Input/Output Control

5. \*OUTPRN \*GRID subkeywords ELCONDUCT, etc., specify grid dump output to the ".out" text file.
6. \*OUTSRF \*GRID subkeywords ELCONDUCT, etc., specify grid dump output to the SR2 graphics file. In addition, \*OUTSRF \*SPECIAL subkeywords ELHEAT, etc., are available for history plots.
7. The EXPLANATION for keyword \*INUNIT documents the electrical units.

## Restrictions

The electric heating option may be used with any grid, component, rock property and fluid well configuration, with the following exceptions:

1. The nine-point, natural fracture and discretized wellbore grid options are not allowed.
2. You may not use the \*RW 0 option or keyword \*GRID \*RADIAL with an electrical boundary in the -I, -J or +J direction, since this would give a radius of 0 to the inner reservoir boundary normally associated with the wellbore.
3. A zero-porosity heat-conducting block conducts electrical current only if rock electrical conductivity is assigned a non-zero value.
4. The \*ISOTHERMAL formulation option is not allowed.
5. Adaptive-implicit (\*AIM) options are not recommended or supported.
6. Dynamic (\*DYNAGRID) and recurrent gridding options are not allowed.

## Electrical Property Sets

Each electrical property set (rock type) can have its own values of conductivities, temperature multipliers and Archie parameters. For example, they are useful for assigning different conductivity data to reservoir and overburden, or to different reservoir formation types.

When \*COPY is used and the old set contains \*ELCONTAB, the presence of \*ELCONTAB in the new set will clear the old conductivity data from the new set before new data is read. However, the temperature entries from the old set will be kept. The same comments apply to \*ELWCOMPTAB and \*ELSCOMPTAB.

To model current flow in non-porous materials like metal, use a separate set for the material, set porosity to zero and use \*ROCKI, etc. Thermal and electrical flow calculations are valid in such blocks. For example, material with a high conductivity will have low resistance and hence low energy generation.

## Examples

The following example shows that subkeyword \*COPY lets you see immediately the relationship between rock types (rock type #2 will have the same Archie parameters as #1, but different temperature multipliers).

```
*ELECTYPE 1          ** Pay zone
  *TEMMLT
    65.      1.
    150.     2.8
    350.     3.2
  *ALITHO .92
  *ATORTU 1.43
*ELECTYPE 2  *COPY 1      ** Overburden
  *TEMMLT
    65.      1.
    250.     1.8
*ELTYPE  *KVAR 5*2 10*1      ** 5 overburden layers, 10 pay
                                zone layers
```

---

## **Electrical Heating Properties (Optional)**

**\*ELCONTAB,**

**\*ELWCOMPTAB, \*ELSCOMPTAB, \*ALITHO, \*ATORTU, \*ASATUR**

### **PURPOSE:**

Specify properties for each electrical heating set.

### **FORMAT:**

```
*ELCONTAB key1 ... keyn
  { T val1 ... valn }

*ELWCOMPTAB 'comp_name1' ... 'comp_namen'
  { T I-val1 J-val1 K-val1 ... I-valn J-valn K-valn }

*ELSCOMPTAB 'comp_name1' ... 'comp_namen'
  { T I-val1 J-val1 K-val1 ... I-valn J-valn K-valn }

*ALITHO Alith
*ATORTU Ator
*ASATUR Asat
```

### **DEFINITIONS:**

#### **\*ELCONTAB**

Specify electrical conductivities of water, rock, solids and oil as a function of temperature by a table where  $T$  is temperature (C|F), and  $val_i$  is electrical conductivity (siemens/metre) corresponding to keyword  $key_i$ . The keyword list may be any number of the following.

*WATERI	water phase, I direction
*WATERJ	water phase, J direction
*WATERK	water phase, K direction
*ROCKI	rock/matrix, I direction
*ROCKJ	rock/matrix, J direction
*ROCKK	rock/matrix, K direction
*SOLIDI	solid phase, I direction
*SOLIDJ	solid phase, J direction
*SOLIDK	solid phase, K direction
*OILI	oil phase, I direction
*OILJ	oil phase, J direction
*OILK	oil phase, K direction

For each I-J-K triplet of keywords, the J and K direction data defaults to the I direction data if the corresponding I direction keyword is present. If all three keywords in a triplet are absent, there is no current flow in that phase or rock. Temperature entries must increase and the number of rows must not exceed 20. Electrical conductivity values must be non-negative. A phase or rock with zero conductivity for all directions and temperatures will be treated like no values were entered, on a per-set basis.

To model current flow in non-porous materials like metal, use a separate rock type for the material, set porosity to zero and use \*ROCKI, etc.

#### \*ELWCOMPTAB

Specify water phase electrical conductivity as a function of composition via table where '*comp\_name<sub>i</sub>*' is the quoted name of a water (aqueous) component defined by keyword \*MODEL and \*COMPNAME, *T* is temperature (C|F), and *I-val<sub>i</sub>*, *J-val<sub>i</sub>* and *K-val<sub>i</sub>* are electrical conductivity (siemens/metre) in the three grid directions, one triplet for each component. Temperature entries must increase and the number of rows must not exceed 20. Electrical conductivity values must be non-negative.

If this table method is used together with \*ELCONTAB then (1) the temperature entries of the two tables must be the same and (2) this table overrides any water phase values entered for \*ELCONTAB.

#### \*ELSCOMPTAB

Specify solid phase electrical conductivity as a function of composition via table where '*comp\_name<sub>i</sub>*' is the quoted name of a solid component defined by keyword \*MODEL and \*COMPNAME, *T* is temperature (C|F), and *I-val<sub>i</sub>*, *J-val<sub>i</sub>* and *K-val<sub>i</sub>* are electrical conductivity (siemens/metre) in the three grid directions, one triplet for each component. Temperature entries must increase and the number of rows must not exceed 20. Electrical conductivity values must be non-negative.

If this table method is used together with \*ELCONTAB then (1) the temperature entries of the two tables must be the same and (2) this table overrides any solid phase values entered for \*ELCONTAB.

#### \*ALITHO *Alith*

Specify Archie lithology parameter *Alith*. The allowed range is  $10^{-5}$  to  $10^{10}$ .

#### \*ATORTU *Ator*

Specify Archie tortuosity parameter *Alith*. *Ator* must be non-negative.

#### \*ASATUR *Asat*

Specify Archie water saturation parameter *Asat*. *Asat* must be non-negative.

#### **DEFAULTS:**

If *Alith* is not assigned a value, *Alith* = 0.88 is assumed.

If *Ator* is not assigned a value, *Ator* = 1.37 is assumed.

If *Asat* is not assigned a value, *Asat* = 2 is assumed.

\*ALITHO, \*ATORTU and \*ASATUR are required only to override their respective defaults, for each rock type.

## **CONDITIONS:**

At most one \*ELCONTAB table may be specified for each electrical set.  
At most one \*ELWCOMPTAB table may be specified for each electrical set.  
At most one \*ELSCOMPTAB table may be specified for each electrical rock type.  
The water phase subkeywords of \*ELCONTAB may not be used together with per-block keywords \*ECONDWI, etc.  
\*ELWCOMPTAB may not be used together with per-block keywords \*ECONDWI, etc.  
Data entered via keywords \*ELWCOMPTAB and \*ELSCOMPTAB will override water and solid phase data, respectively, entered via \*ELCONTAB.

## **EXPLANATION:**

### **Composition-Dependent Water Phase Conductivities**

If \*ELWCOMPTAB is used then the aqueous phase value will be equal to the mole weighted average of the values of all aqueous components. Only aqueous components that are assigned conductivities in \*ELWCOMPTAB will contribute to the water phase value. This table should be used only if the process depends significantly on the differences between conductivity values of the water phase components. If there is only one component in the water phase, or all water phase components have nearly the same value, then use \*WATERI, etc. with \*ELCONTAB instead.

### **Composition-Dependent Solid Phase Conductivities**

If \*ELSCOMPTAB is used then the solid phase value will be equal to the mole-weighted average of the values of all solid components. Only components that are assigned conductivities in \*ELSCOMPTAB will contribute to the solid phase value. This table should be used only if the process depends significantly on the differences between conductivity values of the solid components. If there is only one solid component, or all solid components have nearly the same value, then use \*SOLIDI, etc. with \*ELCONTAB instead.

### **Archie Parameters**

Bulk electrical conductivity can be calculated from water conductivity, porosity and S<sub>w</sub> in each of the three directions using

$$\sigma_p(T, S_w) = \sigma_{w,p}(T) * [\varphi_f^{\text{Atr}} S_w^{\text{Asat}} / \text{Alith}] \quad p = i, j, k$$

Note that overburden can conduct electrical current since it has non-zero porosity and water saturation, even though the fluid permeability is very low or zero. Alternatively, overburden can be modelled with “rock” conductivity.

### **Examples**

Non-isotropic water and rock conductivities can be specified as follows. Here the J direction values are equal to the I direction values since the J direction keywords are absent.

```
*ELCONTAB *WATERI *WATERK *ROCKI *ROCKK
      50      0.8      0.4      0.1      0.02
     150      1.6      0.8      0.2      0.04
     250      2.4      1.2      0.3      0.06
```

## Water Phase Electrical Conductivity (Optional)

\*ECONDWI,

\*ECONDWJ, \*ECONDWK, \*TEMMULT

### PURPOSE:

Specify per-block water-phase electrical conductivities.

### ARRAY:

\*ECONDWI  
\*ECONDWJ  
\*ECONDWK

### FORMAT:

\*TEMMULT  
 $\{ T \ F_T \}$

### DEFINITIONS:

\*ECONDWI, \*ECONDWJ, \*ECONDWK

Specify water-phase electrical conductivities for each block separately (siemens per metre). The \*EQUALSI keyword may be used, similar to permeability assignment via \*PERMI, etc.

These keywords are an alternative to specifying water phase conductivities via keyword \*ELCONTAB, but they are considered obsolete and the other conductivity keywords are recommended instead.

\*TEMMULT

Specify temperature dependence by a multiplier function in table form, where  $T$  is temperature (C|F) and  $F_T$  is the electrical conductivity multiplier at that temperature. Temperature entries  $T$  must be increasing. The allowed range for  $F_T$  is  $10^{-10}$  to  $10^{10}$ . The number of rows must not exceed 20.

### DEFAULTS:

Any block that is not explicitly assigned a value will use zero.

If \*ECONDWJ (or \*ECONDWK) is not entered, the J (or K) direction values are assumed to be equal to the I direction values.

The default multiplier is 1 at all temperatures. This keyword is required only to override its default, for property set separately.

### CONDITIONS:

These keywords may not be used together with the water phase subkeywords of \*ELCONTAB.

\*TEMMULT may be used only together with \*ECONDWI, etc.

## **EXPLANATION:**

### **Examples**

*Note: These keywords are considered obsolete, having been replaced with the temperature-dependent tables \*ELCONTAB, etc.*

```
*ECONDWI  *CON  0.8
*ECONDWJ  *EQUALSI
*ECONDWK  *EQUALSI * 0.3  ** K direction is reduced
```

### **Temperature Dependence**

Whenever a value of water conductivity is required for block i, it is multiplied by table look-up function  $F_T(T_i)$ . Therefore, the water conductivity data entered via \*ECONDWI, etc., are referenced to  $T_{ref}$  at which  $F_T(T_{ref}) = 1$ .



# Component Properties

## Component Types and Names (Required)

\*MODEL, \*COMPNAME

### PURPOSE:

Indicate number of each type of component in preparation for fluid data input.

### FORMAT:

```
*MODEL ncomp numy numx (numw)  
COMPNAME 'namec' (1) ... 'namec' (ncomp)
```

### DEFINITIONS:

ncomp

Total number of components in the simulation. The minimum allowed value is numy. You will be required to enter 'ncomp' component names, and 'ncomp' reaction stoichiometric coefficients.

numy

Total number of components in the oil, water or gas (fluid) phases. The allowed range is numx to ncomp, inclusive. You will be required to enter numy values for component gas phase properties, such as heat capacity.

numx

Total number of components in the water or oil (liquid) phases. These components may exist also in the gas phase, and as adsorbed species. The allowed range is numw to numy. You will be required to enter numx values for component liquid phase data, such as density.

numw

Number of water-like or aqueous components. The allowed range normally is 1 to numx. If you use one of the Z formulations (\*TFORM \*ZH or \*ZT in the Numerical Methods section) then the allowed range for numw is 0 to numx.

namec

Component name character string. Only the first eight (8) characters are used. Component names must be unique.

## **DEFAULTS:**

If numw is absent, it is assumed to be 1.

If \*COMPNAME is absent, the component names will be 'COMP\_x', where x is the component number.

## **CONDITIONS:**

\*MODEL is always required.

There are no strict limits for the number of each kind of component. However, the computational and storage requirements increase significantly with the total number of fluid components (numy).

## **EXPLANATION:**

A detailed discussion of component design concepts can be found in Appendices D.1 and D.2. A discussion of many advanced recovery processes, including how those processes are implemented with components, can be found in Appendix C as well as Appendices D.7, D.8 and D.13 to D.16.

The structure of component ordering is shown in Table 1.

### **Aqueous or Water-Like Components**

These are components 1 to numw. For these components the water phase is the reference liquid phase for K value definition:

$$K(\text{gas/liq}) = (\text{gas mole fraction}) / (\text{water mole fraction}) = y/w,$$
$$K(\text{liq/liq}) = (\text{oil mole fraction}) / (\text{water mole fraction}) = x/w.$$

For example, a component which will partition only sparingly in the oil phase should be this type, so that the liquid-liquid K values will be very small instead of very large. A condensable component which is not at all soluble in the oil phase must be aqueous, and its liquid-liquid K value must be zero.

Any injected water component using the \*QUAL option must be component #1. The normal case with one water component automatically satisfies this restriction. However, when there are multiple water components (e.g., injected versus in-place) then the injected water component must be #1 if the \*QUAL option is used. See **CWE Water Component** in the explanation for keyword \*QUAL in the Well and Recurrent Data section.

Use keyword \*ICE in this data section to enable modelling of water ice.

### **Oleic or Oil-Like Components**

These are components numw+1 to numx, and the oil phase is the reference liquid phase for K value definition:

$$K(\text{gas/liq}) = (\text{gas mole fraction}) / (\text{oil mole fraction}) = y/x,$$
$$K(\text{liq/liq}) = (\text{water mole fraction}) / (\text{oil mole fraction}) = w/x.$$

For example, a component which will partition only sparingly in the water phase should be this type, so that the liquid-liquid K values will be very small instead of very large. A condensable component which is not at all soluble in the water phase must be oleic, and its liquid-liquid (w/x) K value must be zero.

In this group must be included volatile components which are significantly soluble in liquid, such as methane and CO<sub>2</sub>.

### Noncondensable Gases

These are components numx+1 to numy, and appear only in the gas phase. The physical interpretation is that these components are so volatile that their condensation and solubility in liquids can be ignored. No liquid phase properties are required, and no K values are needed. These components are allowed to mix freely with the vapours of liquid components.

Use of 1 noncondensable gas component, that is, numy = numx + 1, may result in poor numerical performance, especially for the combustion process. If stability is poor, try numy = numx (even if the K value of component numy is very large and the solubility is slight) or numy > numx + 1.

### Solid or Trapped Components

These are components numy+1 to ncomp, and appear only in the solid or immobile phase state. These components require only basic data such as density and heat capacity. Examples of such components are:

- a) coke fuel created by cracking reaction,
- b) a component in the adsorbed or trapped state due to non-equilibrium mass transfer,
- c) rock that will dissolve, such as carbonate.

See Appendix F.8 for a discussion of solid component treatment. If there is at least one solid component then there must be at least one reaction, otherwise that component's moles will not be conserved.

### Default Partitioning of Components

Default phase partitioning will be done for components for which (a) no K value table is entered, or (b) the correlation coefficient is absent or zero. The default partitioning is

1. Aqueous components (numbers 1 to numw) are based in the water phase, and will vapourize according to the internal steam table. There is no partitioning in the oil phase.  
This is desired for the normal case of one water component, as well as the case of multiple labeled water components. This default is NOT appropriate for non-water components that partition principally in the aqueous phase, such as polymer. The individual manual entries show how to get zero K values for these components.
2. Oleic components (numbers numw+1 to numx) are based in the oil phase, and will not vapourize. There is no partitioning into the aqueous phase.  
This is desired for a dead oil component. Any volatility must be specified explicitly via correlation or table.
3. Non-condensable components (numbers numx+1 to numy) are found only in the gas phase, by definition.

<b>Component</b>	<b>Comp #</b>	<b>Aqueous</b>	<b>Oleic</b>	<b>Gaseous</b>	<b>Solid</b>
Aqueous	1	X		X	
"	...	X		X	
"	numw	X		X	
Oleic	numw+1		X		
"	...		X		
"	numx		X		
Non-condensable	numx+1			X	
"	...			X	
"	numy			X	
Solid	numy+1				X
"	...				X
"	ncomp				X

#### Default Phase Partitioning

Example:

Reservoir fluid contains water, heavy oil, light oil and methane components. Surfactant soluble only in water phase and nitrogen are injected. Nitrogen is considered non-condensable, i.e., it stays in gas phase only.

numw	=	2	Water and surfactant are considered as water-like components
numx	=	5	Additional three oil like components (heavy, light oil and methane)
numy	=	6	Add nitrogen as a non-condensable gas
ncomp	=	6	There is no solid component present

## K Value Correlations

\*KV1, \*KV2, \*KV3, \*KV4, \*KV5

### PURPOSE:

Specify gas-liquid K value correlations.

### FORMAT:

```
*KV1 kv1(1) ... kv1(numx)
*KV2 kv2(1) ... kv2(numx)
*KV3 kv3(1) ... kv3(numx)
*KV4 kv4(1) ... kv4(numx)
*KV5 kv5(1) ... kv5(numx)
```

### DEFINITIONS:

kv1

First coefficient in the correlation for gas-liquid K value (kPa | psi).

kv2

Second coefficient in the correlation for gas-liquid K value (1/kPa | 1/psi).

kv3

Third coefficient in the correlation for gas-liquid K value.

kv4

Fourth coefficient in the correlation for gas-liquid K value (C | F). This coefficient has the unit of temperature difference. It has the same value for temperature scales C and K, and has the same value for temperature scales F and R.

kv5

Fifth coefficient in the correlation for gas-liquid K value (C | F). This coefficient has the unit of temperature, and is different for each temperature scale. Often this coefficient is quoted in other sources in absolute degrees K or R, even though all other temperatures are quoted in C or F. Here, this coefficient is quoted in the same unit as all other temperatures, so it may be necessary to convert it from absolute to C or F.

### DEFAULTS:

Each absent keyword implies a zero value (for each component) for the corresponding coefficient.

The absence of all these keywords implies zero K values, unless the table option is used instead. The exception is aqueous components (1 to NUMW) for which the default K values (all coefficients zero) are internal water K values.

In order to assign zero K value to an aqueous component, enter a non-zero value for KV4 and zeroes for the other coefficients. This forces the component to use the correlation, below; if kv1 = kv2 = kv3 = 0 then K will be zero for all p and T.

## **EXPLANATION:**

As a function of p and T, the K value is

$$K = (kv1/p + kv2 * p + kv3) * \text{EXP} (kv4 / (T-kv5))$$

where T is temperature and p is gas phase pressure. This expression for K accounts for curvature of the vapour pressure curve with pressure.

See Table 2 for suggested correlation values for selected components; values are available for kv1, kv4 and kv5, and kv2 = kv3 = 0. See Appendix D.3 for a discussion of the basis for the correlation form. See Appendix F.3 for a discussion of phase equilibrium relationships.

Use of the correlation forces you to enter values for all condensable components. If you want to use the default partitioning for a component, enter zero in that component's column.

This correlation applies only to gas-liquid K values. You must use the table to specify liquid-liquid partitioning.

Over the operating pressure and temperature ranges, each component's resulting correlation K value must be either all zero or all positive (greater than zero). Specifically, K values are not allowed to "dip" below zero for some p and T.

## **Bubble Point Pressure**

Keyword \*PBC (initial bubble point pressure) uses gas-liquid K value to convert from bubble point pressure to oil mole fraction via the formula  $X_i = 1 / K_i(P_{bi}, T)$ . In this case, the correlation must give K values greater than 1 at the specified  $P_{bi}$  and T. This condition is usually satisfied since (1) the component in question probably is a volatile gas with initial K values greater than 1 and increasing as pressure decreases, and (2) the  $P_{bi}$  do not exceed the initial reservoir pressure.

Subkeyword \*BPP of \*OUTPRN \*GRID reports oil mole fraction in the form of bubble point pressure by solving  $K_i(P_{bi}, T) = 1/X_i$  for  $P_{bi}$ . If  $P_{bi}$  cannot be found, then  $P_{bi} = 0$  is reported.

The correlation coefficients may define a function such that  $P_{bi}$  cannot be found at the desired T, in which case  $P_{bi} = 0$  is reported. This may happen when coefficient kv2 is non-zero and large. To see this, consider the equation  $K_i(P_{bi}, T) = 1/X_i$  written as  $a \cdot P_{bi} \cdot P_{bi} + b \cdot P_{bi} + c = 0$  where

$$a = kv2$$

$$b = kv3 - (1/X_i) \cdot \text{EXP} [ -kv4 / (T-kv5) ]$$

$$c = kv1$$

If kv2 is zero, the equation is linear and  $P_{bi} = -c/b$ . If  $b$  is close to zero or  $-c/b$  is not positive, then  $P_{bi}$  cannot be obtained.

If kv2 is non-zero, the equation is quadratic and the discriminant  $D = b \cdot b - 4 \cdot a \cdot c$ . If D is negative then there is no real root, and  $P_{bi}$  cannot be obtained. If D is non-negative then there are two real roots, and the minimum root above zero is chosen. If neither root satisfies this last condition, then  $P_{bi}$  cannot be obtained.

See also **Bubble Point Pressure** description under \*KVTABLE.

---

## K Value Tables

**\*KVKEYCOMP**

**\*GASLIQKV, \*LIQLIQKV, \*KVTABLIM, \*KVTABLE,**

**PURPOSE:**

Assign K values in table form. Liquid-liquid K values must be assigned this way.

**FORMAT:**

```
*GASLIQKV | *LIQLIQKV
*KVTABLIM plow phigh Tlow Thigh
*KVTABLE comp_name
  K_value_table
*KVKEYCOMP key_comp key_phase xlow xhigh (slope int)
*KVTABLE comp_name
  { *KEYCOMP
    K_value_table }
```

**DEFINITIONS:**

**\*GASLIQKV**

The following \*KVTABLIM, \*KVKEYCOMP and \*KVTABLE are applied to gas-liquid K values, until \*LIQLIQKV appears.

**\*LIQLIQKV**

The following \*KVTABLIM, \*KVKEYCOMP and \*KVTABLE are applied to liquid-liquid K values, until \*GASLIQKV appears.

**plow, phigh**

Low and high pressure limits in K value table (kPa | psi). The allowed range for plow is  $[10^{-3} \text{ kPa}, 10^6 \text{ kPa}]$  and for phigh is  $[plow + 1 \text{ kPa}, 10^8 \text{ kPa}]$ .

Look-up of K values more than 1 kPa outside this range is allowed only a limited number of times.

**Tlow, Thigh**

Low and high temperature limit in K value table (C | F). The allowed range for Tlow is from  $-100^\circ \text{ C}$  ( $-148^\circ \text{ F}$ ) to  $10^6$  (user's unit), and for Thigh it is from  $Tlow + 1 \text{ C}^\circ$  to  $10^6$  (user's unit). Look-up of K values more than  $1 \text{ C}^\circ$  outside this range is allowed only a limited number of times.

**comp\_name**

Component name in quotes. Component must exist in a liquid phase, that is, the component number must not exceed numx (see \*MODEL).

**K\_value\_table**

$K(Tlow,plow) \dots K(Tlow,phigh)$

.....

$K(Thigh,plow) \dots K(Thigh,phigh)$

The maximum allowed number of table entries assigned to a component, or a \*KEYCOMP set, is 1500. The maximum allowed number of rows (temperature values) is 50; the maximum allowed number of columns (pressure values) is 30. There must be at least two columns and two rows.

All gas-liquid tables must have the same number of columns and the same number of rows. All liquid-liquid tables must have the same number of columns and the same number of rows.

#### key\_comp

Quoted name of key component corresponding to the compositions  
\*KEYCOMP x.

#### key\_phase

Phase or interpolation option corresponding to the compositions  
\*KEYCOMP x. The following are allowed:

##### **SXY formulation only**

- |        |                                       |
|--------|---------------------------------------|
| W      | - water phase mole fraction, explicit |
| X or O | - oil phase mole fraction, explicit   |
| Y or G | - gas phase mole fraction, explicit   |
| YK     | - gas phase mole fraction, implicit   |
| M      | - maximum of W, X and Y, explicit     |

Note: Y and YK are available for condensable components only.

##### **ZT or ZH formulations (recommended)**

- |              |   |
|--------------|---|
| GLOBAL or Z  | - global mole fraction (implicit)   |
| RATIOW or RW | - use Hands rule tie-line parameter involving the ratio of Z(keycomp)/Z(1), implicit      |
| RATIOO or RO | - use Hands rule tie-line parameter involving the ratio of Z(keycomp)/Z(numw+1), implicit |

Note: All key\_phase options are available with Z formulations, but only these three are recommended.

Note: When WinProp writes \*KVKEYCOMP, it uses only Z, RW or RO.

See \*TFORM in the NUMERICAL METHODS CONTROL chapter for an explanation of formulations. The value of the interpolating quantity is updated either at the beginning of the timestep (explicit) or during the non-linear iteration process (implicit).

#### xlow, xhigh

Low and high composition limit in composition-dependent K value table.  
The allowed range for xlow is [0,1] and for xhigh is [xlow+1.0e-6,1].

slope

Slope of maximum tie line used in Hand's tie line interpolation parameter, used only for RW and RO.

int

Intercept of maximum tie line used in Hand's tie line interpolation parameter (for RW and RO only).

#### \*KEYCOMP

Denotes a composition-dependent K value table, and the following table corresponds to a certain composition of key\_comp in key\_phase. The first table corresponds to xlow, the last corresponds to xhigh, and intermediate tables correspond to equally spaced compositions in between xlow and xhigh. All gas-liquid tables with \*KEYCOMP must have the same number of \*KEYCOMP entries; the same comment applies to liquid-liquid tables.

#### DEFAULTS:

Absence of all gas-liquid K value data will result in default K values for aqueous components and zero K values for other components.

Absence of liquid-liquid K values for a component will result in no liquid-liquid solubility. Only aqueous components will go in the water phase; only oleic components will go in the oil phase.

The simulator starts reading data with \*GASLIQKV assumed.

If 'slope' and 'int' are absent, then slope = 1 and int = 0 are assumed.

#### CONDITIONS:

The gas-liquid K values of aqueous components (numbers 1 to numw) must be specified the same way, that is, either table or correlation.

The gas-liquid K values of oleic components (numbers numw+1 to numx) must be specified the same way, that is, either table or correlation.

The only combination not allowed is table for aqueous and correlation for oleic components.

This data will be applied to gas-liquid or liquid-liquid K values, depending on which of \*GASLIQKV or \*LIQLIQKV is in force.

If gas-liquid \*KKEYCOMP appears more than once then only the last instance is used. The same comment applies to liquid-liquid \*KKEYCOMP.

Over the operating pressure and temperature ranges, each component's table K value must be either all zero or all positive (greater than zero). Specifically, a component's K value table is not allowed to be zero for some p and T and positive for other p and T.

#### EXPLANATION:

The look-up table for component 'Pseudo 3':

P=	20	70	120	170	220
T=70	1.0	2.0	3.0	4.0	3.0
T=170	2.0	3.0	4.0	3.0	1.0
T=270	3.0	3.0	4.0	5.0	6.0

would be entered as:

```
*KVTABLIM 20 220 70 270  
*KVTABLE 'Pseudo 3'
```

1.	2.	3.	4.	3.
2.	3.	4.	3.	1.
3.	3.	4.	5.	6.

### Interpolation Between Table Entries

Between two K value table entries for two adjacent pressures, K varies linearly with 1/p.

Between two non-zero K value table entries for two adjacent temperatures,  $\ln(K)$  varies linearly with 1/T; when one of the K value entries is zero, K varies linearly with 1/T.

### Hand's Rule Option

Hand's rule tie line parameter for components A and B is defined as

$$R = Z(A) / (Z(B) * \text{slope} + \text{int}).$$

See the Appendices D.3 and F.3 for further discussion.

### Bubble Point Pressure

Keyword \*PBC (initial bubble point pressure) uses gas-liquid K value to convert from bubble point pressure to oil mole fraction via the formula  $X_i = 1 / K_i(P_{bi}, T)$ . In this case, the table must give K values greater than 1 at the specified  $P_{bi}$  and T. This condition is usually satisfied since (1) the component in question probably is a volatile gas with initial K values greater than 1 and increasing as pressure decreases, and (2) the  $P_{bi}$  do not exceed the initial reservoir pressure. Since K value table look-up does not extrapolate past the lower or upper pressure limits, the table pressure range should include anticipated  $P_{bi}$  values. If  $P_{bi}$  falls outside the K value table pressure range, the resulting mole fraction will correspond to the nearest table pressure entry instead of  $P_{bi}$ .

Subkeyword \*BPP of \*OUTPRN \*GRID reports oil mole fraction in the form of bubble point pressure by solving  $K_i(P_{bi}, T) = 1/X_i$  for  $P_{bi}$ . If  $P_{bi}$  cannot be found within the table pressure range at that T, then  $P_{bi} = 0$  is reported. If many output  $P_{bi}$  values are lower than the operating pressure, then  $P_{bi}$  reporting may require that the K value table be extended to lower pressures.

If composition-dependent K value tables are used,  $P_{bi}$  is based on the lowest \*KEYCOMP composition value.

### Table Look-ups Outside Specified Limits

The user should ensure that the pressure and temperature limits of the K value table exceed the expected operating range of the simulation run. A warning message is issued at the end of each timestep for each grid block p and T value that lies outside the table ranges. To avoid possible unphysical results and poor convergence, only a certain number of such warning messages are allowed before the simulation is terminated.

## Molecular Weight (Required)

\*CMM

### PURPOSE:

Assign molecular weights.

### FORMAT:

\*CMM cmm(1) ... cmm(ncomp)

### DEFINITIONS:

cmm

Molecular mass of component (kg/gmol | lb/lbmol). The unit is (mass/mole), even if \*MASSBASIS was specified.

### DEFAULTS:

Enter cmm(k) = 0 for aqueous component k to get the water default of 0.01802 kg/gmole (18.02 lb/lbmole).

### EXPLANATION:

Since many fluid properties are on a per-mole basis, cmm is very important. For example, liquid density, which determines the hydrostatic head of each phase, will depend directly on the mass density, that is, the product mole density times mass/mole.

The molecular masses of some common pure substances are:

Water	0.01802 kg/gmole (18.02 lb/lbmole)
Nitrogen	0.02801 kg/gmole (28.01 lb/lbmole)
Oxygen	0.03199 kg/gmole (31.99 lb/lbmole)

Very frequently a hydrocarbon component is actually a pseudo-component, representing a group of pure components over a range of C numbers. For example, a heavy oil pseudo-component may cover the range C15 to C30, as suggested by a distillation analysis. The mass density of this fraction can be measured directly, but its molecular mass usually is postulated or estimated using a mathematical model. The value of the cmm is, in itself, not critical. However, it is crucial that the mass density used in the simulator be equal to the measured mass density, in which case the \*MASSDEN density input option may be preferred.

A table of molecular masses for selected components is in Table 6.

---

## Critical Properties (Required)

\*TCRIT, \*PCRIT, \*IDEALGAS

### PURPOSE:

Assign critical pressure and temperature.

### FORMAT:

\*TCRIT terit(1) ... terit(numy)  
\*PCRIT pcrit(1) ... pcrit(numy)    or    \*IDEALGAS

### DEFINITIONS:

terit

Component critical temperature (C | F | C). Suggested values for selected components can be found in Table 5. You may enter zero for a nonvolatile component since such values are not used internally.

pcrit

Component critical pressure (kPa | psi | kPa). Suggested values for selected components can be found in Table 5. You may enter zero for a nonvolatile component since such values are not used internally.

\*IDEALGAS

Specifies that the ideal gas law is used, that is, compressibility factor Z is 1. This option will save some CPU, but can be very inaccurate if any component is not far from its critical point.

### DEFAULTS:

Enter tcrit(k) = 0 for aqueous component k to get the water value of 374.15 C (705.47 F).

If \*PCRIT is used, enter pcrit(k) = 0 for aqueous component k to get the water value of 22048 kPa (3198 psi).

### CONDITIONS:

\*TCRIT is a required keyword.

You must specify either \*PCRIT or \*IDEALGAS.

### EXPLANATION:

Critical temperatures TCR(I) are used in two property calculations:

- a) Gas density compressibility factor Z (see Appendix D.4), and
- b) Vapourization enthalpy (see **Calculation of Vapourization Enthalpies** under the manual page "Fluid Enthalpies" for keyword \*HVR, etc.).

---

## **Reference Conditions**

**\*K\_SURF, \*KL\_SURF, \*AQFRCOMP**

**\*PRSR, \*TEMR, \*PSURF, \*TSURF, \*SURFLASH,**

### **PURPOSE:**

Specify reference conditions for fluid properties and surface conditions.

### **FORMAT:**

```
*PRSR prsr
*TEMR temr
*PSURF psurf
*TSURF tsurf
*SURFLASH ( *SEGREGATED | phase_list | *KVALUE | *THERMAL )
*K_SURF 'name' Ks(i)
*KL_SURF 'name' KLs(i)
*AQFRCOMP 'name'
```

### **DEFINITIONS:**

**prsr**

Reference pressure (kPa | psi) corresponding to the densities entered by  
\*MOLDEN, \*MASSDEN or \*MOLVOL, and \*SOLID\_DEN.

**temr**

Reference temperature used by many T-dependent and thermal properties  
(C | F). See EXPLANATION, below.

**psurf**

Pressure corresponding to surface conditions, for reporting well rates and  
accumulations in terms of standard densities after being flashed to surface  
(kPa | psi).

**tsurf**

Temperature corresponding to surface conditions, for reporting well rates and  
accumulations in terms of standard densities after being flashed to surface (C|F).

#### **\*SEGREGATED**

Components are segregated into single phases. See keyword \*MODEL for  
definition of component types.  $K_s(i)$  denotes the gas-liquid K value of  
component i, either entered via \*K\_SURF or calculated at psurf and tsurf.

Aqueous component i:	Water phase if $K_s(i) < 1$ ; gas phase otherwise
Oleic component i:	Oil phase if $K_s(i) < 1$ ; gas phase otherwise
Noncondensable gas:	Gas phase

For example, a black-oil type component set would default to:

Water component in water phase, since  $K_s(1) = .001$

Dead oil in oil phase, since  $K_s(2) = 0$

Solution gas in gas phase, since  $K_s(3) = 50$

#### phase\_list

A list of numy phase designators W (water phase), O (oil phase) or G (gas phase), indicating into which phase each component is segregated for surface condition (SC) production reporting purposes. Generally, use of \*K\_SURF with \*SEGREGATED is recommended instead of phase\_list.

G is available only if the component exists in the gas phase at SC. G is required for non-condensable gases. It is recommended that W and O be used only if the condensable component exists in the respective phase at SC. It is recommended that \*KVALUE be used if at least one condensable component exists in both liquid phases at SC.

#### \*KVALUE, \*THERMAL

For production reporting purposes components are partitioned into phases by an isothermal (\*KVALUE) or thermal (\*THERMAL) flash, according to K values calculated at conditions PSURF and TSURF or specified by

\*K\_SURF and \*KL\_SURF. A component may appear in more than one phase, e.g., solution gas component in both the oil and gas phases.

\*THERMAL is not available with \*K\_SURF and \*KL\_SURF.

#### \*K\_SURF 'name' $K_s(i)$

For gas-liquid K value of component 'name' at surface conditions, use  $K_s(i)$  instead of the value calculated from K value data at purf and tsurf. This value is used for both the \*SEGREGATED and \*KVALUE options, but is not used if the explicit phase\_list is specified.

This option is useful when K value tables do not extend to surface conditions, in which case the default surface K value is that at the lowest table pressure and temperature. Also, when using \*KVALUE and the component's default K value is non-zero, you can reduce the number of surface phases of the component by setting  $K_s(i) = 0$ . On the other hand, a non-volatile component can be forced to report in the gas phase with \*SEGREGATED when  $K_s(i) > 1$ , which is useful for non-equilibrium processes like foamy oil.

#### \*KL\_SURF 'name' $KL_s(i)$

Same as \*K\_SURF, but for the component's liquid-liquid K value.

#### \*AQFRCOMP 'name'

Specify which aqueous component is the fluid in the \*AQUIFER model.

## **DEFAULTS:**

If \*PRSR is absent, prsr = 100 kPa.

If \*TEMR is absent, temr = 25 C.

If \*PSURF is absent, psurf = 101 kPa = 14.65 psi.

If \*TSURF is absent, tsurf = 290 K = 62 F.

If \*SURFLASH is absent, then \*SURFLASH \*SEGREGATED is assumed.

If \*AQFRCOMP is absent, then the aquifer fluid is component #1.

If \*K\_SURF is absent for a component, that component's surface gas-liquid K value is the value calculated from user input data at psurf and tsurf. For the table calculation the p and T values used are those within the table's p and T range closest to psurf and Tsurf. The same comments apply to \*KL\_SURF and surface liquid-liquid K value.

## **CONDITIONS:**

\*SURFLASH must be followed by either \*SEGREGATED, a list of phase designators, \*KVALUE or \*THERMAL.

## **EXPLANATION:**

TEMR is used in conjunction with the following input data:

1. Liquid density data (\*MOLDEN, \*MASSDEN or \*MOLVOL),
2. Liquid and gas phase enthalpy data (\*CPL1, \*CPG1, etc.),
3. Formation heat capacity (\*ROCKCP),
4. Reaction enthalpy data (\*RENTH). Most reaction enthalpy data is referred to 25 C, so the default value is recommended for combustion simulations.
5. Wellbore heatloss option, where TEMR may be the temperature of water entering the steam boiler.

## **Reporting Well Performance at Surface Conditions**

When well performance is reported in terms of phase volumes at surface conditions, the corresponding phase densities and compositions must be defined. Densities of each component are based on \*PSURF and \*TSURF. Mixing of components in a phase is assumed to be ideal, that is, the total phase volume is the sum of the individual component volumes in that phase. The \*SURFLASH options indicate how to partition the components between phases. Both \*SEGREGATED and "phase\_list" result in each component being partitioned entirely in the indicated phase, but the other options partition each component between multiple phases according to K values. See Appendix A.5 for further discussion.

## **Aquifers and Multiple Aqueous Components**

The analytical aquifer enabled via keyword \*AQUIFER assumes that the fluid in the aquifer pore space consists entirely of the "aquifer component" in the water phase. When there is only one aqueous component (numw = 1 for keyword \*MODEL) then that must be the aquifer component. When there are multiple aqueous components you can choose to override the default aquifer component (#1) via keyword \*AQFRCOMP.

There is a condition on the fluid which flows from reservoir to aquifer, that is, the aquifer model can accept from the reservoir only the aquifer component. However, the composition of fluid flowing from reservoir to aquifer depends on the composition of the water phase in the reservoir. At the end of each timestep the water phase composition is checked for each block experiencing flow into an aquifer: if the mole fraction of the aquifer component is less than 99%, a warning message is issued. After a number of these messages the run is terminated. One strategy to satisfy this constraint is to assign water phase composition of 100% aquifer component in blocks around the aquifer as a buffer zone.

---

## **Fluid Enthalpies** \*CPG1, \*CPG2, \*CPG3, \*CPG4, \*CPL1, \*CPL2, \*CPL3, \*CPL4, \*HVR, \*EV, \*HVAPR, \*WATPENTH

### **PURPOSE:**

Over-ride defaults for fluid heat capacities.

### **FORMAT:**

```
*CPG1 cpg1(1) ... cpg1(numy)
*CPG2 cpg2(1) ... cpg2(numy)
*CPG3 cpg3(1) ... cpg3(numy)
*CPG4 cpg4(1) ... cpg4(numy)
*CPL1 cpl1(1) ... cpl1(numy)
*CPL2 cpl2(1) ... cpl2(numy)
*CPL3 cpl3(1) ... cpl3(numy)
*CPL4 cpl4(1) ... cpl4(numy)
*HVR hvr(1) ... hvr(numx)
*EV ev(1) ... ev(numx)
*HVAPR hvapr(1) ... hvapr(numx)
*WATPENTH
```

### **DEFINITIONS:**

- \*CPG1 *cpg1*(1) ... *cpg1*(numy)  
First coefficient (J/gmol-C | Btu/lbmol-F) in gas phase heat capacity correlation, for each fluid component (see \*MODEL).
- \*CPG2 *cpg2*(1) ... *cpg2*(numy)  
Second coefficient (J/gmol-C<sup>2</sup> | Btu/lbmol-F<sup>2</sup>) in gas phase heat capacity correlation, for each fluid component (see \*MODEL).
- \*CPG3 *cpg3*(1) ... *cpg3*(numy)  
Third coefficient (J/gmol-C<sup>3</sup> | Btu/lbmol-F<sup>3</sup>) in gas phase heat capacity correlation, for each fluid component (see \*MODEL).
- \*CPG4 *cpg4*(1) ... *cpg4*(numy)  
Fourth coefficient (J/gmol-C<sup>4</sup> | Btu/lbmol-F<sup>4</sup>) in gas phase heat capacity correlation, for each fluid component (see \*MODEL).
- \*CPL1 *cpl1*(1) ... *cpl1*(numy)  
First coefficient (J/gmol-C | Btu/lbmol-F) in liquid phase heat capacity correlation, for each fluid component (see \*MODEL).
- \*CPL2 *cpl2*(1) ... *cpl2*(numy)  
Second coefficient (J/gmol-C<sup>2</sup> | Btu/lbmol-F<sup>2</sup>) in liquid phase heat capacity correlation, for each fluid component (see \*MODEL).

**\*CPL3** *cpl3(1) ... cpl3(numy)*

Third coefficient (J/gmol-C<sup>3</sup> | Btu/lbmol-F<sup>3</sup>) in liquid phase heat capacity correlation, for each fluid component (see \*MODEL).

**\*CPL4** *cpl4(1) ... cpl4(numy)*

Fourth coefficient (J/gmol-C<sup>4</sup> | Btu/lbmol-F<sup>4</sup>) in liquid phase heat capacity correlation, for each fluid component (see \*MODEL).

**\*HVR** *hvr(1) ... hvr(numx)*

First coefficient (J/gmol-C<sup>ev</sup> | Btu/lbmol-F<sup>ev</sup>) in vapourization enthalpy correlation, for each condensable component (see \*MODEL). The unit for *hvr(i)* depends upon the value of *ev(i)*. \*HVR should not be used for the third enthalpy base option “Liquid and Vapour Heat Capacities” (see \*HVAPR description and **Enthalpy Base Option**, below).

**\*EV** *ev(1) ... ev(numx)*

Second coefficient in vapourization enthalpy correlation, for each condensable component (see \*MODEL). The unit for *hvr(i)* depends upon the value of *ev(i)*. \*EV is not used for the third enthalpy base option “Liquid and Vapour Heat Capacities” (see \*HVAPR description and **Enthalpy Base Option**, below).

**\*HVAPR** *hvapr(1) ... hvapr(numx)*

Vapourization enthalpy (J/gmol | Btu/lbmol) at reference temperature \*TEMR, for each condensable component (see \*MODEL). \*HVAPR should be used only for the third enthalpy base option “Liquid and Vapour Heat Capacities” (see \*HVR description and **Enthalpy Base Option**, below).

**\*WATPENTH**

Enthalpy of water in liquid phase depends upon pressure as well as temperature, using data from National Institute of Standards and Technology (<http://webbook.nist.gov/chemistry/fluid>) in table form. This table is limited to 60 MPa (8700 psi), so for higher pressures the enthalpy will be at 60 MPa and the requested temperature. This table is also limited to the 647 K (705 F), the critical temperature of water, so for higher temperatures there is no pressure dependence. See **Pressure-Dependent Water Enthalpy**, below.

This option is available only if the internal (default) enthalpy method is chosen for aqueous component.

## DEFUALTS:

If none of these keywords is present, the fluid heat capacities will be:

- a) Aqueous components: liquid and vapour from internal water table from the Engineering data book, Gas Processors Suppliers Association (1980), itself adapted from *Steam Tables*, J.H. Keenan, F.G. Keyes, P.G. Hill, J.G. Moore (J. Wiley and Sons, 1969),

- b) Oleic components: vapour is 0.25 Btu/lb-F; liquid is 0.5 Btu/lb-F for liquid-based enthalpy options, 0.25 Btu/lb-F for gas-based enthalpy option,
- c) All other components: vapour is 0.25 Btu/lb-F.

These defaults usually are quite adequate and are recommended unless you have specific values to use. When the gas-based enthalpy option is used, for a “dead” component assign a liquid value to the gas keyword *cpg1(i)*, etc.

Defaulting is done on a per-component basis. Therefore, if you want to over-ride the default for a particular component, enter zeroes in that component’s position for each enthalpy definition keyword that appears in the data. It is necessary to do this since the enthalpy keywords require a number for each component. For example, to over-ride the default of only component #3 out of 5 components, and only for liquid heat capacity, use the following:

```
*CPL1 0 0 30.5 0 0
```

Here components #1, #2, #4 and #5 each will end up with zero for all enthalpy data, which indicates that the defaults above are to be used.

The absence of \*EV results in *ev* = 0.38.

If \*WATPENTH is absent, liquid water enthalpy depends only on temperature.

#### **EXPLANATION:**

#### **Temperature Units and Correlation Coefficients**

Heat capacity correlation coefficients correspond to the correlation T expressed in absolute degrees, and STARS accepts directly such coefficients for all STARS input temperature units. For example, when input temperature unit is C, STARS still assumes that coefficients *cpg1*, etc., correspond to the correlation T expressed in K. See the example in Table 5 **Gas Heat Capacity Coefficients for Selected Components**.

If coefficients from another source are quoted for temperature in non-absolute degrees (C or F) then they must be converted. Let the correlation in non-absolute degrees  $T_N$  (C or F) be  $A_1 + A_2*T_N + A_3*T_N^2 + A_4*T_N^3$ . By substituting  $T_N = T - T_D$  ( $T_D = 273.15$  for C and  $459.67$  for F), rewrite the correlation in terms of absolute temperature T and gather terms to get

$$cpg1 = A_1 - A_2*T_D + A_3*T_D^2 - A_4*T_D^3$$

$$cpg2 = A_2 - 2*A_3*T_D + 3*A_4*T_D^2$$

$$cpg3 = A_3 - 3*A_4*T_D$$

$$cpg4 = A_4$$

For example, assume that gas phase heat capacity of component N2 is quoted as  $7.077 - 3.412e-4*T_F$  Btu/lbmol-F where  $T_F$  is the temperature in F. Here  $A_1 = 7.077$ ,  $A_2 = -3.412e-4$ ,  $A_3 = A_4 = 0$  and  $T_D = 459.67$ . The resulting STARS coefficients are *cpg1* = 7.234 and *cpg2* =  $-3.412e-4$ . See British-unit 2-coefficient entries for component N2 in Table 5.

#### **Enthalpy Base Option**

With the above keywords it is possible to define for each condensable component the following three quantities as a function of temperature:

- a) Heat capacity in a liquid phase CPL(T),
- b) Heat capacity in the gas phase CPG(T), and
- c) Enthalpy of vapourization HVAP(T).

However, only two of these three quantities are independent, since they are related by the definition

$$\text{HVAP}(T) = \text{HG}(T) - \text{HL}(T).$$

Here, HL(T) is component enthalpy in a liquid phase, defined by  $\text{CPL}(T) = d(\text{HL}(T))/dT$ ; HG(T) is component enthalpy in the gas phase, defined by  $\text{CPG}(T) = d(\text{HG}(T))/dT$ , and  $T_c$  is component critical temperature.

Three enthalpy datum options are available, allowing the user to choose which two of the three quantities, above, to enter as data. The option is determined by the choice of keywords.

#### 1. Liquid Heat Capacities

Enter coefficients \*CPL1 to 4 for liquid heat capacity and coefficients \*HVR and \*EV for vapourization enthalpy. Enthalpy datum is liquid phase at  $T = \text{TEMR}$ .

Condensable Components:

$$\text{CPL}(T) = cpl1 + cpl2 \cdot T + cpl3 \cdot T^2 + cpl4 \cdot T^3$$

$$\text{HL}(T) = \int \text{CPL}(\tau) d\tau \text{ from TEMR to } T$$

This gives  $\text{HL}(\text{TEMR}) = 0$  (enthalpy datum)

$$\text{HVAP}(T) = \text{hvr} \cdot (T_c - T)^{\text{ev}} \text{ for } T < T_c, 0 \text{ otherwise}$$

$$\text{HG}(T) = \text{HL}(T) + \text{HVAP}(T)$$

This gives  $\text{HG}(T) = \text{HL}(T)$  for  $T \geq T_c$ .

Non-condensable Components:

$$\text{CPG}(T) = cpl1 + cpl2 \cdot T + cpl3 \cdot T^2 + cpl4 \cdot T^3$$

$$\text{HG}(T) = \int \text{CPG}(\tau) d\tau \text{ from TEMR to } T$$

#### 2. Vapour Heat Capacities

Enter coefficients \*CPG1 to 4 for vapour heat capacity and coefficients \*HVR and \*EV for vapourization enthalpy. Enthalpy datum is gas phase at  $T = \text{TEMR}$ .

Condensable Components:

$$\text{CPG}(T) = cpg1 + cpg2 \cdot T + cpg3 \cdot T^2 + cpg4 \cdot T^3$$

$$\text{HG}(T) = \int \text{CPG}(\tau) d\tau \text{ from TEMR to } T$$

This gives  $\text{HG}(\text{TEMR}) = 0$  (enthalpy datum)

$$\text{HVAP}(T) = \text{hvr} \cdot (T_c - T)^{\text{ev}} \text{ for } T < T_c, 0 \text{ otherwise}$$

$$\text{HL}(T) = \text{HG}(T) - \text{HVAP}(T)$$

This gives  $\text{HG}(T) = \text{HL}(T)$  for  $T \geq T_c$ .

Non-condensable Components:

$$CPG(T) = cpg1 + cpg2 \cdot T + cpg3 \cdot T^2 + cpg4 \cdot T^3$$

$$HG(T) = \int CPG(\tau) d\tau \text{ from TEMR to } T$$

### 3. Liquid and Vapour Heat Capacities

Enter both liquid and vapour heat capacity coefficients \*CPL1 to 4 and \*CPG1 to 4, along with \*HVAPR. Enthalpy datum is liquid phase at  $T = \text{TEMR}$ .

Condensable Components:

$$CPL(T) = cpl1 + cpl2 \cdot T + cpl3 \cdot T^2 + cpl4 \cdot T^3$$

$$HL(T) = \int CPL(\tau) d\tau \text{ from TEMR to } T$$

This gives  $HL(\text{TEMR}) = 0$  (enthalpy datum)

$$CPG(T) = cpg1 + cpg2 \cdot T + cpg3 \cdot T^2 + cpg4 \cdot T^3$$

$$HG(T) = hvapr + \int CPG(\tau) d\tau \text{ from TEMR to } T$$

This gives  $HG(\text{TEMR}) = hvapr$ , vapourization enthalpy at TEMR.

The user should enter data that satisfies  $HL(T_c) = HG(T_c)$ , although this is not enforced.  $HL(T) = HG(T)$  is not enforced above  $T = T_c$ , which limits the usefulness of this option in that T range.

Non-condensable Components:

$$CPG(T) = cpg1 + cpg2 \cdot T + cpg3 \cdot T^2 + cpg4 \cdot T^3$$

$$HG(T) = \int CPG(\tau) d\tau \text{ from TEMR to } T$$

## Calculation of Vapourization Enthalpies

The component vapourization enthalpy  $HVAP(T)$  may be modelled as a function of temperature using Watson's correlation

$$HVAP(T) = HVAP(Tb) \cdot ((Tc - T) / (Tc - Tb))^{ev}$$

where  $T_c$  is the component's critical temperature,  $T_b$  is the component's normal boiling temperature,  $HVAP(Tb)$  is the component's vapourization enthalpy at  $T_b$ , and  $ev$  is a constant with a value between 0.375 and 0.38. The constant part of  $HVAP(T)$  for component I is lumped into

$$HVR = HVAP(Tb) / (Tc - Tb)^{ev}$$

Suggested values of HVR for selected components are included in Table 8.

Alternatively, to determine  $HVAP(Tb)$  if  $T_b$ ,  $T_c$  and critical pressure  $P_c$  are known, the Reidal correlation can be used:

$$HVAP(Tb) = 1.093 \cdot R \cdot T_c \cdot ((Tb/Tc) * (\ln(Pc) - 1) / (0.93 - Tb/Tc))$$

where  $T_b$  and  $T_c$  are in degrees K,  $P_c$  is in atm,  $R$  is the gas constant 1.987 cal/gm mole-K, and  $HVAL(Tb)$  is in cal/gm mole.  $HVAP(Tb)$  must be converted to the correct user input units.

## Phase Enthalpies

The enthalpies and internal energies of the water, oil, gas and phases are calculated as follows. Note that oil phase pressure  $P_o$  is used for the work term in all three phases.

Water:

$\text{ENTHW}(T)$  = water mole fraction weighted sum of  $\text{HL}(T)$  of component I, for I from 1 to  $\text{NUMX}$ .

$\text{UINW}(T) = \text{ENTHW}(T) - P_o/\text{DENW}$ , where  $\text{DENW}$  is water phase mole density.

Oil:

$\text{ENTHO}(T)$  = oil mole fraction weighted sum of  $\text{HL}(T)$  of component I, for I from 1 to  $\text{NUMX}$ .

$\text{UINO}(T) = \text{ENTHO}(T) - P_o/\text{DENO}$ , where  $\text{DENO}$  is oil phase mole density.

Gas:

$\text{ENTHG}(T)$  = gas mole fraction weighted sum of  $\text{HG}(T)$  of component I, for I from 1 to  $\text{NUMY}$ .

$\text{UING}(T) = \text{ENTHG}(T) - P_o/\text{DENG}$ , where  $\text{DENG}$  = gas phase mole density.

## Exceeding Maximum Pressure of Internal Steam Table

The internal table for steam (water vapour) enthalpy depends on both temperature and pressure. This pressure data consists of tables up to 60 MPa (8700 psi). If a block or well fluid pressure exceeds this value, the enthalpy returned will be the value at the table's maximum pressure and the requested temperature. This measure is necessary since extrapolating pressure outside the table leads to incorrect temperature dependence and hence heat capacity values; it is acceptable because enthalpy dependence on pressure is low at these pressures.

## Pressure-Dependent Water Enthalpy

Since the compressibility of liquid water is very small, ignoring the pressure dependence of liquid water enthalpy is a good approximation for most thermal EOR processes. However, for applications with low temperatures and high pressures, such as gas hydrate recovery, the pressure dependence of liquid water enthalpy can be significant, in which case keyword \*WATPENTH can be used.

There will be result differences between runs whose data differs only in \*WATPENTH. The magnitude of the result differences is an indication of the relative importance of pressure dependence of liquid water enthalpy.

## Solid Phase Properties (Required)

\*SOLID\_DEN, \*SOLID\_CP,

\*GASSYSLD

### PURPOSE:

Assign properties of components in the solid/adsorbed/trapped phase.

### FORMAT:

*SOLID_DEN	'name' density cp ct ( cpt )
*SOLID_CP	'name' cps1 cps2
*GASSYSLD	'name'

### DEFINITIONS:

'name'

Component name in quotes, defined via keywords \*MODEL and \*COMPNAME.

density

Mass density  $\rho_{k0}$  (kg/m<sup>3</sup> | lb/ft<sup>3</sup> | kg/cm<sup>3</sup>) at reference pressure PRSR and temperature TEMR.

cp

Compressibility (1/kPa | 1/psi) at constant temperature.

ct

Thermal expansivity (1/C | 1/F) at constant pressure.

cpt

Pressure-temperature cross term (1/kPa-C | 1/psi-F) for density.

cps1

First coefficient in solid heat capacity correlation (J/gmol-C | Btu/lbmol-F).

cps2

Second coefficient in solid heat capacity correlation (J/gmol-C\*\*2 | Btu/lbmol-F\*\*2).

\*GASSYSLD

Indicates that the density of the indicated component in the trapped (adsorbed/solid) phase will be calculated using a gas-like compressibility instead of value cp entered via \*SOLID\_DEN. See "Gassy Solids" in the EXPLANATION below.

### DEFAULTS:

For each adsorbed/trapped **condensable component**, if \*SOLID\_DEN is absent then  $\rho_{k0}$ , cp, ct and cpt are obtained from \*MASSDEN (or equivalent), \*CP, \*CT1 and \*CPT, for that component's liquid reference phase. If \*SOLID\_CP is absent then cps1 and cps2 are

obtained from \*CPL1 and \*CPL2 when the fluid enthalpy is referenced to the liquid phase; otherwise the default is the same as for solid component.

If \*SOLID\_DEN appears but its cpt is absent, cpt = 0 is assumed for that component.

For each **solid** and adsorbed/trapped **non-condensable component**, \*SOLID\_DEN is required.

If \*SOLID\_CP is absent then cps1 = 17 J/gmole-C (4.06 Btu/lbmole-F) and cps2 = 0.

The absence of \*GASSYSLD implies that no component uses the gas-like compressibility option.

#### **EXPLANATION:**

The density of component k in the solid phase at pressure p and temperature T is given by

$$\rho_{sk}(p,T) = \rho_{k0} \cdot \exp[ cp \cdot (p - PRSR) - ct \cdot (T TEMR) + cpt \cdot (p - PRSR) \cdot (T TEMR) ]$$

The total solid phase volume is the sum of

$$C_{ck} / \rho_{sk}(p,T)$$

over all components k found in that phase, where  $C_{ck}$  is the solid phase concentration of component k in the pore space.

When this heat capacity model is being used (see **DEFAULTS**), the heat capacity is

$$Cp(T) = cps1 + (cps2 * T)$$

where T is the temperature in absolute degrees (K or R). If cps2 from another source is quoted for T in non-absolute degrees (i.e., C or F), then cps2 must be converted.

#### **Fluid Porosity Calculation**

Solid phase component density accounts for porosity variations caused by the amount of solid and adsorbed or trapped component present in the pore space.

Consider a grid block with a void (no-fuel) porosity of

$$\phi_v = 0.30$$

and fuel concentration of 1.8 lb/reservoir ft<sup>3</sup>, also expressed as  $C_c = 6.0$  lb/ pore ft<sup>3</sup>. Assume coke fuel has a pure density of

$$\rho_c = 60 \text{ lb/ft}^3$$

Therefore, the fluid porosity

$$\phi_f = \phi_v [1 - C_c / \rho_c] = 0.27$$

at this value of  $C_c$ , but will vary with time as the  $C_c$  varies. For each solid component the ratio  $C_c / \rho_c$  represents the fraction of the void porosity that it occupies, and these fractions must be summed to obtain  $\phi_f$ .

#### **Gassy Solids**

Nonequilibrium processes involving gas evolution such that a quantity of immobile gas remains trapped in the reservoir include foams, foamy oils, and gas evolution from hydrates and coal.

When these trapped bubbles are viewed as a "dispersed component" in the trapped phase, the keyword \*GASSYSLD allows this component to import a gas-like compressibility to this phase.

It is important when using this keyword that the appropriate component partial molar density corresponding to the reference pressure be employed. Thus when \*PRSR is high, an almost liquid-like density is appropriate, while if \*PRSR is at or near surface pressure, a gas-like partial molar density should be employed.

Most often it is expected that this keyword is used in conjunction with a nonequilibrium mass transfer expression (via the chemical reaction model) which quantifies the rate of creation, and possibly the rate of coalescence, of this gas-like dispersed component.

#### **Conversion from Obsolete Keywords \*SOLDEN and \*ADSDEN**

To convert mole density from obsolete keywords \*SOLDEN and \*ADSSEN to mass density  $\rho_{k0}$  for \*SOLID\_DEN, multiply by the component's molecular mass. The default of \*SOLDEN and \*ADSSEN was 48000 gmole/m<sup>3</sup> (2.997 lbmole/ft<sup>3</sup>).

## Liquid Phase Designation

\*LIQPHASE, \*WATPHASE, \*OILPHASE

### PURPOSE:

Specify the liquid phase(s) to which the subsequent liquid data is to be assigned.

### FORMAT:

\*LIQPHASE  
\*WATPHASE  
\*OILPHASE

### DEFINITIONS:

#### \*LIQPHASE

Assign the following liquid phase data to both the water and oil phases. This is the default and should be overwritten only if necessary.

#### \*WATPHASE

Assign the following liquid phase data to the water phase only. Use this when a component is found in both liquid phases but its component property is not the same in the two phases.

#### \*OILPHASE

Assign the following liquid phase data to the oil phase only. Use this when a component is found in both liquid phases but its component property is not the same in the two phases.

### DEFAULTS:

The reading of data begins with the assumption of \*LIQPHASE.

A specified phase designator stays in effect until over-written by another one further down the data file.

### EXPLANATION:

This option applies to liquid densities and viscosities, that is, the following keywords

*MOLDEN	*CT2	*DNMIXENDP
*MASSDEN	*AVISC	*DNMIXFUNC
*MOLVOL	*BVISC	*VSMIXCOMP
*CP	*VISCTABLE	*VSMIXENDP
*CT1	*DNMIXCOMP	*VSMIXFUN

\*WATPHASE and \*OILPHASE are needed only if the contribution of a component to a liquid phase property depends on the phase.

Ideal mixing of components implies that a component's contribution to a phase depends only on the pure component's corresponding property and the mole fraction of the component in the phase. This corresponds to \*LIQPHASE case. For example, you may assume that CO<sub>2</sub> has the same liquid density in the water phase as in the oil phase; the difference showing up only in the difference between the mole fraction of CO<sub>2</sub> in water and in oil.

An example of a need for the other option might be assuming that a component is dissolved in the water phase but forms an emulsion in the oil phase. The effect of the component on the oil phase would be completely different from the water phase.

Internally there are two copies of density and viscosity definition data – one for each of the liquid phases. For example, keywords \*MOLDEN, \*MASSDEN and \*MOLVOL end up assigning a value den(k) for each condensable component k. Internally there is space for the water and oil phases separately, namely denw(k) and deno(k), respectively.

The following illustrates how to assign different property data to the water and oil phases.

```
*WATPHASE
  *MASSDEN . . .    ** Assign to denw(k) only
*OILPHASE
  *MASSDEN . . .    ** Assign to deno(k) only
```

Keyword \*LIQPHASE causes data to be assigned to both water and oil phases.

```
*LIQPHASE
  *MASSDEN . . .    ** Assign to both denw(k) and deno(k)
```

The following shows that you can use all three phase designation keywords if not all the density and/or viscosity data is different between the phases.

```
*WATPHASE
  *MASSDEN . . .    ** Assign to denw(k) only
*OILPHASE
  *MASSDEN . . .    ** Assign to deno(k) only
.
*LIQPHASE
  *VISCTABLE . . . ** Liquid viscosities the same in two phases
```

This last example shows that each phase designation keyword sets a rule, but does not assign data itself. \*WATPHASE indicates that data following it is assigned only to the water phase. The keyword \*LIQPHASE must appear in order to change the “assignment” rule for viscosity data.

Note that the order of appearance of these phase designators can be significant. For example, to assign the same data to both liquid phases, with one exception, \*LIQPHASE must appear first and the exception must appear below it. The following data fragments illustrate this.

```
*** CORRECT ***
*LIQPHASE
  data including *MASSDEN . . .    ** Assign to both phases
*OILPHASE
  *MASSDEN . . .    ** Exception is kept

*** INCORRECT ***
*OILPHASE
  *MASSDEN . . .    ** Assign to oil phase only
*LIQPHASE
  data including *MASSDEN . . .    ** Exception is lost
```

## Liquid Densities (Required)

\*CT1, \*CT2, \*CPT, \*GASSYLIQ

\*MOLDEN, \*MASSDEN, \*MOLVOL, \*CP,

### PURPOSE:

Assign component liquid densities.

### FORMAT:

*MOLDEN	den(1)	...	den(numx)
*MASSDEN	denm(1)	...	denm(numx)
*MOLVOL	vol(1)	...	vol(numx)
*CP	cp(1)	...	cp(numx)
*CT1	ct1(1)	...	ct1(numx)
*CT2	ct2(1)	...	ct2(numx)
*CPT	cpt(1)	...	cpt(numx)
*GASSYLIQ	comp_name		

### DEFINITIONS:

den

Partial molar density (inverse of partial molar volume) at reference pressure PRSR and temperature TEMR ( $\text{gmol}/\text{m}^3$  |  $\text{lbmol}/\text{ft}^3$  |  $\text{gmol}/\text{cm}^3$ ).

denm

Mass density at reference pressure PRSR and temperature TEMR ( $\text{kg}/\text{m}^3$  |  $\text{lb}/\text{ft}^3$  |  $\text{kg}/\text{cm}^3$ ). This is partial molar density times molecular mass.

vol

Partial molar volume at reference pressure PRSR and temperature TEMR ( $\text{m}^3/\text{gmol}$  |  $\text{ft}^3/\text{lbmol}$  |  $\text{cm}^3/\text{gmol}$ ).

cp

Liquid compressibility ( $1/\text{kPa}$  |  $1/\text{psi}$ ) at constant temperature.

ct1

First coefficient of the thermal expansion correlation ( $1/\text{C}$  |  $1/\text{F}$ ).  $ct1$  is the thermal expansion coefficient when  $ct2 = 0$ .

ct2

Second coefficient of the thermal expansion correlation ( $1/\text{C}^{**2}$  |  $1/\text{F}^{**2}$ ). The thermal expansion coefficient is  $ct1 + T \cdot ct2$  where T is temperature expressed in absolute degrees (R or K). See **Second Temperature Coefficient** below.

cpt

Pressure-temperature cross term for liquid density ( $1/\text{kPa-C}$  |  $1/\text{psi-F}$ ).

## \*GASSYLIQ

Indicates that the liquid density of the specified component will be calculated using a gas-like compressibility instead of the number entered via \*CP. See "Gassy Liquids" in the EXPLANATION below.

This keyword may be used for multiple components.

If keyword \*CP is present there still must be "numx" numbers after it, but the cp value for this component will not be used internally.

## comp\_name

Component name in quotes defined via keywords \*MODEL and \*COMPNAME of a liquid component.

## DEFAULTS:

The absence of \*CP implies that all cp(k) = 0.

The absence of \*CT1 implies that all ct1(k) = 0.

The absence of \*CT2 implies that all ct2(k) = 0.

The absence of \*CPT implies that all cpt(k) = 0.

There is an additional default for each aqueous component k up to numw: if all of cp(k), ct1(k), ct2(k) and den(k) or vol(k) are zero, then that component k will be assigned internal values for water liquid density from the paper J. Phys. Chem. Ref. data, Vol. 16, No. 4, 1987. All temperatures are expressed in absolute degrees (K or R).

- $cp(k) = 4.57e-7 - 1.076823e-12 \cdot (prsr-101.325) \text{ 1/kPa}$ , prsr in kPa; or  $3.15e-6 - 5.119e-11 \cdot (prsr-14.7) \text{ 1/psi}$ , prsr in psi. This results in a value of  $4.57e-7 \text{ 1/kPa}$  ( $3.15e-6 \text{ 1/psi}$ ) at 1 atm, and a value of  $3.48e-7 \text{ 1/kPa}$  ( $2.4e-6 \text{ 1/psi}$ ) at 1000 atm.
- $ct1(k) = -1.9095e-3 \text{ 1/K}$ , and
- $ct2(k) = 7.296e-6 \text{ 1/K}^{**2}$ .
- critical density  $\rho_c = 1.788888e4 \text{ gmole/m}^3$ , corresponding to  $322.36 \text{ kg/m}^3$  when cmm is  $0.01802 \text{ kg/gmol}$ .
- if  $T < T_{cr}(k)$ , density at p and T is
$$1/V_w(k) = \rho_c (1 + \alpha) \exp [cp(k)(p - psat(T))]$$
$$\alpha = 1.99206 \tau^{**}(1/3) + 1.10123 \tau^{**}(2/3) - 0.512506 \tau^{**}(5/3)$$
$$- 1.75263 \tau^{**}(16/3) - 45.4485 \tau^{**}(43/3) - 675615 \tau^{**}(110/3)$$
$$\tau = 1 - [T / T_{cr}(k)]$$

where  $T_{cr}(k)$  is critical temperature and  $psat(T)$  is water saturation pressure at temperature T.

- If  $T \geq T_{cr}(k)$ , density at p and T is
- $1/V_w(k) = \rho_c \exp [cp(k)(p - psat(T)) - ct1(k)(T - T_{cr}(k))$   
 $- ct2(k)(T(T - T_{cr}(k))T_{cr}(k))]$

Each aqueous component is defaulted independently.

It is not possible to enter correlation coefficients for an aqueous component that will cause the correlation to match the default aqueous density calculation.

The absence of \*GASSYLIQ implies that no component uses the gas-like compressibility option.

### CONDITIONS:

One of \*MOLDEN, \*MASSDEN and \*MOLVOL is required, and a non-zero value must be specified for each oleic component.

The phase to which this data will be assigned depends on which of \*LIQPHASE, \*WATPHASE and \*OILPHASE is in force.

### EXPLANATION:

See Appendix D.4 for further discussion.

#### Aqueous Phase Density

For non-defaulted aqueous components, the partial molar volume of component k in the aqueous phase at pressure p and temperature T (in absolute units) is given by

$$V_w(k) = \exp[ ct1(k)*(T-TEMR) + ct2(k)*(T*T - TEMR*TEMR)/2 \\ - cp(k)*(p-prsr) - cpt(k)*(p-prsr)*(T-TEMR) ] / den(k)$$

Aqueous phase molar volume  $V_{aq}$  is given by the sum of  $V_w(k) * w(k)$ , k=1 to NUMX, where  $w(k)$  is the mole fraction of component k in the aqueous phase. The aqueous phase molar density is given by  $1/V_{aq}$ .

#### Oil Phase Density

The partial molar volume of component k in the oil phase at pressure p and temperature T (in absolute units) is given by

$$V_o(k) = \exp[ ct1(k)*(T-TEMR) + ct2(k)*(T*T - TEMR*TEMR)/2 \\ - cp(k)*(p-prsr) - cpt(k)*(p-prsr)*(T-TEMR) ] / den(k)$$

Oil phase molar volume  $V_{oil}$  is given by the sum of  $V_o(k) * x(k)$ , k=1 to NUMX, where  $x(k)$  is the mole fraction of component k in the oil phase. The oil phase molar density is given by  $1/V_{oil}$ .

#### Second Temperature Coefficient

Quantity  $ct2(k)$  is an additional coefficient for temperature dependence. The thermal expansion coefficient is  $ct1(k)+T\cdot ct2(k)$  where T is temperature expressed in the corresponding absolute temperature scale (R or K). For example, to specify thermal expansion coefficients 1.0e-4 1/F at 100°F and 2.0e-4 1/F at 500°F, solve

$$ct1(k) + (100+460)\cdot ct2(k) = 1.0e-4$$

$$ct1(k) + (500+460)\cdot ct2(k) = 2.0e-4$$

from which  $ct1(k) = -4e-5$  and  $ct2(k) = 2.5e-7$ . The temperature-dependent part of the density correlation is generated by integrating the thermal expansion expression from TEMR to T:

$$ct1(k)\cdot(T-TEMR) + ct2(k)\cdot(T\cdot T - TEMR\cdotTEMR)/2$$

Using the previous example with TEMR = 60°F = 520°R and T = 300°F = 760°R, the temperature dependent part of the correlation is

$$(-4e-5) \cdot (760-520) + (2.5e-7) \cdot (760 \cdot 760 - 520 \cdot 520) / 2 = 2.88e-2$$

To get this result, the constant model ( $ct2=0$ ) would need  $ct1 = 1.2e-4$  which is between the two values specified above.

Note that the density correlations above need T and TEMR expressed in absolute degrees (R or K) only when  $ct2$  is non-zero. For example, if  $ct2 = 0$  and  $ct1 = 1.2e-4$  then the same result is obtained from the correlation using the F values:  $(1.2e-4) \cdot (300-60) = 2.88e-2$ .

### Dissolved Gases

Gaseous components like methane dissolved in the liquid oil also use the concept of partial molar volume. Many misunderstandings stem from the fact that reservoir engineers commonly do not distinguish between component names and phase names. When working with a compositional simulator like STARS, care must be taken to note the component or fluid as well the phase of interest (e.g., solution gas component in the gas phase versus the same component in the oil phase).

There is a definite distinction between the inverse of partial molar volume (what would be entered via \*MOLDEN or \*MASSDEN) and two other commonly used gas densities that have very different meanings: bulk density and gas phase density. Bulk density is the mass of gas divided by the total oil phase volume, whereas \*MASSDEN is the mass of gas divided by the volume of just the gas component in its liquid (dissolved) form. Gas phase density is the mass of gas divided by its volume in its gaseous form. Usually \*MASSDEN has a value corresponding to specific gravities ranging roughly from 0.3 to 0.7, much like a light liquid. Bulk density usually has a smaller number, and gas phase density is very small. The correct value of \*MASSDEN will result in the correct live oil density after the mixing rule is applied.

### Gassy Liquids

The evolution of solution gas when a live oil drops below the bubble point often can occur sufficiently slowly that a nonequilibrium stage in this process should be modelled directly. In this "foamy oil" situation, small bubbles of gas flow with the oil and contribute to an abnormally high oil phase compressibility. These bubbles are viewed as a "dispersed" component in the oil phase, and the keyword \*GASSYLIQ allows this component to impart a gas-like compressibility to the liquid phase.

It is important when using this keyword that the appropriate component partial molar density corresponding to the reference pressure be employed. Thus when \*PRSR is high, an almost liquid-like density is appropriate, while if \*PRSR is at or near surface pressure, a gas-like partial molar density should be employed.

Most often it is expected that this keyword is used in conjunction with a nonequilibrium mass transfer expression (via the chemical reaction model) which quantifies the rate of creation, and possibly the rate of coalescence, of this gas-like dispersed liquid component.

---

## Liquid Density Nonlinear Mixing

**\*DNMIXFUNC**

**\*DNMIXCOMP, \*DNMIXENDP,**

### PURPOSE:

Specify nonlinear mixing rule for liquid density.

### FORMAT:

**\*DNMIXCOMP comp\_name**  
**\*DNMIXENDP xlow xhigh**  
**\*DNMIXFUNC f(1) ... f(11)**

### DEFINITIONS:

**comp\_name**

Quoted name of component using density nonlinear mixing.

**xlow**

Abscissa corresponding to the first table entry. The allowed range is from 0 to xhigh.

**xhigh**

Abscissa corresponding to the last table entry. The allowed range is from xlow to 1.

**f(i)**

Table entries that define the nonlinear mixing rule function which must be monotonically increasing. The function also should be reasonably smooth in order to minimize convergence difficulties.

### DEFAULTS:

If \*DNMIXCOMP is absent, then linear mixing is assumed for all components.

If \*DNMIXENDP is absent, xlow = 0 and xhigh = 1 are assumed.

If \*DNMIXFUNC is absent, the entries f(i) are equal to (i-1)/10 for i = 1 to 11 which corresponds to linear spacing from 0 to 1.

### CONDITIONS:

The phase to which this data will be assigned depends on which of \*LIQPHASE, \*WATPHASE and \*OILPHASE is in force.

A nonlinear function may be specified for more than one component in each of the water and oil phases. At least one component in each liquid phase must not be a key component, since the algorithm involves adjusting the weighting factors of the non-key components.

Keywords \*DNMIXENDP and \*DNMIXFUNC are applied to the last key component defined via \*DNMIXCOMP. A key component may not be specified more than once in each liquid phase.

## EXPLANATION:

### Density Nonlinear Mixing

The linear mixing rule for liquid density actually is linear in mole volumes

$$V = \text{sum of } V(i) * X(i)$$

where

- V        is molar volume of phase (inverse of phase mole density),  
V(i)     is partial molar volume of component i in the phase,  
X(i)     is mole fraction of component i in the phase.

The nonlinear mixing option replaces the mole fractions with a more general weighting factor vector. When the weighting factors are equal to the mole fractions, the "linear mixing" result is obtained. Keywords \*DNMIXENDP and \*DNMIXFUNC define a function that specifies how to change the weighting factors from "linear".

More exactly, this function  $f(x)$  is the desired weighting factor  $f$  corresponding to the "linear" factor which is the mole fraction  $x$ . For example, consider a medium oil with partial molar volume 100 cc/mole and CO<sub>2</sub> that may dissolved in that oil with a partial molar volume of 200 cc/mole. If the mole fraction of CO<sub>2</sub> in the mixture is  $x = 0.3$  then the linear mixing rule gives a mixture molar volume of

$$\begin{aligned} V &= (1-x) * 100 \text{ cc/mole} + x * 200 \text{ cc/mole} \\ &= 130 \text{ cc/mole} \end{aligned}$$

Suppose that an EOS estimation of oil phase molar volume at the same conditions is 137 cc/mole, reflecting a certain degree of nonlinear volume mixing. If  $f$  is the "nonlinear" weighting factor for key component CO<sub>2</sub> at these conditions, then

$$(1-f) * 100 \text{ cc/mole} + f * 200 \text{ cc/mole} = 137 \text{ cc/mole}$$

giving  $f = 0.37$ . Since this point corresponds to mole fraction 0.3, the nonlinear function has point  $f(0.3) = 0.37$ . The function is filed in for other values of mole fraction of key component CO<sub>2</sub> until 11 points  $f(0.0)$  to  $f(1.0)$  are obtained. These 11 function points are entered for \*DNMIXFUNC. Keyword \*DNMIXENDP allows you to customize the range of mole fractions corresponding to the 11 function points. The precise definition of  $f(x)$  from 11 points  $f(i)$ ,  $i = 1$  to 11, along with  $x_{\text{low}}$  and  $x_{\text{high}}$  is (all interpolation is linear):

$$x(i) = x_{\text{low}} + (i-1)*(x_{\text{high}}-x_{\text{low}})/10, \quad i = 1 \text{ to } 11$$

so that  $x_{\text{low}} = x(1)$  and  $x_{\text{high}} = x(11)$ ;

$0 < x < x_{\text{low}}$ : interpolate between  $f(0) = 0$  and  $f(x_{\text{low}}) = f(1)$

$x(i) \leq x < x(i+1)$ : interpolate between  $f(i)$  and  $f(i+1)$

$x_{\text{high}} < x < 1$ : interpolate between  $f(x_{\text{high}}) = f(11)$  and  $f(1) = 1$ .

The nonlinear mixing option is used in the density calculation as shown above (in the equation used to get the weighting function  $f$  at mole fraction  $x = 0.3$ ), except that the weighting factors are known and the phase mole volume is the result. Since the weighting factors used for all the components must sum to 1, the factors used by the non-key components are multiplied by  $R$  where

wold = sum of mole fractions of key components  
wnew = sum of nonlinear weighting factors of key components  
R = ( 1 - wnew ) / ( 1 - wold )

If wnew or wold is 1 or greater (no non-key components present) then the weighting factors of the key components are normalized to sum to 1.

Nonlinear mixing data should be entered only for truly key components. In the example above with only 2 components, it is true that the same result could be obtained by entering "mirror-image" nonlinear mixing data for the other component. However, designating CO2 as the key component makes it clear what is happening when there is more than 1 other oleic component and when there is more than 1 key component. For example, if the medium oil in the example is split then the nonlinear data is unchanged if CO2 is key.

Example: Two key components 'CO2' and 'Naphtha'

```
*DNMIXCOMP 'CO2' *DNMIXENDP 0 0.2
*DNMIXFUNC
** 0.000 0.020 0.040 0.060 0.080 0.100 0.120 0.140 0.160 0.180 0.200
    0.000 0.025 0.048 0.069 0.091 0.112 0.129 0.145 0.163 0.181 0.200

*DNMIXCOMP 'Naphtha' *DNMIXENDP 0 0.05
*DNMIXFUNC
** 0.000 0.005 0.010 0.015 0.020 0.025 0.030 0.035 0.040 0.045 0.050
    0.000 0.014 0.018 0.022 0.026 0.030 0.034 0.033 0.042 0.046 0.050
```

See Appendix D.4 for further discussion.

## Gas Phase Density (Optional)

\*GASD-ZCOEF, \*GASD-Z-MIN

### PURPOSE:

Control gas phase density calculation.

### FORMAT:

```
*GASD-ZCOEF (*EXPLICIT | *IMPLICIT )
*GASD-Z-MIN Zmin
```

### DEFINITIONS:

#### \*EXPLICIT

Gas compressibility factor  $Z$  is updated at the beginning of each timestep, that is, its treatment is “explicit” in time.

#### \*IMPLICIT

Gas compressibility factor  $Z$  is updated continuously, that is, its treatment is “fully implicit” in time.

#### \*GASD-Z-MIN $Z_{min}$

The minimum allowed value of gas compressibility factor  $Z$  is  $Z_{min}$ . The allowed range of  $Z_{min}$  is 0.001 to 0.27.

### DEFAULTS:

If keyword \*GASD-ZCOEF is absent then \*EXPLICIT is assumed.

If keyword \*GASD-Z-MIN is absent then  $Z_{min} = 0.27$  is assumed.

### EXPLANATION:

Gas phase density is calculated from the Redlich-Kwong equation of state assuming zero interaction coefficients. See Appendix D.4. This calculation reduces to finding the gas compressibility factor  $Z$ . Since the  $Z$  calculation is somewhat expensive and  $Z$  does not vary much under normal circumstances, the default action is that  $Z$  is updated only at the beginning of each timestep (\*EXPLICIT). In some cases a more implicit method is required, so \*IMPLICIT is available. An example of such a case is crossing into and out of water supercritical regions.

---

## Viscosity Type (Optional)

\*VISCTYPE, \*VSTYPE

### PURPOSE:

Define and assign multiple viscosity property types.

### FORMAT:

\*VISCTYPE key (COPY old\_key)

### ARRAY:

\*VSTYPE

### DEFINITIONS:

key

Viscosity property type key. Allowed range is from 1 to 50, inclusive.

Viscosity properties will be assigned to this key number until another \*VISCTYPE is encountered.

\*COPY old\_key

Initialize the set corresponding to 'key' with values from the set corresponding to 'old\_key'. This is useful when you want two viscosity types that are the same except for a few properties.

\*VSTYPE

Enter a viscosity type key for each grid block. Allowed values are 1 and key values that have been defined.

### DEFAULTS:

The default viscosity type key value is 1. \*VISCTYPE is needed only to define multiple viscosity types.

The default key assigned to each block is 1. \*VSTYPE is needed only to assign multiple viscosity type keys to the grid.

### EXPLANATION:

Unless you have multiple viscosity types, you do not need \*VISCTYPE or \*VSTYPE.

The viscosity type can be changed at any time in recurrent data. This can model some of the effects of hysteresis when one set is used for injection and another set is used for production. However, this technique is considered obsolete and the hysteresis keywords in the Rock-Fluid Data section are recommended instead.

The following keywords may be assigned to multiple viscosity types:

\*AVG, \*BVG, \*GVISCOR  
\*AVISC, \*BVISC, \*VISCTABLE  
\*VSMIXCOMP, \*VSMIXENDP, \*VSMIXFUNC  
\*SHEARTHIN \*SHEARTHICK, \*SHEARTAB

Note that \*AVG and \*BVG must appear either for all viscosity types or for none.

## Gas Phase Viscosities

\*AVG, \*BVG, \*TORTIKE\_VG, \*GVISCOR

### PURPOSE:

Override the internal gas phase viscosity with a composition-dependent, and possibly pressure-dependent, calculation.

### FORMAT:

```
*AVG avgi ... avgnumy
*BVG bvgi ... bvgnumy
*TORTIKE_VG comp_name
*GVISCOR
```

### DEFINITIONS:

*avg<sub>i</sub>*, *bvg<sub>i</sub>*

Coefficients in power-law correlation for temperature dependence of gas-phase viscosity of component i. The correlation is

$$\mu_{gi}(T_{abs}) = avg_i * ( T_{abs}^{**} bvg_i ) \text{ where } T_{abs} \text{ is temperature in degrees K or R.}$$

The unit of *avg<sub>i</sub>* is (cp/K<sup>\*\*</sup>*bvg<sub>i</sub>* | cp/R<sup>\*\*</sup>*bvg<sub>i</sub>*); *bvg<sub>i</sub>* is dimensionless. *avg<sub>i</sub>* and *bvg<sub>i</sub>* must be non-negative. *avg<sub>i</sub>* may be zero only if *bvg<sub>i</sub>* = 0 or the component does not appear in the gas phase.

Keywords \*AVG and \*BVG, if present, must be followed by numy values (see \*COMPNAME).

\*TORTIKE\_VG *comp\_name*

For component *comp\_name* use the steam correlation of Tortike et al instead of the power-law correlation. *comp\_name* must be the quoted name of an aqueous component specified via \*COMPNAME. \*TORTIKE\_VG is effective only if the per-component gas-phase viscosity calculation is enabled via \*AVG.

\*TORTIKE\_VG may be specified for any number of your aqueous components, but it must appear once for each component so specified.

\*GVISCOR

Enable correction to gas phase viscosity to account for high gas density.

### DEFAULTS:

If \*AVG or \*BVG is absent, the gas viscosity is independent of composition.

For aqueous component i, if *avg<sub>i</sub>* = *bvg<sub>i</sub>* = 0 is entered then *avg<sub>i</sub>* = 2.3518e-5 cp/K<sup>\*\*</sup>1.075 and *bvg<sub>i</sub>* = 1.075 are assumed.

For non-aqueous component i, if *avg<sub>i</sub>* = *bvg<sub>i</sub>* = 0 is entered then *avg<sub>i</sub>* = 0.01 cp and *bvg<sub>i</sub>* = 0 are assumed.

### CONDITIONS:

Both \*AVG and \*BVG must be present to enable the composition-dependent option.

The \*GVISCOR option is allowed only when \*AVG, \*BVG and \*PCRIT are present, since its calculation uses component critical pressures. \*GVOSCOR is not allowed together with \*MASSBASIS.

If component  $i$  appears in the gas phase,  $avg_i = 0$  is allowed only when  $bvg_i = 0$ .

For multiple viscosity sets (\*VISCTYPE), keywords \*AVG and \*BVG should be either present for all sets or absent for all sets.

### EXPLANATION:

There are two basic options for calculation of gas-phase viscosity. Keywords \*AVG and \*BVG cause the calculation to depend upon gas phase composition; otherwise, it is independent of composition.

#### Independent of Composition

If \*AVG or \*BVG is absent, the following correlation is used for gas-phase viscosity:

$$\mu_g(T_C) = (0.00864 \text{ cp}) \cdot (1.574 + 0.0044 \cdot T_C) \text{ where } T_C \text{ is in deg C.}$$

This correlation is designed for general purpose use and does not correspond to any particular component. Viscosity values at select temperatures are

Temperature (C):	10	100	500
Gas phase viscosity (cp):	0.01398	0.01740	0.03261

Of all the gas viscosity options this method requires the least computation and the accuracy is quite good, so override this default only if necessary. Another consideration is that variations in gas viscosity usually do not significantly affect liquid recover results.

#### Dependent on Composition

If both keywords \*AVG and \*BVG are present then the gas-phase viscosity depends on gas-phase composition. This is accomplished by calculating viscosity  $\mu_{gi}$  for each component separately and mixing these values together using the mixing rule

$$\mu_g = \frac{\sum_{i=1}^{numy} visg(i) * y(i) * \sqrt{cmm(i)}}{\sum_{i=1}^{numy} y(i) * \sqrt{cmm(i)}}$$

The component viscosity  $\mu_{gi}$  is calculated from absolute temperature  $T_{abs}$  using:

$$\mu_{gi}(T_{abs}) = avg_i * (T_{abs} ** bvg_i).$$

#### Steam Viscosity Default

There is a default power-law correlation for steam accessible by specifying  $avg_i = bvg_i = 0$  for an aqueous component. This triggers the internal assignment  $avg_i = 2.3518e-5 \text{ cp/K}^{**} 1.075$  and  $bvg_i = 1.075$ , and the result at selected temperatures is:

Temperature (C):	10	100	500
Power-Law (cp):	0.01017	0.01368	0.02994
Tortike et al (cp):	0.00842	0.01183	0.03565

### Alternate Steam Correlation

For each aqueous component for which \*TORTIKE\_VG is specified, the component viscosity  $\mu_{gi}$  is calculated instead using the pure-water correlation of Tortike et al (Tortike, W.S., Farouq Ali, S.M, "Saturated-Steam-Property Functional Correlations for Fully Implicit Thermal Reservoir Simulator", SPE Res. Eng., November 1989, pp. 471-474.):

$$\mu_{gi}(T_{abs}) = ((( (4.71914d-14 \cdot T_{abs} - 9.9706d-11) \cdot T_{abs} + 8.29842d-8 ) \\ \cdot T_{abs} - 3.39999d-5 ) \cdot T_{abs} + 6.8949d-3 ) \cdot T_{abs} - 0.546807$$

where  $\mu_{gi}$  has unit cp. This correlation is applicable only up to the critical temperature of water, so  $\mu_{gi}(T > T_{cri}) = \mu_{gi}(T_{cri})$  where  $T_{cri}$  is the component's critical temperature specified by \*TCRIT.

### High-Density Correction

It is possible to apply a high-density correction to gas-phase viscosity via keyword \*GVISCOR. This correction was suggested by Dean and Stiel, AIChEJ 1965, vol.11, p. 526 and has the form:

$$(vishp-vislp)C = 1.08 \exp ( 1.439 \text{ denr} - \exp ( -1.111 \text{ denr} ^{**} 1.858 ) )$$

vishp	-	high pressure mixture viscosity [microP]
vislp	-	low pressure mixture viscosity [microP]
denr	-	pseudo reduced mixture density = denm/dencm
denm	-	mixture density [gmol/cm3]
dencm	-	pseudo critical mixture density [gmol/cm3]
C	-	$(T_{cm}^{**1/6}) / (MW_m^{**0.5} * P_{cm}^{**2/3})$
Tcm	-	pseudo critical mixture temperature
Pcm	-	pseudo critical mixture pressure
MWm	-	mixture molecular mass

Pcm and Tcm are calculated by taking mole fraction weighted averages of critical properties and Zcm was also approximated to be 0.27.

See Appendix D.5 for further discussion.

### Example

Specify individual gas-phase viscosities for a combustion tube. Note that numy = 5 and numw = 1. Use Tortike et al correlation for steam, and add high-density correction.

```
*INUNIT FIELD
.
.
*MODEL 6 5 3 1
*COMPNAME 'WATER' 'HEVY OIL' 'LITE OIL' 'INRT GAS' 'OXYGEN' 'COKE'
.
.
*AVG      0      3.926e-6   2.166e-6   2.127e-4   2.196e-4
*BVG      0      1.102      0.943      0.702      0.721
*TORTIKE_VG 'WATER'  ** Tortike et al steam correlation
*GVISCOR          ** High-density correction
```

## Liquid Viscosities (Required)

\*AVISC, \*BVISC, \*VISCTABLE, \*XNACL

### PURPOSE:

Assign liquid viscosities.

### FORMAT:

\*AVISC *avisc<sub>1</sub>* ... *avisc<sub>numx</sub>*  
\*BVISC *bvisc<sub>1</sub>* ... *bvisc<sub>numx</sub>*

or

\*VISCTABLE  
*viscosity\_T\_table*

or

\*VISCTABLE  
{ \*ATPRES *pres*  
*viscosity\_T\_table* }  
\*XNACL *xnac1*

### DEFINITIONS:

*avisc<sub>i</sub>, bvisc<sub>i</sub>*

Coefficients of the correlation for temperature dependence of component viscosity in the liquid phases. The unit of *avisc<sub>i</sub>* is cp (viscosity). The unit of *bvisc<sub>i</sub>* is temperature difference, which has the same value if the temperature unit is C or K and the same value if the temperature unit is F or R.

The correlation for component *i* viscosity  $\mu_{Li}$  is

$$\mu_{Li} = avisc_i \cdot \exp[ bvisc_i / T_{abs} ]$$

where  $T_{abs}$  is in absolute degrees. Neither *avisc<sub>i</sub>* nor *bvisc<sub>i</sub>* may be negative.

See Table 4 for suggested coefficient values for selected components.

For an aqueous component, enter zero to get the internal water table (water phase only). For a component not found in the phase in question, enter zero for *avisc<sub>i</sub>*. For a component with *avisc<sub>i</sub>* > 0 and *bvisc<sub>i</sub>* = 0 (or \*BVISC absent), the result is  $\mu_{Li} = avisc_i$  at all temperatures.

*viscosity\_T\_table*

Table of viscosity versus temperature, with each row in the form

*T visc<sub>1</sub>* ... *visc<sub>numx</sub>*

The maximum allowed number of rows is 40.

When the viscosities do not vary with temperature (e.g., isothermal mode) you may use either (1) *viscosity\_T\_table* with only one row or (2) \*AVISC without \*BVISC.

*T*

Table temperature (C | F). The specified temperature range must be large enough to include all the temperatures encountered during the simulation. A warning message is issued at the end of each timestep for each grid block temperature that lies more than 1 deg C outside this table range. To avoid possible unphysical results and poor convergence, only a certain number of such warning messages are allowed before the simulation is terminated.

*visc<sub>i</sub>*

Viscosity (cp) for each component *i*. A zero value is allowed only for an aqueous component to specify defaulting to internal water data. A warning is issued when viscosity increases with temperature. Such a condition can be allowed for soluble gases dissolved in liquid components (see Appendix D.5) but should not be allowed for the liquid components themselves.

A value must be entered for each of the numx components specified by keyword \*MODEL. For a component not found in the phase in question, enter zero to satisfy the row syntax.

\*ATPRES *pres*

Subkeyword \*ATPRES indicates that there is a *viscosity\_T\_table* for each of several pressures, making liquid viscosity a function of both pressure and temperature. Each *viscosity\_T\_table* may have a different set of *T* entries. Each viscosity type specified via \*VISCTYPE may have a different number of pressure-dependent viscosity tables.

Quantity *pres* is the pressure (kPa | psi) corresponding to the following table and must increase from one \*ATPRES to the next. The allowed range of *pres* is [10<sup>3</sup> kPa, 10<sup>8</sup> kPa]. No extrapolation is done outside the pressure range specified by the first and last \*ATPRES.

*xnacl*

Brine concentration (mass fraction of salt). The allowed range is from 0 to 0.26.

## DEFUALTS:

If \*AVISC is present but \*BVISC is absent, the component viscosities are  $\mu_{Li} = avisc_i$ .

If zero is specified for *visc<sub>i</sub>* at all temperatures for an aqueous component, internal water data will be used for that component in the water phase only. There is no default for aqueous component data in the oil phase, even if the data is assigned with \*LIQPHASE in force.

If \*XNACL is absent, *xnacl* = 0 is assumed.

## CONDITIONS:

Either \*AVISC (and possibly \*BVISC) or \*VISCTABLE must be present, but not both. Once the method of specifying liquid viscosity is decided, that method must be used for all components.

The phase to which this data will be assigned depends on which of \*LIQPHASE, \*WATPHASE and \*OILPHASE is in force. The default is \*LIQPHASE.

## EXPLANATION:

### Component Liquid Viscosity

The viscosity correlation for a component is

$$\mu_{Li} = avisc_i \cdot \exp( bvisc_i / T_{abs} ) \text{ or}$$

$$\ln[\mu_{Li}] = \ln[avisc_i] + bvisc_i / T_{abs}$$

where  $T_{abs}$  is absolute temperature corresponding to temperature T. To use this correlation by hand you must convert T from C to K, or from F to R.

The table option obtains  $\mu_{Li}$  from interpolation of  $visc_i$  between adjacent temperature rows. The interpolation formula has the same form of T dependence as the correlation above, that is,  $\ln[\mu_{Li}]$  varies linearly with  $1/T_{abs}$ . The \*ATPRES option calculates  $\ln[\mu_{Li}]$  from linear interpolation of  $\mu_{Li}(T)$  obtained from adjacent \*ATPRES tables at the same T.

### Phase Mixing Rules

Oil phase viscosity  $\mu_o$  depends on component viscosities  $\mu_{oi}$  and weighting factors  $f_{oi}$  as follows:

$$\ln(\mu_o) = \sum_i [ f_{oi} \cdot \ln(\mu_{oi}) ]$$

Factors  $f_{oi} = x_i$  (oil mole fractions) for linear mixing. Use keywords \*VSMIXCOMP, \*VSMIXENDP and \*VSMIXFUNC for oil phase to specify non-linear mixing, that is, factors  $f_{oi}$  that are different from  $x_i$ .

Water phase viscosity  $\mu_w$  depends on component viscosities  $\mu_{wi}$  and weighting factors  $f_{wi}$  as follows:

$$\ln(\mu_w) = \sum_i [ f_{wi} \cdot \ln(\mu_{wi}) ]$$

Factors  $f_{wi} = w_i$  (water mole fractions) for linear mixing. Use keywords \*VSMIXCOMP, \*VSMIXENDP and \*VSMIXFUNC for water phase to specify non-linear mixing, that is, factors  $f_{wi}$  that are different from  $w_i$ .

### Water/Oil Data Same versus Different

When oil and water phases are given the same oleic component viscosity data via \*LIQPHASE,  $\mu_{oi} = \mu_{wi}$  at all T and p. However, phase viscosities  $\mu_o$  and  $\mu_w$  will be different if the phase weighting factors are different. This is certainly the case when there is no liquid-liquid solubility (no component in both liquid phases).

When an oleic component occurs in water phase and you need  $\mu_{oi} \neq \mu_{wi}$  at some T and p, you must enter separate water phase data via \*WATPHASE. For example, dissolved CO<sub>2</sub> has a very different effect in oil phase versus water phase. When an aqueous component occurs in oil phase, you must enter separate oil phase data via \*OILPHASE since no default data is available.

### Salt Content of Water Phase

\*XNACL is used only with the internal water viscosity option. *xnacl* = 0.25 represents a 25 wt% salt concentration. The water viscosity correction for brine is derived from the SPE monograph "Pressure Buildup and Flow Tests in Wells" by C.S. Matthews and D.G. Russell (1967). See Figure 9 below.

See Appendix D.5 for further discussion.

### Examples

- Tables for three oil components and defaulted water.

```
*INUNIT FIELD
*COMPNAME      'WATER'   'LITE OIL'  'MEDM OIL'  'HEVY OIL'
*VISCTABLE
**          Temp
    75        0        2.328     10.583      5780
   100        0        1.9935    9.061       1380
   150        0        1.4905    6.775       187
   200        0        1.1403    5.183        47
   250        0        0.8896    4.0434     17.4
   300        0        0.7058    3.2082      8.5
   350        0        0.5683    2.5833      5.2
   500        0        0.319     1.4498      2.5
```

- Correlations for two oil components and two non-defaulted water components.

```
*INUNIT FIELD
*COMPNAME 'ORIG WAT' 'COMB WAT' 'HEVY OIL' 'LITE OIL'
*AVISC    .00752    .00752   4.02e-4   4.02e-4
*BVISC    2492.75   2492.75  7830.6    6121.6
```

- Pressure-dependent oil viscosity.

```
*INUNIT FIELD
*COMPNAME      'WATER'   'OIL'
*VISCTABLE
  *ATPRES 40
    75        0        5780
   100        0        1380
   200        0        47
   300        0        8.5
  *ATPRES 70
    80        0        5790
   100        0        1400
   200        0        40
   300        0        8.5
```

- 4) Liquid-liquid solubility; default water component viscosity in water phase; specify values for water component in oil phase; CO2 has different effect in water and oil phases, while Naphtha has the same.

```
*INUNIT FIELD
*COMPNAME 'Water' 'Bitumen' 'CO2' 'Naphtha'
*WATPHASE
*VISCTABLE
** Temp
    70      0.0    296660     640      640
   100      0.0    33500      71       71
   200      0.0    360.7     3.8      3.8
   300      0.0     31.7     1.00     1.00
   400      0.0     7.22     0.540     0.540
   500      0.0     2.71     0.250     0.250
   600      0.0     1.87     0.200     0.200
*oilphase
*visctable
** Temp
    70     10.0    296660    64.0      640
   100      6.8    33500      7.1       71
   200      3.2    360.7     .38      3.8
   300      1.7     31.7     0.100     1.00
   400      1.4     7.22     0.054     0.540
   500      1.1     2.71     0.025     0.250
   600      0.9     1.87     0.020     0.200
```

- 5) Brine modification.

```
*INUNIT FIELD
*COMPNAME 'WATER' 'LITE OIL' 'MEDM OIL' 'HEVY OIL'
*XNACL 0.17 ** 17% salt in brine
*VISCTABLE
** Temp
    75      0     2.328    10.583     5780
   100      0     1.9935    9.061     1380
   200      0     1.1403    5.183      47
   250      0     0.8896    4.0434     17.4
   350      0     0.5683    2.5833     5.2
   500      0     0.319     1.4498     2.5
```

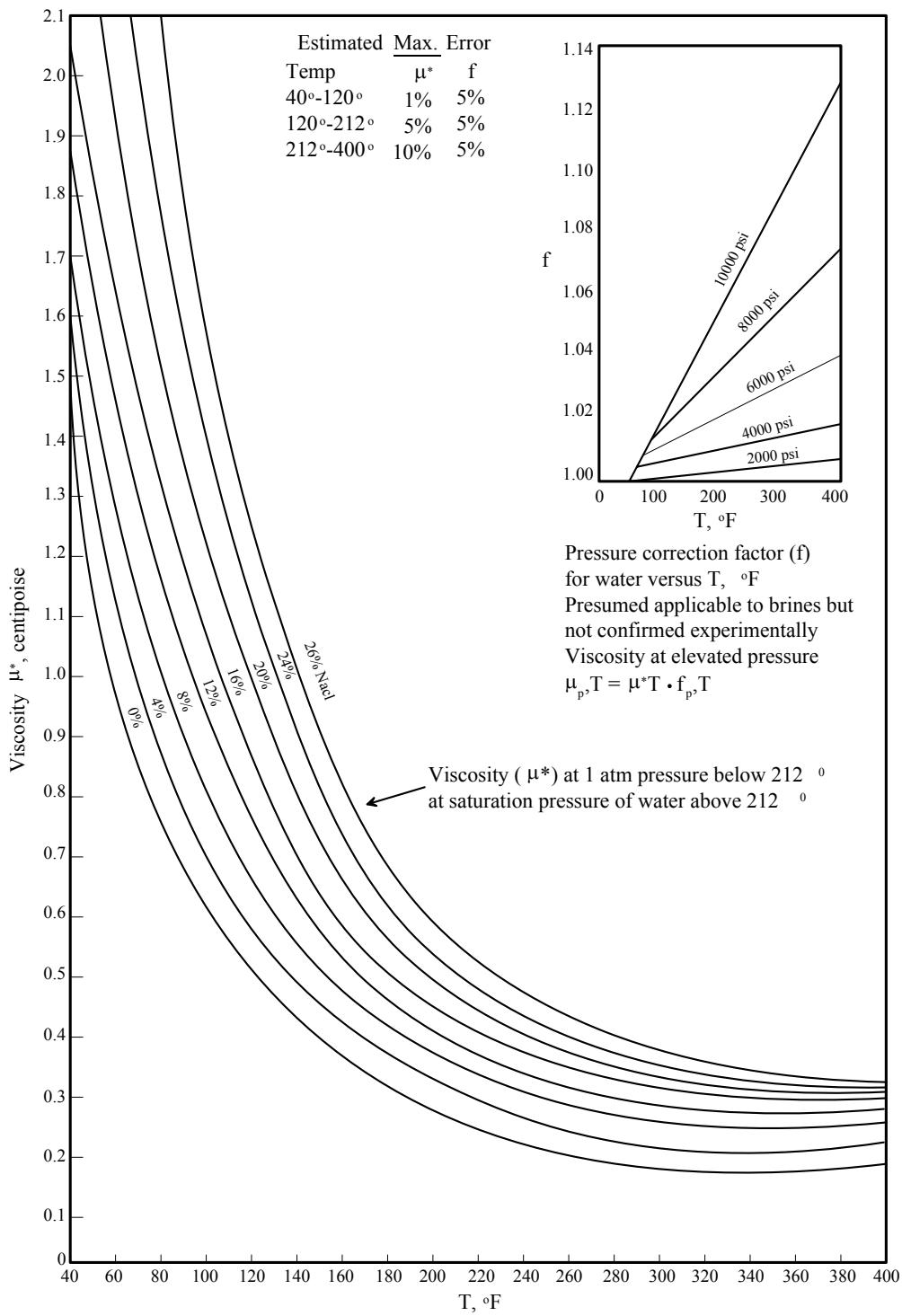


Figure 9: Dependence of Water Viscosity on Salinity

---

## Liquid Viscosity Nonlinear Mixing

**\*VSMIXFUNC**

**\*VSMIXCOMP, \*VSMIXENDP,**

### PURPOSE:

Specify nonlinear mixing rule for liquid viscosities.

### FORMAT:

**\*VSMIXCOMP** *comp\_name*  
**\*VSMIXENDP** *x<sub>low</sub> x<sub>high</sub>*  
**\*VSMIXFUNC** *f<sub>1</sub> ... f<sub>11</sub>*

### DEFINITIONS:

*comp\_name*

Quoted name of component using viscosity nonlinear mixing. This must be one of the component names specified via **\*COMPNAME**.

*x<sub>low</sub>*

Abscissa corresponding to the first table entry. The allowed range is from 0 to *xhigh*.

*x<sub>high</sub>*

Abscissa corresponding to the last table entry. The allowed range is from *xlow* to 1.

*f<sub>1</sub> ... f<sub>11</sub>*

Eleven table entries that define the nonlinear mixing rule function. For good numerical convergence behavior this function should be reasonably smooth.

### DEFAULTS:

If **\*VSMIXCOMP** is absent, linear mixing is assumed for all components.

If **\*VSMIXENDP** is absent, *xlow* = 0 and *xhigh* = 1 are assumed.

If **\*VSMIXFUNC** is absent, entries *f<sub>i</sub>* = (i-1)/10 for *i* = 1 to 11, corresponding to linear spacing from 0 to 1.

### CONDITIONS:

The phase to which this data will be assigned depends on which of **\*LIQPHASE**, **\*WATPHASE** and **\*OILPHASE** is in force.

A nonlinear function may be specified for more than one component in each of the water and oil phases. At least one component in each liquid phase must not be a key component, since the algorithm involves adjusting the weighting factors of the non-key components.

Keywords **\*VSMIXENDP** and **\*VSMIXFUNC** are applied to the last key component defined via **\*VSMIXCOMP**. A key component may not be specified more than once in each liquid phase.

## **EXPLANATION:**

The viscosity nonlinear mixing option works the same way as the density option described with keyword \*DNMIXCOMP, the key difference being that the quantity that is weighted is logarithm of viscosity.

Composition of the viscosity nonlinear mixing key components may be examined via subkeyword \*VISCCMP of \*GRID \*OUTPRN and subkeywords \*VISWCOM and \*VISOCOM of \*OUTSRF \*GRID and \*SPECIAL.

See Appendix D.5 for further discussion.

## **Non-Newtonian Behavior**

In addition to non-linear mixing, non-Newtonian (velocity-dependent) viscosity may be specified via keywords \*SHEARTHIN, \*SHEARTHICK or \*SHEARTAB, for the component and phase specified via \*VSMIXCOMP.

---

## Shear Effects Power Law

\*SHEARTHIN, \*SHEARTHICK

### PURPOSE:

Specify shear thinning and thickening effects using a power law relation.

### FORMAT:

\*SHEARTHIN  $n_{thin}$   $u_{l,lower}$   
\*SHEARTHICK  $n_{thick}$   $u_{l,max}$   $\mu_{l,max}$

### DEFINITIONS:

$n_{thin}$

Power in viscosity shear thinning equation (dimensionless). The allowed range is from 0.1 to 0.99, inclusive. Values below 0.3 can result in unacceptable numerical performance and so are not recommended. Values close to 1 approximate Newtonian behavior.

$u_{l,lower}$

Reference Darcy velocity (m/day | ft/day | cm/min) in viscosity shear thinning equation. The allowed range is  $10^{-5}$  to  $10^3$  m/day ( $3.28 \cdot 10^{-5}$  ft/day to  $3.28 \cdot 10^3$  ft/day |  $6.94 \cdot 10^{-7}$  to  $69.4$  cm/min).

$n_{thick}$

Power in viscosity shear thickening equation (dimensionless). The allowed range is from 1.01 to 5, inclusive. Values above 2.5 can result in unacceptable numerical performance and so are not recommended. Values close to 1 approximate Newtonian behavior.

$u_{l,max}$

Reference Darcy velocity (m/day | ft/day | cm/min) in viscosity shear thickening equation. The allowed range is  $10^{-5}$  to  $10^5$  m/day ( $3.28 \cdot 10^{-5}$  to  $3.28 \cdot 10^5$  ft/day |  $6.94 \cdot 10^{-7}$  to  $6.94 \cdot 10^3$  cm/min).

$\mu_{l,max}$

Maximum viscosity (cp) in viscosity shear thickening equation. The allowed range is  $10^{-5}$  to 1000 cp.

### DEFAULTS:

None.

### CONDITIONS:

These keywords are applied to the component and phase specified via the immediately preceding \*VSMIXCOMP, so \*VSMIXCOMP must be present before these keywords to take effect.

If combined shear thinning and thickening effects are desired then \*SHEARTHICK must be used together with \*SHEARTHIN.

\*SHEARTHICK and \*SHEARTHIN may not be used together with \*SHEARTAB.

## EXPLANATION:

### Shear Thinning Alone

The bounded power law relation between apparent fluid viscosity  $\mu_{app}$  and Darcy fluid velocity  $u_l$  is:

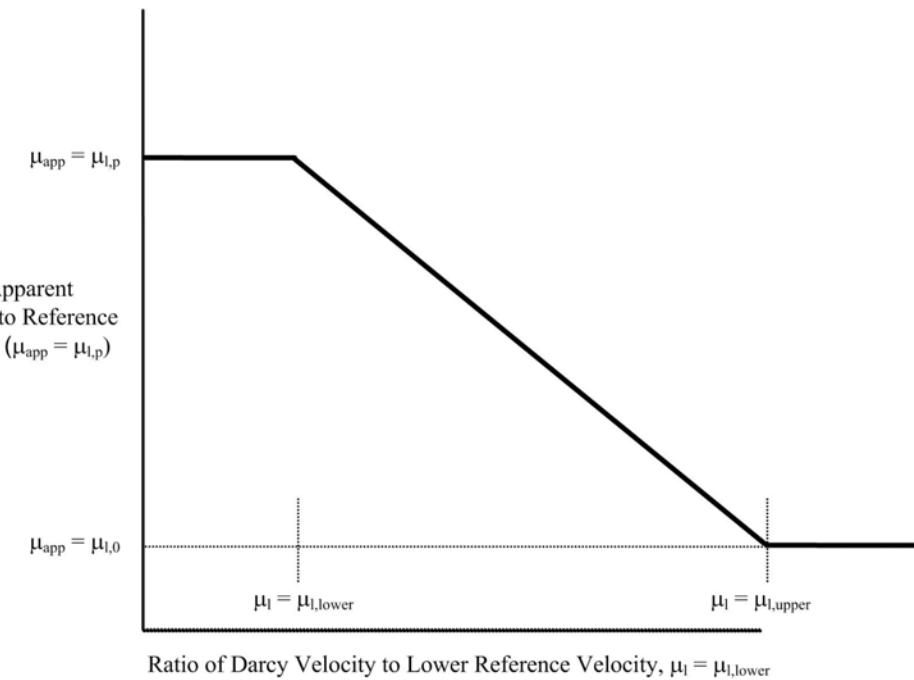
$$\mu_{app} = \mu_{l,p} \quad \text{for } u_l \leq u_{l,lower}$$

$$\mu_{app} = \mu_{l,p} \left[ \frac{u_l}{u_{l,lower}} \right]^{n_{thin}-1} \quad \text{for } u_{l,lower} \leq u_l \leq u_{l,upper}$$

$$\mu_{app} = \mu_{l,0} \quad \text{for } u_l \geq u_{l,upper}$$

The upper velocity boundary of the shear thinning regime  $u_{l,upper}$  is defined by the point on the power law curve when the apparent viscosity  $\mu_{app}$  equals the phase fluid viscosity in the absence of polymer ( $\mu_{l,0}$ ). The lower velocity boundary of the shear thinning regime,  $u_{l,lower}$ , is defined by the point on the power law curve when the apparent viscosity  $\mu_{app}$  equals the fluid phase viscosity in the absence of thinning. For further discussion on the calculation of phase viscosities for Newtonian flow, see the manual page for \*AVISC, etc.

The bounded power law relation of apparent viscosity versus velocity for shear thinning is depicted in the log/log plot of Figure 2. The shear thinning regime is represented by a linear relation of slope ( $n_{thin}-1$ ).



**Figure 2: Shear thinning power law: Apparent viscosity vs. Darcy velocity**

## Shear Thickening Alone

The power law relation between apparent fluid viscosity  $\mu_{app}$  and Darcy fluid velocity  $u_l$  is:

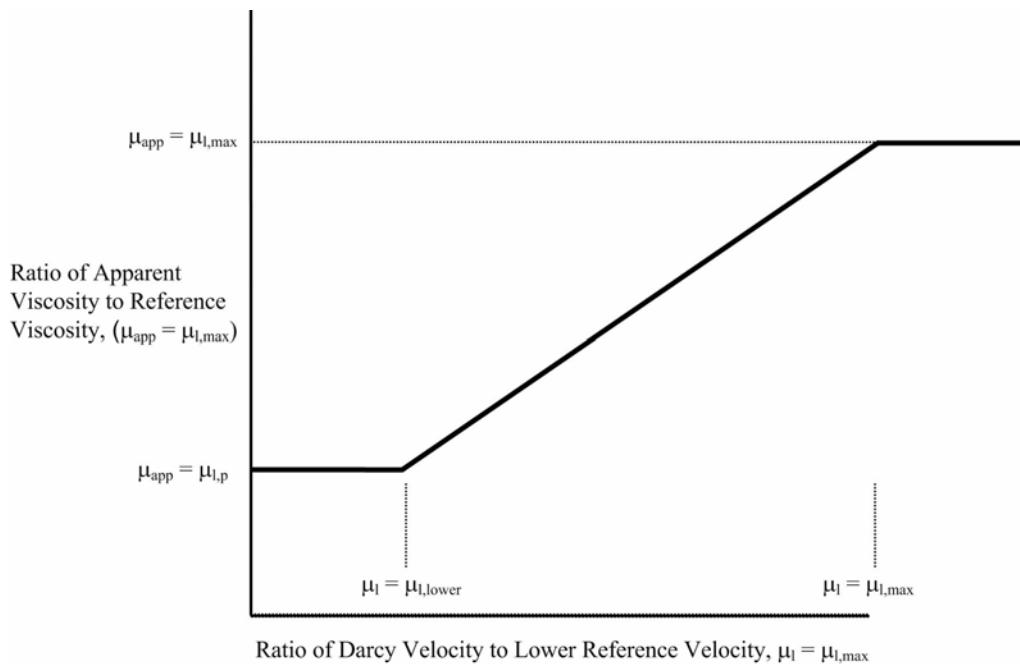
$$\mu_{app} = \mu_{l,p} \quad \text{for } u_l \leq u_{l,lower}$$

$$\mu_{app} = \mu_{l,max} \left[ \frac{u_l}{u_{l,max}} \right]^{n_{thick}-1} \quad \text{for } u_{l,lower} \leq u_l \leq u_{l,max}$$

$$\mu_{app} = \mu_{l,max} \quad \text{for } u_l \geq u_{l,lower}$$

The lower velocity boundary of the shear thickening regime  $u_{l,lower}$  is defined by the point on the power law curve when the apparent viscosity  $\mu_{app}$  equals the phase fluid viscosity in the absence of thickening ( $\mu_{l,p}$ ). For further discussion on the calculation of phase viscosities for Newtonian flow, see the manual page for \*AVISC, etc. The upper velocity boundary of the shear thickening regime  $u_{l,max}$  is defined by the point on the power law curve when the apparent viscosity  $\mu_{app}$  equals the user defined maximum viscosity ( $\mu_{l,max}$ ).

The bounded power law relation of apparent viscosity versus velocity for shear thickening is depicted in the log/log plot of Figure 3. The shear thickening regime is represented by a linear relation of slope ( $n_{thick}-1$ ).



*Figure 3: Shear thickening power law: Apparent viscosity vs. Darcy velocity*

## Combined Shear Thinning and Thickening

Use \*SHEARTHIN and \*SHEARTHICK together to activate combined shear thickening and thinning of fluid viscosity.

The combined shear thinning and thickening apparent viscosity is the sum of the shear thinning and thickening apparent viscosities defined in the sections above,

$$\mu_{app} = \mu_{app,thin} + \mu_{app,thick}$$

The summed power law relation between apparent fluid viscosity  $\mu_{app}$  and Darcy fluid velocity  $u_l$  is:

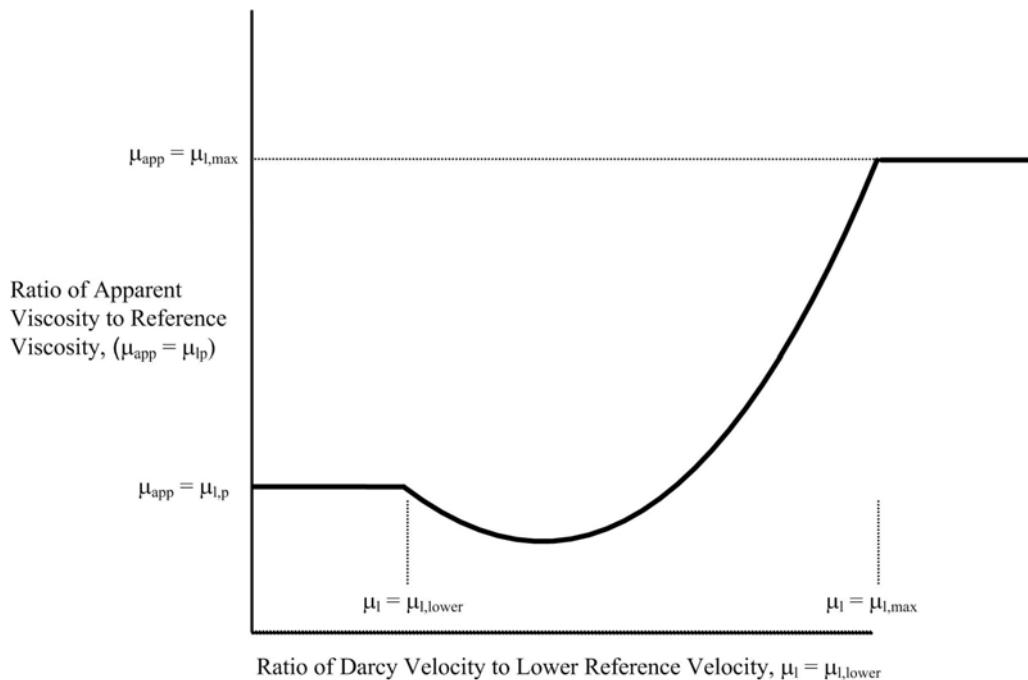
$$\mu_{app} = \mu_{l,p} \quad \text{for } u_l \leq u_{l,lower}$$

$$\mu_{app} = \mu_{app,thin} + \mu_{app,thick} \quad \text{for } u_{l,lower} \leq u_l \leq u_{l,max}$$

$$\mu_{app} = \mu_{l,max} \quad \text{for } u_l \geq u_{l,max}$$

The lower velocity boundary of the shear thinning and thickening regime  $u_{l,lower}$  is defined by the point on the thinning power law curve when the apparent viscosity  $\mu_{app}$  equals the fluid phase viscosity in the absence of thinning. The upper velocity boundary of the shear thinning and thickening regime  $u_{l,max}$  is defined by the point on the thickening power law curve when the apparent viscosity  $\mu_{app}$  equals the user defined maximum viscosity ( $\mu_{l,max}$ ).

The summed power law relation of apparent viscosity versus velocity for shear thickening is depicted in the log/log plot of Figure 4.



**Figure 4: Shear thinning and thickening power laws: Apparent viscosity vs. Darcy velocity**

See appendix D.20 for further discussion.

---

## Shear Effects Table

\*SHEARTAB

### PURPOSE:

Specify shear effects on fluid viscosity using tabular input data.

### FORMAT:

\*SHEARTAB  
{ *velocity* *viscosity* }

### DEFINITIONS:

#### \*SHEARTAB

A viscosity-versus-velocity table follows. The maximum allowed number of table rows is 40.

#### *velocity*

Phase velocity (m/day | ft/day | cm/min). The allowed range is  $10^{-5}$  to  $10^5$  m/day ( $3.28 \cdot 10^{-5}$  to  $3.28 \cdot 10^5$  ft/day |  $6.94 \cdot 10^{-7}$  to  $6.94 \cdot 10^3$  cm/min).

#### *viscosity*

Viscosity (cp) at corresponding *velocity*. The allowed range is  $10^{-5}$  to  $10^6$  cp.

### DEFAULTS:

None.

### CONDITIONS:

\*SHEARTAB is applied to the component and phase specified via the immediately preceding \*VSMIXCOMP, so \*VSMIXCOMP must be present before \*SHEARTAB.

\*SHEARTAB may not be used together with \*SHEARTHICK and \*SHEARTHIN.

### EXPLANATION:

A tabular input option for velocity-dependent viscosity is useful when the viscosity-versus-velocity relation is specified by laboratory data or when a simple power law relation is not sufficient.

For phase velocity outside the *velocity* table range, the nearest *viscosity* table entry is used.

See appendix D.20 for further discussion.

---

## Nonequilibrium Blockage

\*BLOCKAGE, \*SOLIDMIN

### PURPOSE:

Specify nonequilibrium blockage by captured solid (non-fluid) components.

### FORMAT:

```
*BLOCKAGE phase_des (comp_name)
  { effp1t rrsft }
*SOLIDMIN sldmin
```

### DEFINITIONS:

#### \*BLOCKAGE

Table to describe dependence of flow restriction factor on effective permeability. Enter one set of effp1t versus rrsft on each line.

#### phase\_des

Phase to which the flow restriction will be applied:

'W' - water phase,

'O' - oil phase,

'G' - gas phase,

'WG' – water and gas phases,

'ALL' - water, oil and gas phases.

#### comp\_name

Quoted name of component whose captured concentration causes the flow restriction to vary.

#### effp1t

Tabular value of permeability (md). It must be greater than zero.

#### rrsft

Flow restriction factor for the captured component ( $\text{m}^3/\text{gmol}$  |  $\text{ft}^3/\text{lbmol}$  |  $\text{cm}^3/\text{gmol}$ ). It must be positive.

#### sldmin

Minimum solid concentration needed in order for blockage to start ( $\text{gmol}/\text{m}^3$  |  $\text{lbmol}/\text{ft}^3$  |  $\text{gmol}/\text{cm}^3$ ).

### DEFUALTS:

If \*BLOCKAGE is absent, rrsft = 0 is assumed.

If comp\_name is absent, then component number numy+1 is assumed.

If \*SOLIDMIN is absent, sldmin = 0 is assumed.

## **CONDITIONS:**

This option is effective only if there is a solid (non-fluid) component, that is, ncomp > numy.

## **EXPLANATION:**

### **Nonequilibrium Blockage**

Particles captured by the porous medium can cause permeability reductions (blockage) in a manner similar to equilibrium mass transfer to the rock (adsorption).

If the captured droplet is assumed to come from the oil phase then the oil phase effective permeability is

$$(\text{absolute perm}) \cdot (\text{oil relative perm}) / R_{fo}$$

where

$$R_{fo} = \prod_j [ 1 + RRSFT_j \cdot \max(0, C_{sj} - sldmin) ]$$

in a manner similar to (equilibrium) adsorption blockage. Here  $R_{fo}$  is the product of the resistance factor of each blocking component  $j$ ,  $C_{sj}$  is the concentration of captured oil droplets, and  $RRSFT_j$  is looked up from the \*BLOCKAGE table. The minimum solid concentration for blockage to start is given by  $sldmin$ . If  $C_{sj}$  is less than  $sldmin$ , no blockage occurs. If the captured droplet comes from the water or gas phases then the phase effective permeability is modified analogously by an  $R_{fw}$  or  $R_{fg}$ .

See Appendix D.7 for further discussion.

---

## Mandatory Chemical Reaction Data

\*STOREAC, \*STOPROD,

\*FREQFAC, \*FREQFACP

### PURPOSE:

Assign reaction data that is mandatory when reactions are used.

### FORMAT:

\*STOREAC sto1(1) ... sto1(ncomp)  
\*STOPROD sto2(1) ... sto2(ncomp)

\*FREQFAC rrf  
or  
\*FREQFACP  
{ p\_rrf rrf }

### DEFINITIONS:

sto1

Stoichiometric coefficient of reacting component. It must be non-negative.  
Enter zero for components which are not reacting.

Normally, the stoichiometric coefficients are based on one mole of one of the reacting components.

sto2

Stoichiometric coefficient of produced component. It must be non-negative.  
Enter zero for components which are not being produced in this reaction.

Normally, the stoichiometric coefficients are based on one mole of one of the reacting components.

rrf

Reaction frequency factor (unit is variable). It must be non-negative.

This is the constant factor in the expression for reaction rate (see below). The unit depends upon data entered via \*STOREAC, \*RORDER, \*O2PP & \*O2CONC.

p\_rrf

Pressure (kPa | psi) corresponding to frequency factor rrf in table. p\_rrf entries must be increasing and evenly spaced. Interpolation of rrf between p\_rrf entries is linear. For pressures outside the table range, rrf corresponding to the closest p\_rrf is used. No more than 30 table rows are allowed.

Use \*FREQFACP only to provide reaction rate dependence on pressure beyond that naturally occurring through concentration factors via phase densities. When it is used, keyword \*FREQFACP replaces keyword \*FREQFAC on a per-reaction basis.

## **DEFAULTS:**

A component with sto1 = 0 will not react, and a component with sto2 = 0 will not be produced by that reaction. A reaction with rrf = 0 will have zero reaction rate, and will not react or produce any components indicated by \*STOREAC and \*STOPROD.

## **CONDITIONS:**

\*STOREAC, \*STOPROD, and either \*FREQFAC or \*FREQFACP, MUST be entered for each chemical reaction or nonequilibrium mass transfer.

Reaction number is not specified explicitly, but is inferred from the sequence and number of sets of critical reaction keywords. The appearance of another one of the critical reaction keywords will cause the reaction number to incremented. Therefore, all the keywords (both critical and non-critical) associated with each reaction must appear together as a group.

Example of Correct Data Entry:

```
*STOREAC .... ** Indicates first reaction
*STOPROD .... ** Still first reaction
*FREQFAC .... ** "
*RENTH, etc. ** "

*STOREAC .... ** Reaction # incremented to 2
*STOPROD .... ** Still second reaction
*RENTH, etc. ** Still second reaction
*FREQFAC .... ** "
```

Example of Incorrect Data Entry:

```
*STOREAC .... ** First reaction
*STOREAC .... ** Second reaction. Wrong!
No *STOPROD and *FREQFAC for
first reaction
```

## **EXPLANATION:**

For each reaction, it is assumed that each reactant reacts in a particular phase. Sometimes one of the components reacts in more than one phase; this must be modelled as two separate reactions. For example, the burning of an oil component in both the liquid and gas phases must be entered as two reactions. The stoichiometry coefficients will be the same, but the reaction kinetics (rate parameters) may be different. Usually, oil burning is assumed to be a reaction between liquid oil and gaseous oxygen; the reaction kinetics and enthalpy accounts for oil vapourization or oxygen dissolution, which in fact may be the rate-determining processes.

The chemical reaction model can be used to model the type of nonequilibrium mass transfer processes that are involved in the in-situ generation or coalescence of emulsions and foams.

A typical set of combustion reactions, is:

1. Heavy oil cracking to form light oil and solid coke,
2. Coke burning to form water and carbon oxides,
3. Light oil burning,
4. Heavy oil burning.

## **Stoichiometric Mass Balance**

The user is responsible for ensuring that the stoichiometric coefficients entered as data represent a mass-conserving set. A set of mass-conserving coefficients will satisfy

$$\text{sum of } cmm(i)*sto1(i) = \text{sum of } cmm(i)*sto2(i), \text{ sums over } i = 1 \text{ to ncomp}$$

where  $cmm(i)$  is the fluid component molecular mass entered via \*CMM.

## **Stoichiometric Volume Balance**

The value of  $cncco$  (\*SOLID\_DEN) normally will be derived from a coke mass density and \*CMM. If the coke mass density is much different from the heavy oil mass density, the products of the cracking reaction will have a volume significantly different from the reactants. If there is little or no gas present, the total system compressibility is small which can cause large pressure changes. Therefore, in some cases it is important to minimize volume changes due to reactions. A volume conservation constraint would be similar to the mass conservation constraint, with molar volume (inverse of mole density) of component  $i$  in phase  $iph(i)$  replacing  $cmm(i)$ .

## **Proportionality of Reaction Parameters**

Quantities \*RENTH, \*STO1, \*STO2 and reaction rate are proportioned to an amount of material involved, usually one mole of one of the reactants.

Example: Coke-burning reaction. If \*STO1 of coke component is 1, then

- a) \*RENTH is energy released per mole of coke burned,
- b) Reaction rate is the rate of disappearance of moles of coke,
- c)  $sto1(\text{numy})$  is moles of oxygen required to burn one mole of coke,
- d)  $sto2(i)$  is moles of product  $i$  from burning one mole of coke.

## **Reaction Keywords**

The data for the following keywords will be associated with a particular reaction number:

1. Mandatory Reaction Keywords: \*STOREAC, \*STOPROD, \*FREQFAC, \*FREQFACP
2. Thermal Reaction Keywords: \*RENTH, \*EACT, \*EACT\_TAB, \*RTEMLOWR, \*RTEMUPR
3. Non-thermal Reaction Keywords: \*RPHASE, \*RORDER, \*O2PP, \*O2CONC, \*RXCRITCON, \*RXCMPFAC
4. Generalized Reactions: \*PERMSCALE, \*MTVEL
5. Partial Equilibrium Reactions: \*RXEQFOR, \*RXEQBAK

## **Solid Components**

If there is at least one solid component then there must be at least one reaction, otherwise that component's moles will not be conserved. See Appendix F.5 and F.8. If you have a data set with a solid component and associated reactions, you can disable those reactions by setting their \*FREQFAC to zero.

See Appendices D.13, D.14 and F.8 for further discussion.

## Thermal Chemical Reaction Data

\*RENTH, \*EACT, \*EACT\_TAB,

\*RTEMLOWR, \*RTEMUPR

### PURPOSE:

Assign thermal chemical reaction and nonequilibrium mass transfer data.

### FORMAT:

*RENTH	$H_r$
*EACT	$E_a$
or	
*EACT_TAB	<i>ref</i>
{ $T$ $E_a$ }	
*RTEMLOWR	$T_{lwr}$
*RTEMUPR	$T_{upr}$

### DEFINITIONS:

$H_r$

Reaction enthalpy  $H_r$  (J/gmol | Btu/lbmol) is positive for exothermic reactions and negative for endothermic reactions. The default is 0. Reaction enthalpy is often quoted on a per mass basis or using kJ or kBtu, so ensure that  $H_r$  is converted to the required units.

Reaction enthalpy is referenced to temperature TEMR and the enthalpy base phase given by the choice of CPL's and CPG's entered elsewhere. In most cases, the reaction enthalpy is based on gas phase at 25°C.

\*EACT  $E_a$

Single activation energy  $E_a$  (J/gmol | Btu/lbmol) gives the dependence of reaction rate on grid block temperature. For chemical reactions (e.g., combustion)  $E_a$  is positive, that is, the reaction rate increases with increasing T. Reaction rate is independent of T when  $E_a = 0$ . Negative  $E_a$  is allowed, with a warning, to accommodate advanced options like non-equilibrium interphase mass transfer. Activation energies are often quoted in cal/gmole, so ensure that  $E_a$  is converted to the required units.

The temperature-dependent reaction rate factor is  $R_T(T_a) = \exp[-E_a/R \cdot T_a]$  where R is the gas constant and  $T_a$  is block temperature T converted to absolute degrees. See **Single Activation Energy** in the EXPLANATION.

\*EACT\_TAB *ref*

Specify reaction rate temperature-dependence factor with multiple activation energy values, one for each temperature interval. Reference row number *ref* is an integer with allowed range from 1 to number of table rows. See **Multiple Activation Energies**, below.

{ T E<sub>a</sub> }

Table of activation energy  $E_a$  versus temperature T (C | F). At most 20 rows are allowed. See \*EACT for the description of  $E_a$ . Temperature entries must increase down the table by at least 0.01 degrees.

T<sub>lwr</sub>

Lower limit of burning zone temperature (C | F) used in the calculation of T-dependent reaction rates. If the grid block temperature is less than  $T_{lwr}$ , then  $T_{lwr}$  is used as burning zone temperature. The suggested range of  $T_{lwr}$  is from 280 K to 2000 K.

This lower limit can be used to ensure that there is vigorous combustion in a field-scale grid block, independent of the grid block average temperature.

T<sub>upr</sub>

Upper limit of burning zone temperature (C | F) used in the calculation of T-dependent reaction rates. If the grid block temperature is greater than  $T_{upr}$ , then  $T_{upr}$  is used as burning zone temperature. The minimum value allowed is equal to  $T_{lwr}$ . The suggested maximum value of  $T_{upr}$  is 2000 K.

This upper limit can be used to ensure that the reaction temperature and hence reaction rate does not get too large. A large burning reaction rate corresponds to nearly complete oxygen utilization. This results in very small amounts of unburned oxygen, which can cause stability problems. Therefore, decreasing  $T_{upr}$  may increase numerical stability in some combustion simulations, especially if activation energy was increased to raise the reaction rate at low temperatures.

## DEFAULTS:

If \*RENTH is absent,  $H_r = 0$  is assumed.

If both \*EACT and \*EACT\_TAB are absent, the reaction is independent of temperature (equivalent to \*EACT with  $E_a = 0$ ).

If RTEMLOWR is absent,  $T_{lwr} = 7$  C (44 F).

If \*RTEMUPR is absent,  $T_{upr} = 1727$  C (3140 F).

## CONDITIONS:

All these keywords are optional. They are assigned to the current reaction number which is determined by their position relative to the critical reaction keywords \*STOREAC, \*STOPROD and \*FREQFAC.

## EXPLANATION:

### Single Activation Energy

When natural log of reaction rate is plotted versus  $1/(R \cdot T_a)$  the points often cluster around a straight line. This suggests the Arrhenius model

$$R_T(T_a) = \exp[-E_a/R \cdot T_a] \text{ or}$$

$$\ln[R_T(T_a)] = -E_a/R \cdot T_a$$

where  $-E_a$  is the slope of a straight line on a plot of  $\ln[R_T(T_a)]$  versus  $1/R \cdot T_a$ . Activation energy  $E_a$  can be regarded as a fitting parameter, whereby the temperature dependence of the reaction rate is reduced to a single number. Use keyword \*EACT to enter a single  $E_a$  value.

### Multiple Activation Energies

Sometimes a modelled reaction has different rate responses at different temperatures, in which case a plot of natural log of reaction rate versus  $1/(R \cdot T_a)$  will not be a straight line. However, sometimes this plot can be approximated adequately by a series of joined straight lines, each with its own temperature range and slope. This can be modelled with a table of  $E_a$  versus  $T$ , where each temperature range uses the same model form as \*EACT.

Let the \*EACT\_TAB table values be  $(T_{ai}, E_{ai})$ ,  $i = 1$  to  $N$ , where user input temperature values  $T_i$  have been converted to absolute degrees  $T_{ai}$ . Function  $R_T(T_a)$  could be defined as

$$\ln[R_T(T_a)] = -E_{ai}/R \cdot T_a \quad \text{for } T_a \text{ in interval } [T_{ai}, T_{ai+1}], i = 1 \text{ to } N-1$$

However, this function is not continuous. Making it continuous (adjacent intervals  $i$  and  $i+1$  give the same result at  $T_{i+1}$ ) requires adjusting each line segment up or down an amount  $A_i$  to match its neighbours, suggesting the form

$$\ln[R_T(T_a)] = A_i - E_{ai}/R \cdot T_a \quad \text{for } T \text{ in interval } [T_{ai}, T_{ai+1}], i = 1 \text{ to } N-1$$

$$A_i - E_{ai}/R \cdot T_{ai+1} = A_{i+1} - E_{ai+1}/R \cdot T_{ai+1} \quad \text{for } i = 1 \text{ to } N-1$$

It only remains to fix  $A_i$  for one interval since all the other  $A_i$  values can be calculated from it. It would be convenient to specify a reference interval "m" with  $A_m = 0$ , so that the table-based function matches the result from \*EACT if  $E_{am} = E_a$ . This reference layer is specified by integer *ref* after keyword \*EACT\_TAB.

Outside of the table domain  $[T_{a1}, T_{aN}]$ ,  $E_{a1}$  applies below  $T_{a1}$  and  $E_{aN}$  applies above  $T_{aN}$ .

$$\ln[R_T(T_a)] = A_1 - E_{a1}/R \cdot T_a \quad \text{for } T_a < T_{a1}$$

$$\ln[R_T(T_a)] = A_N - E_{aN}/R \cdot T_a \quad \text{for } T_a > T_{aN}$$

Note that the only usage of  $E_{aN}$  is for  $T_a > T_{aN}$ . Note also that  $T_1$  is somewhat arbitrary since  $E_{a1}$  is used both above and below it.

A one-row table for \*EACT\_TAB is converted internally to \*EACT with  $E_a = E_{a1}$ . The following two data fragments give the same result.

\*EACT 53500

\*EACT\_TAB 1  
400 53500

You can add more table entries while keeping the same reference  $E_a$ . This is useful when building a more complex table from simpler, well-known data, since other factors in the reaction rate (e.g., \*FREQFAC) would not need immediate adjustment. For example, the following data keeps  $R_T(T)$  the same as the previous data from  $300^\circ$  to  $600^\circ$  but reduces  $E_a$  in steps above and below that temperature.

*EACT_TAB	3
150	15000
200	25000
300	53500
600	40000
700	20000

\*\* Reference Eact

The following example shows graphically the relationship between \*EACT\_TAB and \*EACT. Consider the two following two data fragments.

```

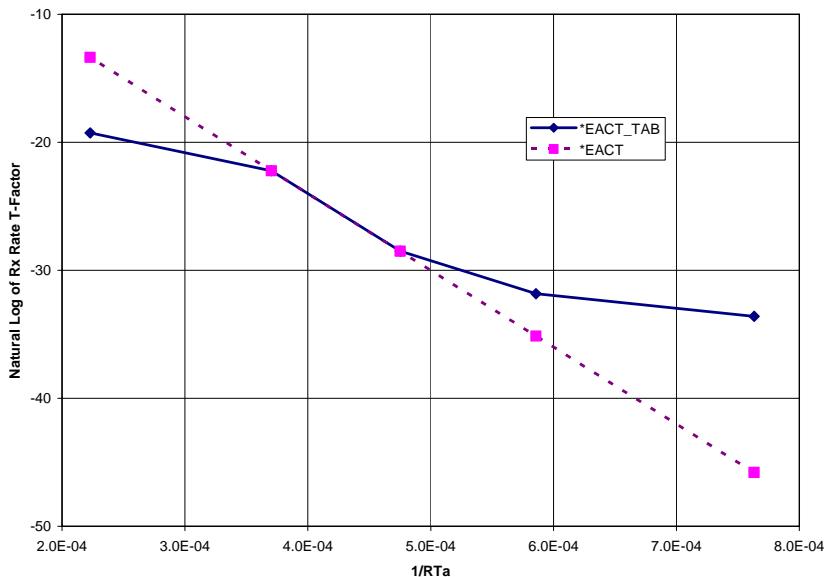
*EACT 60000
*EACT_TAB 3
200 10000
400 30000
600 60000
900 20000
1800 30000

```

The .out file echo for reactions shows the following summary. The first two columns contain the \*EACT\_TAB input data, and F(T) is  $R_T(T_a)$ . Each formula corresponds to a temperature interval. The reference interval's formula is the same as for \*EACT, but all other intervals have an adjusting factor that makes the function continuous over the entire T range.

Temper. (F)	Activation Energy Btu/lbmole	Absolute Temper. (R)	F(Tabs)	F(T)	Formula in T Interval
200.00	10000	659.67	2.540E-15	5.251E-12 * EXP(-10000/R*T)	
400.00	30000	859.67	1.500E-14	5.251E-12 * EXP(-10000/R*T)	
600.00	60000	1059.67	4.136E-13	6.4319E-7 * EXP(-30000/R*T)	
900.00	20000	1359.67	2.233E-10	EXP(-60000/R*T) <- Reference interval	
1800.00	30000	2259.67	4.2686E-9	3.6814E-7 * EXP(-20000/R*T)	
				3.4186E-6 * EXP(-30000/R*T)	

The following plot shows graphically that the reference interval of \*EACT\_TAB gives the same temperature dependence as \*EACT, for the same value of activation energy.



Compare \*EACT\_TAB and \*EACT

---

## **Non-thermal Chemical Reaction Data**

**\*RPHASE, \*RORDER, \*O2PP,  
\*O2CONC, \*RXCRITCON, \*RXCMPFAC**

### **PURPOSE:**

Assign non-thermal chemical reaction and nonequilibrium mass transfer data.

### **FORMAT:**

<b>*RPHASE</b>	Iphas(1) ... iphas(ncomp)
<b>*RORDER</b>	Enrr(1) ... enrr (ncomp)
<b>*O2PP</b>	( ‘component name’ )
<b>*O2CONC</b>	( ‘component name’ )
<b>*RXCRITCON</b>	‘component name’ crit_conc
<b>*RXCMPFAC</b>	‘component name’ phase A B

### **DEFINITIONS:**

#### **iphas**

Flag defining phase for reacting component. The allowed range is 0 to 4.

- = 0 non-reacting components,
- = 1 water phase (fluid components only)
- = 2 oil phase (fluid components only)
- = 3 gas phase (fluid components only)
- = 4 solid phase (solid components only)

Note that an adsorbing component may not react in the adsorbed phase.

#### **enrr**

Order of reaction with respect to each reacting component's concentration factor. It must be non-negative.

Enter zero for non-reacting components. Normally, enrr = 1. If enrr = 0, the reaction rate will be independent of that component's concentration.

#### **\*O2PP ( ‘component name’ )**

Partial pressure in the gas phase is used for the concentration factor of the indicated component in the reaction's rate expression. If ‘component name’ is absent, then component number numy (usually oxygen) is assumed. This option may be applied only to components reacting in the gas phase.

#### **\*O2CONC ( ‘component name’ )**

Mole (or mass) density is used for the concentration factor of the indicated component in the reaction's rate expression. If ‘component name’ is absent, then component number numy (usually oxygen) is assumed.

## **crit\_conc**

Critical value of reactant's concentration factor, below which the dependence of reaction rate on the concentration factor is linear. The unit is (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>) if the factor is a mole density, and is (kPa | psi) if the factor is a pressure.

Use this option only when the reactant's \*RORDER is less than 1 and its concentration is expected to approach zero. Use a value 3 to 6 orders of magnitude below the significant operating range of the quantity in question. For example, for an oil component with density 0.1 lbmol/ft<sup>3</sup>, use 1e-5.

## **\*RXCMPFAC 'component name' phase A B**

Reaction rate is divided by factor  $(1+A \cdot x)^{**B}$  where x is the component's mole fraction in the indicated *phase* (W for water, O for oil, G for gas). Both A and B must be non-negative. A value of zero for either A or B disables the option. This keyword is allowed at most once for each reaction.

## **DEFUALTS:**

If \*RPHASE is absent, the assumption is:

iphas = 0 for non-reacting components,  
iphas = 1 for aqueous components 1 to numw,  
iphas = 2 for oleic components numw+1 to numx,  
iphas = 3 for noncondensable components numx+1 to numy, and  
iphas = 4 for solid components numy+1 to ncomp.

If \*RORDER is absent, the assumption is:

enrr = 0 for non-reacting components, and  
enrr = 1 for reacting components.

If both \*O2PP and \*O2CONC are absent for a component, the assumption is:

\*O2PP for component number numy reacting in the gas phase, and  
\*O2CONC otherwise.

If \*RXCMPFAC is absent, no factor is applied.

## **CONDITIONS:**

All these keywords are optional. They are assigned to the current reaction number which is determined by their position relative to the critical reaction keywords \*STOREAC, \*STOPROD and \*FREQFAC.

## **EXPLANATION:**

### **Reaction Kinetics**

A reaction's kinetics provides information on the speed with which the reaction is proceeding. The expression for volumetric reaction rate is the following series of factors:

$$\text{rrf} * \exp(-E_a/(T * R)) * c(1)^{**\text{enrr}(1)} * \dots * c(ncomp)^{**\text{enrr}(ncomp)}$$

where

rrf	- constant part of the expression,
$E_a$	- activation energy, which provides the temperature dependence,
T	- temperature, constrained by $T_{lwr}$ and $T_{upr}$ ,
R	- universal gas constant,
c(i)	- concentration factor contributed by reactant component i, and
enrr(i)	- order of reaction with respect to component i.

The concentration factor for component i in a fluid phase is usually based on density (by default or flagged by keyword \*O2CONC):

$$c(i) = \varphi_f * \text{den}(\text{iphas}(i)) * \text{sat}(\text{iphas}(i)) * x(\text{iphase}(i), i).$$

$$\varphi_f = \varphi_v * \left[ 1 - \sum C_{sk} / \rho_{sk}(p, T) \right]$$

Here void porosity  $\varphi_v$  is corrected for pore pressure and temperature, and fluid porosity  $\varphi_f$  is corrected for the volume of the solid phase in the pore space. Each component k in the solid phase has concentration  $C_{ck}$  and density  $\rho_{sk}(p, T)$  (see keyword \*SOLID\_DEN).

The second and third factors are the mole density and, saturation respectively, of the fluid phase in which the component I is reacting. The last factor is the mole fraction of component I in the fluid phase in which the component I is reacting.

The solid phase concentration factor for component I is

$$c(i) = \varphi_v * C_{ci}$$

where por is the pore space porosity, which has been corrected for pore pressure and temperature only, and cncc is the current solid concentration in the pore space.

Optionally, the concentration factor for component i in a fluid phase may be based on partial pressure (by default or flagged by keyword \*O2PP):

$$c(i) = y(i) * p_g$$

where y(i) is gas mole fraction of component i, and  $p_g$  is the gas phase pressure.

The temperature normally used in conjunction with  $E_a$  is the grid block temperature. In field scale, this may not correspond to the temperature of the burning zone, resulting in a burn rate which is unrealistically low, or premature extinction. An assumed range for burning zone temperature may be specified using  $T_{lwr}$  and  $T_{upr}$ .

Example: The following describes how a set of chemical reaction data would look.

A chemical reaction in the six-component model

H<sub>2</sub>O - water

HO - heavy oil

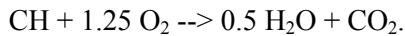
LO - light oil

IG - inert gas, includes nitrogen and carbon oxides

O<sub>2</sub> - oxygen

CH - coke fuel.

might be the burning of heavy oil in the liquid phase



The reaction enthalpy is 6.3e5 J/gm mole, and the activation energy is 53,500 J/gmole. The reaction rate is

$$1.45\text{e}5 * \exp(-53,500/R/T) * (\text{por} * \text{cncc(ch)}) * (\text{y(o2)} * p_g)$$

and has units (gmole/m<sup>3</sup>-day). The oxygen partial pressure option is being used. The unit of the frequency factor RRF is 1/day-kPa, because

- a) Pressure  $P_g$  has unit kPa,
- b) The CH concentration factor has unit (gmole/m<sup>3</sup>), and
- c) The result is (gmole/day-m<sup>3</sup>).

A data for this example reaction is:

```
**          H2O    HO    LO    IG    O2    CH
*STOREAC    0      0      0      0    1.25   1
*STOPROD    .5     0      0      1      0      0
*FREQFAC  1.45e5  ** units are 1/day-kPa
*EACT  53,500
*RENTH 6.3e5
```

Note that \*RPHASE, \*RORDER, \*O2PP, \*RTEMLOWR and \*RTEMUPR were defaulted.

### Lower Reaction Order

A warning is issued when enr from \*RORDER is less than 1 and keyword \*RXCRITCON is not used, since numerical stability may be compromised when the component's concentration approaches zero. This stability concern stems from the fact that the derivative of reaction rate with respect to mole fraction is  $d/(x^{*\text{a}})/dx = ax^{*(\text{a}-1)}$  which for  $\text{a} < 1$  is unbounded as x approaches zero.

### Maximum Solid Phase Volume

Since void porosity  $\varphi_v$  contains both the solid and fluid phases, the fraction of void space occupied by solid phase,

$$\sum C_{sk} / \rho_{sk}(p, T),$$

must not exceed 1. This constrains the total amount of solid phase present in a block, including solid components and adsorbed/trapped fluid components.

Normally the dependence of reaction rate on fluid concentration (and hence  $\varphi_f$ ) naturally prevents  $\varphi_f$  from going negative. However, even a modest amount of numerical over shoot during convergence can produce negative  $\varphi_f$ .

In addition, some types of reactions are able to produce solid components even as the fluid porosity decreases to zero, such as when reaction rate depends upon partial pressure instead of concentration.

When  $\varphi_f$  approaches zero due to increasing solid volume, reaction rates are reduced to preserve stoichiometry as well as satisfy the constraint that  $\varphi_f$  is positive. See Appendix F.8 for details.

---

## Generalized Reactions

\*PERMSCALE, \*MTVEL

### PURPOSE:

Specify the dependence of chemical reactions and nonequilibrium mass transfer on permeability or phase velocity.

### FORMAT:

```
*PERMSCALE
{ effpt freqt }
*MTVEL phase_des exp vref (vcrit)
```

### DEFINITIONS:

#### \*PERMSCALE

Table to describe dependence of reaction or mass transfer rate on permeability. Enter one set of effpt versus freqt on each line.

effpt

Effective permeability (md). It must be greater than zero.

freqt

Reaction rate scaling factor. The allowed range is from 0 to 10,000.

#### \*MTVEL

Indicates that the rate of mass transfer described by the current reaction has the following dimensionless velocity dependent factor:

$$((V - vcrit) / vref)^{\text{exp}}$$

phase\_des

Indicates for which phase the velocity factor applies. The allowed choices are W for water, O for oil and G for gas.

exp

Exponent in the velocity factor. The allowed range is -4 to +4. A value of 1 will result in a linear dependence. A zero value will disable the factor.

vref

Reference velocity for the reaction rate (m/day | ft/day | cm/min). When V - vcrit = vref, the factor is one. This parameter provides a velocity scale for the factor.

vcrit

Critical velocity for the reaction rate (m/day | ft/day | cm/min). The factor is non-zero when the phase velocity exceeds vcrit. This parameter provides a cut-off velocity for the factor. vcrit is optional, and defaults to zero.

## **DEFAULTS:**

If \*PERMSCALE is absent, the reaction or mass transfer rate remains independent of effective permeability.

In the absence of \*MTVEL no phase velocity dependence is assumed. If \*MTVEL is present and vcrit is absent, then vcrit = 0 is assumed.

## **CONDITIONS:**

This option is effective only in conjunction with a current chemical reaction, since it applies to \*FREQFAC.

## **EXPLANATION:**

### **Nonequilibrium Mass Transfer**

The reaction model's heterogeneous mass transfer (source- sink) terms can be applied to the nonequilibrium capture and release of emulsion fines particles by the porous rock. This requires that the (reaction) rate constants depend upon permeability, to account for the changes in capture efficiency as the droplet size to pore throat size ratio changes. These effects are documented in:

- Radke, Soc. Pet. Eng. J., June 1984, p 351,
- Radke, J. Coll. Int. Sc., v. 102, 1984, p 462,
- Folger, Soc. Pet. Eng. J., Feb 1983, p 55, and
- Folger, J. Coll. Int. Sc., v 101, 1984, p 214.

Example: The simple capture of oil-in-water emulsion globules of molar concentration w(2) by the porous medium can be represented by the first order capture process

$$d(cc)/dt = ka * denw * w(2)$$

where cc is the moles of captured globules, t is time, denw is the water phase density. The rate constant ka is also known as the filter coefficient and, in general, is permeability and/or velocity dependent.

This process is modelled using a two-step procedure. First a reference rate constant ka is obtained for a given permeability and is entered via \*FREQFAC. Then rate constants are obtained for several other permeabilities and are entered as scaling factors relative to the reference ka. The permeability corresponding to the scaling factor FREQT=1 defines ka.

Sample data might appear as follows:

```
*FREQFAC 40  ** unit is 1/min, referenced at 0.21 Darcy
*PERMSCALE    ** Permeability   Scaling Factor   Rate
                ** EFFPT          FREQT        Constant
                ** (Darcy)           ka (1/min)
                  0.14            2.000      **     80
                  0.21            1.000      **     40 <--
                  0.32            0.675      **     27
                  0.58            0.200      **      8
                  1.28            0.100      **      4
                  5.12            0.075      **      3
```

Current research into the mechanisms of emulsion and/or foam creation and decay in-situ indicate possible dependence on phase velocity.

---

## Partial Equilibrium Reactions

\*RSEQFOR, \*RSEQBAK

### PURPOSE:

Chemical reactions which deviate from equilibrium.

### FORMAT:

```
*RSEQFOR comp_name rxk1 rxk2 rxk3 rxk4 rxk5
*RSEQFOR comp_name *KVTABLE ( *GL | *LL )
  K_value_table
*RSEQFOR comp_name *KVTABLE ( *GL | *LL )
  { *KEYCOMP
    K_value_table}
```

The syntax for \*RSEQBAK is obtained by substituting \*RSEQFOR for \*RSEQBAK in the above.

### DEFINITIONS:

#### \*RSEQFOR

Keyword indicating the entry of data for a forward reaction.

#### \*RSEQBAK

Keyword indicating the entry of data for a backward reaction.

#### comp\_name

Fluid component name in quotes assigned via \*COMPNAME. *comp\_name* may not be a solid component.

#### rxk1

First coefficient in the correlation for K value (kPa | psi).

#### rxk2

Second coefficient in the correlation for K value (1/kPa | 1/psi).

#### rxk3

Third coefficient in the correlation for K value

#### rxk4

Fourth coefficient in the correlation for K value (C | F). This coefficient has the unit of temperature difference. It has the same value for temperature scales C and K, and has the same value for temperature scales F and R.

#### rxk5

Fifth coefficient in the correlation for K value (C | F). This coefficient has the unit of temperature, and is different for each temperature scale. Often this coefficient is quoted in other sources in K or R, so it must be converted if C or F was specified for temperature in \*INUNIT. Convert it from absolute to C or F.

**\*GL, \*LL**

Limit values for the table's p, T and composition parameters will be those assigned via \*KVTABLIM and \*KVKEYCOMP for the gas-liquid (\*GL) liquid-liquid (\*LL) K value tables. See the definitions for keywords \*GASLIQKV and \*LIQLIQKV.

**K\_value\_table**

See the definition for keyword \*KVTABLE.

**\*KEYCOMP**

See the definition for keyword \*KVTABLE.

**DEFUALTS:**

If keywords \*RSEQFOR and \*RSEQBAK are absent, then the option is not used.

If \*KVTABLE appears with neither \*GL nor \*LL, then \*GL is assumed.

**CONDITIONS:**

If the table option \*KVTABLE is used, then \*KVTABLIM corresponding to \*GL or \*LL must have appeared previously. If \*KEYCOMP is used then \*KVKEYCOMP corresponding to \*GL or \*LL must have appeared previously.

Any table defined here must have the same number of columns and rows as any preceding K value table of the same type (\*GL or \*LL). Also, if this is the first of a type (\*GL or \*LL) of K value type, then any subsequent K value tables must have the same number of columns and rows.

\*RSEQFOR and \*RSEQBAK may not occur more than once in any one reaction definition. Also, a reaction may have only \*RSEQFOR or \*RSEQBAK or neither, but not both.

For each reaction you may choose one of the following five options:

*rxk1 rxk2 rxk3 rxk4 rxk5  
\*KVTABLE \*GL  
\*KVTABLE \*LL  
\*KVTABLE \*GL with \*KEYCOMP  
\*KVTABLE \*LL with \*KEYCOMP*

**EXPLANATION:**

See Appendix D.14 for further discussion.

This option modifies the reaction expression for the specified component in that the deviation from equilibrium mole fraction is employed. Thus, the concentration factor for the component (indicated here as subscript 'i') becomes

$$c(i) = \text{porf} * \text{den(iphas}(i)\text{)} * \text{sat(iphas}(i)\text{)} * \text{delta\_x(iphas}(i),i)$$

where all terms except delta\_x are defined above. For \*RSEQFOR

$$\text{delta\_x(iphas}(i),i) = \max ( 0, x(iphas}(i),i) - x_{\text{equil}} )$$

while for \*RXEQBAK,

$$\text{delta\_x(iphas(i),i)} = \max(0, \text{xequil} - \text{x(iphas(i),i)})$$

The equilibrium value "xequil" is the inverse of the K value from the correlation

$$K(p,T) = (rxk1/p + rxk2*p + rxk3) * \exp(rxk4 / (T-rxk5))$$

or from the table K(p,T,Xkey), where p is pressure, T is temperature and Xkey is the optional composition dependence.

See Table 2 for suggested correlation values, when the equilibrium is a gas-liquid type.

The physical meaning of the K values used in this option is specified by the reaction stoichiometry keywords. Possible processes include gas-to-liquid; liquid-to-liquid; or liquid-to-solid and vice-versa. However if the table option \*KVTABLE is used to specify the K values, then the table limit parameters from previously specified K value tables are employed. The keywords \*GL and LL indicate which equilibrium K value limit parameters are employed. Thus for example either \*GL or \*LL keywords can be employed for a solid-liquid partial equilibrium process, as specified by the appropriate reaction stoichiometry.

This option is useful in describing a rate-dependent approach to equilibrium, such as in foamy oil modelling.

### Unused Coefficients

According to the keyword syntax, there must be a value for each one of the data items after the keyword. If the correlation you wish to model is a subset of the one shown above for K(p,T), then enter zero for each "unused" coefficient rxk1 to rxk5. Whether or not rxk5 is "unused" depends on the temperature scale specified in \*INUNIT.

For example, assume that the input temperature scale specified via \*INUNIT is Celsius. We wish to assign a forward partial equilibrium K value for component 'Gas Bubl' based on the correlation

$$K1(p,T) = (A / p) * \exp(B / T)$$

where T is absolute degrees, as is common. First, we must rewrite the desired correlation in our input units

$$K1(p,T) = (A / p) * \exp(B / (TC+273))$$

where TC is in the input temperature scale Celsius. We see that rxk5 is not "unused" but has the value -273, since 0 K = -273 C. The keyword data in this case is

```
*RXEQFOR 'Gas Bubl' A 0 0 B -273
```

---

## **Ice Modelling (Optional)**

**\*ICE**

### **PURPOSE:**

Allow modelling of ice (solid water).

### **FORMAT:**

**\*ICE**

### **DEFINITIONS:**

**\*ICE**

Enable the modelling of ice formation from liquid water when temperature falls below 0° C (32° F). In addition, allow values of \*MINTEMP down to -100° C (-148° F).

### **DEFAULTS:**

If keyword \*ICE is absent then the formation of ice from water is not allowed and \*MINTEMP may be restricted to a value above 0° C (32° F).

### **CONDITIONS:**

You may not use the \*THERMAL option of \*SURFLASH with temperatures below 0° C.

### **EXPLANATION:**

#### **Minimum Temperature**

Two separate steps are required to enable the ice option: (1) add keyword \*ICE to the Component Properties data section, and (2) override the default of \*MINTEMP with a value below the lowest expected temperature. When keyword \*ICE is present, the minimum allowed value of \*MINTEMP decreases from just above 0° C (32° F) to -100° C (-148° F). To get ice you must explicitly specify a temperature below freezing, e.g., \*MINTEMP -20.

You can enter any temperature data (e.g., \*TEMP) down to the minimum \*MINTEMP. When initial temperature \*TEMP is below 0° C (32° F), initial liquid water is changed internally to ice resulting in an initial water saturation of zero. A value of injection temperature \*TINJW below 0° C is not allowed for water phase.

#### **Physical Process**

Consider a block containing liquid water at 10° C and hence no ice. As heat is withdrawn the temperature decreases. At the freezing point, continued heat withdrawal causes the liquid water to convert to solid ice at a constant temperature of 0° C. Once the conversion to ice is complete, the temperature will again decrease. Liquid water co-exists with ice only at 0° C (32° F). The same process occurs in reverse when heat is added to ice below 0° C.

#### **Output**

To see the ice concentration, use subkeyword \*ICECONC with \*OUTPRN \*GRID, \*OUTSRF \*GRID and \*OUTSRF \*SPECIAL \*BLOCKVAR, etc. This subkeyword uses the same unit indicator as \*SOLCONC and \*ADSORP: \*MOLE, \*MASS, \*VOL or \*NUM. For example, to see the full grid dump of fraction of void volume occupied by ice use \*OUTSRF \*GRID \*VOL \*ICECONC.

## **Effect on Porosity**

Ice is water component in the solid/immobile phase, just like a solid component or an adsorbed fluid component. Therefore, variations in ice concentration will affect the fluid porosity (see STARS User's Guide Appendix F.2). To see the effect of ice concentration on fluid porosity, use subkeyword \*FPOROS with \*OUTPRN \*GRID, \*OUTSRF \*GRID and \*OUTSRF \*SPECIAL \*BLOCKVAR, etc. Remember that water, oil and gas saturations are fractions of the fluid pore volume, so saturations will change when water freezes even though oil and gas volumes are unchanged.

## **Effect on Permeability**

Modelling of ice in a fluid-flow context often requires use of a variable permeability option (\*PERMCK, etc.) since permeability can vary with fluid porosity which itself varies with ice concentration. All variable permeability options work well for porosities significantly larger than zero.

If very small or zero fluid porosities are expected then it is best to use the \*PERMTAB option with a zero permeability ratio entry at a non-zero fluid porosity ratio. All the other variable permeability options achieve zero permeability only at zero fluid porosity. This often leads to fluid attempting to flow through a vanishingly small pore volume, a situation which is physically questionable and numerically very difficult.

## **Properties of Ice**

Density and enthalpy for ice are calculated from internal correlations which are valid from 0° C to -40° C (1998 ASHRAE Refrigeration Handbook (SI), page 8.2). Since STARS allows you to work with temperatures down to -100° C, you must ensure that the resulting property values below -40° C are appropriate. In the following,  $T_c$  is temperature in C.

Ice density (kg/m<sup>3</sup>):

$$916.89 - 0.13071 \cdot T_c$$

Ice specific heat (kJ/kg-K):

$$2.0623 + 6.0769 \cdot 10^{-3} \cdot T_c$$

Ice latent heat of fusion (kJ/kg):

333.6 at 0° C

Properties for ice correspond to the pure water component #1. At present there is no facility for the user to change the density and enthalpy correlations for ice. Water vapour cannot be specified initially or injected below 0° C. The ice formulation is strictly non-compositional and may not be appropriate for more complex freezing phenomenon like hydrates.

## **Other Sub-zero Properties**

Below 0° C the properties of other components in the oil and gas phases are obtained from their usual tables and correlations. For example, in the liquid viscosity table \*VISCTABLE you can enter temperatures down to the \*MINTEMP value. In general, temperatures below 0° C are disallowed only for a component that is using internal water properties.

## Related Keywords

Use keyword \*MINTEMP to decrease the minimum allowed temperature.

Use keywords \*HEATR, \*TMPSET, \*UHTR, etc., in recurrent data to add or remove heat similar to a heater. Keywords \*AUTOCOOLER and \*HEATSLAVE allow more advanced cooling control. In an actively cooled reservoir, heat transfer to and from adjacent formations can be modelled with keywords \*HLOSSPROP, etc., or \*TMPSET, \*UHTR, etc.

## Template Data Sets

The following template data sets illustrate the simulator's ice modelling capabilities.

STFLU026	Steam cycling in an initially frozen formation. Some initial time is spent heating the wellbore since initial injectivity is very small.
STFLU027	A steam flood encounters a cooling well. Special histories for key process parameters are dumped for block (9,1,1) which contains a cooling well. View all these special histories together on the same plot with RESULTS graph. The first 10 days show the freezing process in which the temperature falls until 32 F and then $S_w$ , fluid porosity and permeability fall while ice concentration rises. Before 2000 days the oil is being swept out, causing the fluid porosity and permeability to fall to zero and the ice concentration to rise to fill the void pore space. Note that the oil has been displaced by water that came from the steam flood but was condensed by the cooling well.
STFLU028	STFLU027 specifies variable permeability with *PERMTAB which gives a zero permeability at a non-zero fluid porosity, resulting in a non-zero final fluid porosity. On the other hand, STFLU028 uses *PERMCK which gives non-zero permeability for all non-zero fluid porosity, resulting in a final fluid porosity of zero but a more difficult numerical problem.
STFLU029	Models a cooling barrier around a single steam cycling well.
STFLU030	Models three steam cycles with cooling in an outer radial block to -20 deg F, with *AUTOCOOLER constraint on maximum cooling rate. Multi-block cooler is controlled from single block via *HEATSLAVE. See temperature and heater rate (CCHLOSS) histories for blocks (11,1,1:4) in Results Graph.
STFLU031	Models a cooling barrier in a fine Cartesian grid. Observe the spread of the frozen front with time in Results 3D.

# Rock-Fluid Data

## Summary of Rock-Fluid Data

Define relative permeabilities, capillary pressures, and component adsorption, diffusion and dispersion.

### List of Options

Relative permeability and capillary pressure has the following options:

- multiple rock types
- temperature dependence
- flexible endpoint over-riding by rock type and by block
- Stone's middle-phase models
- Linear interpolation middle-phase model
- interpolation between sets according to composition or cap number
- water-wet, oil-wet or one of three intermediate-wet options
- relative permeabilities in vertical direction different from horizontal
- capillary pressure can determine initial saturation distributions
- relative permeability hysteresis for wetting and non-wetting phase and capillary pressure hysteresis

Mechanical dispersivity has the following options:

- velocity dependence
- any phases
- different values for each grid block for each grid direction

Molecular diffusion has the following options:

- any component
- any phases
- different values for each grid block for each grid direction
- temperature and viscosity dependence

Total dispersion:

- combines mechanical dispersivity and molecular diffusion

Adsorption has the following options:

- any component in any phase
- composition dependence from Langmuir isotherm model or table
- temperature dependence
- multiple adsorbing rock types
- reversible and irreversible
- residual resistance factor

### Required Data

The minimum data required in this section is one set of relative permeability curves (\*SWT and \*SLT).

If multiple rock types are used, the minimum data is:

```
*SWT and *SLT ** Rock type #1 relative permeability curves  
*RPT 2 ** Enable rock type #2  
*SWT and *SLT ** Rock type #2 relative permeability curves  
*KRTYPE ** Assigns rock types to grid
```

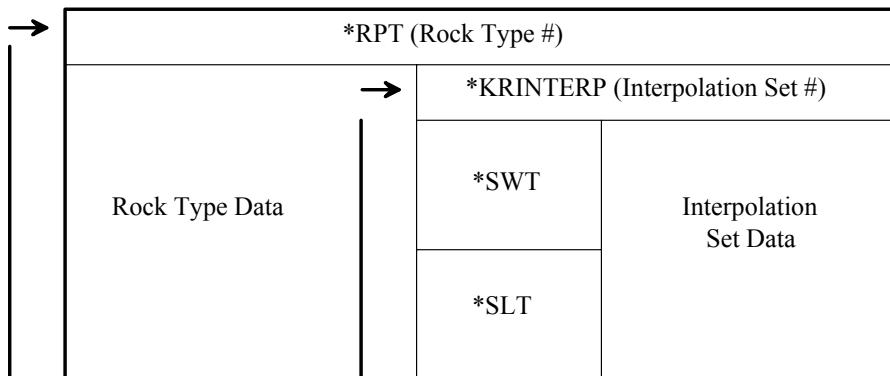
### Critical Keyword Ordering

1. Water-oil curve \*SWT must come before liquid-gas curve \*SLT, because \*SLT has an endpoint check which depends on an endpoint in \*SWT.
2. The rock-fluid data can be divided into three groups: data which varies only by rock types, data which varies by interpolation sets, and data which varies strictly by block. This is a summary:

Rock Type	Interpolation Set	Block
*RPT *IFTTABLE	*KRINTRP *SCRV *KRWIRO	*KRTYPE *RTYPE
*FMGCP *EPSURF	*KRTEMTAB *SWT *KRCOW	*KRTYPE_VERT
*FMOMF *EPCAP	*DTRAPW *SLT *KRGCW	*KRTYPE_CTRWAT
*INTCOMP *FMSURF	*DTRAPN *SWR *PCWEND	*KRTYPE_CTROIL
*INTLIN *FMCAP	*WCRV *SORW *PCGEND	*KRTYPE_CTRGAS
*EPGCP *INTLOG	*OCRV *GCRV *SGR *SORG	*BSWR *BSWCRT
*EPOMF *EPOIL	*HYS_KRO *HYS_KRW	*BSORW *BSOIRW
*FMOIL *FMMOB	*HYS_KRG *HYS_PCOW	*BSGR *BSCON
*FMSALT *EPSALT	*HYS_PCOG *PC_3RD_PHASE	*BSORG *BSOIRG *BSWRG
*FLOIL *FLSALT		*BSWIRG *BKRWIRO
		*BKRCOW *BKRGCW
		*BPCWMAX *BPCGMAX

The following diagram shows the scope and required ordering of each keyword. The vertical direction denotes critical ordering, whereas keywords linked in the horizontal direction can be intermixed. This diagram shows there is always a 'current' rock type number which is changed only by \*RPT. Reading starts with rock type #1 so the first \*RPT is not needed. When all the data for rock #1 is entered, \*RPT for the second rock type appears and the diagram is repeated as indicated by the outside return loop. Similar comments apply to the interpolation set number and the inner return loop.

Since the interpolation option operates within a rock type, the Interpolation Set loop falls inside the Rock Type outer loop. When the interpolation option is not used (keyword \*KRINTRP absent for a rock type), most keywords in the Interpolation Set column are valid for the rock type, that is, may appear once for each rock type.



See Appendix D.6 for further discussion of rock-fluid properties.

---

## Multiple Sets of Rock-Fluid Data

The ability to describe phenomena such as multiple rock types and interpolation of curves based on fluid compositions implies that multiple sets of rock-fluid data may be required. The following gives an overview of the manner in which this data is entered. Here, rock type #1 has interpolation between three sets of data which correspond to three surfactant concentrations; rock type #2 has no interpolation.

### ROCK TYPE 1: Rock type designator (\*RPT)

Interpolation Definition (\*INTCOMP, etc.)

#### Interpolation Set #1

Interpolation Set designator (\*KRINTRP)

Interpolation parameters (\*DTRAPW, etc.)

Relative Permeability/Cap Pressure data (\*SWT and \*SLT)

Optional data (\*KRTEMTAB, etc.)

#### Interpolation Set #2

Interpolation Set designator (\*KRINTRP)

Interpolation parameters (\*DTRAPW, etc.)

Relative Permeability/Cap Pressure data (\*SWT and \*SLT)

Optional data (\*KRTEMTAB, etc.)

#### Interpolation Set #3

Interpolation Set designator (\*KRINTRP)

Interpolation parameters (\*DTRAPW, etc.)

Relative Permeability/Cap Pressure data (\*SWT and \*SLT)

Optional data (\*KRTEMTAB, etc.)

### ROCK TYPE 2: Rock type designator (\*RPT)

Relative Permeability/Cap Pressure data (\*SWT and \*SLT)

Optional data (\*KRTEMTAB, etc.)

### ASSIGN ROCK TYPES (\*KRTYPE)

#### Rock Type Keywords (No Interpolation Sets)

When a rock type has no Interpolation Sets (keyword \*KRINTRP absent) then the following keywords may appear at most once for that rock type.

Rock Type Number	*RPT
Tables	*SWT *SLT
Hysteresis Parameters	*HYS_KRO, etc.
Endpoints	*SWR, etc.
Temperature Dependence	*KRTEMTAB

## **Rock Type Keywords (With Interpolation Sets)**

When a rock type has Interpolation Sets (keyword \*KRINTRP present) then the following keywords may appear at most once for that rock type

Rock Type Number	*RPT
Interpolation Component	*INTCOMP
Interfacial Tension	*IFTTABLE, etc.
Foam Interpolation	*FMSURF, etc.

and the following keywords may appear at most once per interpolation set for that rock type.

Interpolation Set Number	*KRINTRP
Interpolation Parameters	*DTRAPW, etc.
Tables	*SWT *SLT
Hysteresis Parameters	*HYS_KRO, etc.
Endpoints	*SWR, etc.
Temperature Dependence	*KRTEMTAB

## Interpolation of Relative Permeability and Capillary Pressure

### Options 1 and 2: Compositional Effects on Relative Permeability

Under special circumstances (nearly miscible fluids, pH changes, surfactant concentration changes, large increases in applied flow velocities), the assumption that rock-fluid properties are functions only of fluid saturations and saturation histories is not sufficient to accurately describe observed flow behavior.

In these cases, the ability to interpolate basic relative permeability and capillary pressure data as functions of concentration or capillary number can prove very useful. Because of the flexibility in the choice of interpolation parameter and the fact that arbitrary tabular data relative permeability and capillary pressure can be employed, a wide variety of phenomena can be handled. Currently, two interpolation options are available.

A capillary number is a dimensionless velocity representing the ratio of viscous to interfacial forces. For typical values of velocity  $v = 1 \text{ ft/day}$ , viscosity  $\mu = 1 \text{ cp}$ , and interfacial tension  $\sigma = 30 \text{ dynes/cm}$ , the capillary number  $N_c = \mu * v / \sigma = 1.0e-7$ . Note that the ratio  $\sigma/\mu$  is equivalent to a reference velocity  $v_r = 1.0e+7 \text{ ft/day}$ .

Example:

The relative permeability interpolation scheme provides the user with a flexible tool for representing surfactant effects on relative permeability. Consider a single rock type with conventional water/oil relative permeability curves corresponding to high interfacial tension. As surfactant is added to the system, residual saturations can decrease and the relative wettability of the phases can change. Ultimately, high surfactant concentrations and the resulting ultra-low interfacial tension values lower the residual saturations and straighten the relative permeability curves. This behavior was demonstrated experimentally by Van Quy and Labrid (Soc. Pet. Eng. J., June 1983, p. 461) and by Amaefule and Handy (Soc. Pet. Eng. J., June 1982, p. 371).

The four curves of van Quy and Labrid can be represented in two ways. Using one interpolation parameter DTRAPW, the four sets of relative permeability curves can be entered corresponding to the critical capillary numbers

$$\begin{aligned} N_c &= 6.0E-8 \text{ (both water and oil residual saturations start to decrease),} \\ N_c &= 2.6E-4 \text{ (intermediate curves reported by Van Quy and Labrid),} \\ N_c &= 1.2E-3 \text{ (residual oil saturation reaches zero),} \\ N_c &= 2.3E-1 \text{ (residual water saturation reaches zero, and relative permeabilities are straight lines).} \end{aligned}$$

Alternatively, the same information can be entered with only two sets of relative permeability curves by using DTRAPW and DTRAPN as follows:

1. High interfacial tension curves (no surfactant)

$$\text{DTRAPW} = \text{DTRAPN} = \log_{10}(6.0E-8)$$

2. Ultra low interfacial tension curves (straight lines)

$$\text{DTRAPW} = \log_{10}(2.3E-1) \text{ and } \text{DTRAPN} = \log_{10}(1.2E-3)$$

In either case, the same high-tension liquid/gas relative permeability curves can be entered for each interpolation set, if we assumed they are unaffected. On the other hand, it is the liquid/gas relative permeability curves which may change with solvent concentration when solvent injection causes oil and gas phases to mix.

These interpolation schemes can be extended to representing the behavior of different rock types. For example, surfactant injection into a parallel core system of differing wettability will require that high interfacial tension curves for the two rock types be distinguished.

The full three-phase relative permeability and capillary pressure results ( $k_{rw}$ ,  $k_{ro}$ ,  $k_{rg}$ ,  $P_{cow}$ ,  $P_{cog}$ ) are calculated for rock-fluid sets A and B, after which the two sets of values are interpolated as:

$$\begin{aligned} k_{rw} &= k_{rwA} \cdot (1-wtr) + k_{rwB} \cdot wtr \\ k_{ro} &= k_{roA} \cdot (1-oil) + k_{roB} \cdot oil \\ k_{rg} &= k_{rgA} \cdot (1-gas) + k_{rgB} \cdot gas \\ P_{cow} &= P_{cowA} \cdot (1-pcw) + P_{cowB} \cdot pcw \\ P_{cog} &= P_{cogA} \cdot (1-pcg) + P_{cogB} \cdot pcg \end{aligned}$$

where

$$\begin{aligned} wtr &= ratw^{WCRV} & gos = ratw^{SCRV} \\ oil &= ratn^{OCRV} & gas = ratn^{GRCV} \\ pcw &= (wtr+oil)/2 & pcg = gas \end{aligned}$$

and where  $ratw$ ,  $ratn$  are the current values of the dimensionless interpolation parameters (varying between 0 and 1)

$$ratw = \frac{\log_{10}(N_c) - DTRAPWA}{DTRAPWB - DTRAPWA} \quad ratn = \frac{\log_{10}(N_c) - DTRAPNA}{DTRAPNB - DTRAPNA}$$

The curvature interpolation parameters WCRV, OCRV, GCRV and SCRV allow additional flexibility in interpolating between sets of curves if experimental evidence requires it. Normally default values of 1 are recommended unless a poor history match is observed. Non-default values imply that the slopes of the interpolated relative permeability curves do not change at the same rate as the endpoint values. Thus  $WCRV = 2$  implies that the interpolated  $krw$  retains its  $krwA$  character more closely over range of interpolation, while  $WCRV = 0.5$  implies that  $krwB$  has the dominant influence. This can be verified by a close inspection of the interpolation formula shown for  $krw$ . Similar comments apply to the role of OCRV, GCRV and SRCV to  $krow$ ,  $krg$  and  $krog$ .

### Option 3: Empirical Foam Interpolation Scheme

Foam treatments are employed to reduce gas phase mobility and improve displacement efficiency as well as areal and vertical sweep. An empirical method of simulating much of the effects of foam can be accomplished by assuming that the mobility reduction corresponds to a decreased gas phase relative permeability (as a function of a product of experimentally observed factors, including surfactant concentration). Such method is useful for preliminary scoping of laboratory experiments as well as history matching and predicting field scale foam treatments. Most often use of this option also implies surfactant flow modelling, requiring additional surfactant property data such as adsorption, partition coefficients in oil, and surfactant decomposition kinetics.

A less empirical approach can also be accomplished with the concept of a lamella as a dispersed component. This requires utilizing appropriate viscosity, adsorption, and resistance factor data. This latter approach can also be viewed as partially validating the empirical foam option. Further details can be found in the STARS Technical manual and a CMG report 90.08.T.

---

## Critical and Connate Saturations, Scale-Up Factors, and Normalization

The critical and connate fluid saturations of the relative permeability and capillary pressure data tables are obtained from the tables. These saturations and endpoints are saved. Then the tables are normalized. Optional endpoint keywords \*SWR, etc., will over-write the saved endpoint data. Keyword \*KRTEMTAB will over-write the saved endpoint data with its lowest-temperature values, effectively making the lowest \*KRTEMTAB temperature the reference for the \*SWT and \*SLT input table values as well as the over-ridden table endpoints specified via keywords \*SWR, etc. In addition, \*KRTEMTAB flags interpolation of the specified endpoints as a function of temperature.

Interpolation between temperature entries of the \*KRTEMTAB tables is linear. For temperatures outside the range specified in the tables, endpoint values corresponding to the nearest table temperature entry are used, that is, values are not extrapolated.

There are two options for handling the two ways to measure liquid-gas systems. By default (\*SLT without \*NOSWC), it is assumed that the liquid in the liquid-gas system contains connate water, so  $S_l = S_o + S_{wc}$ . When the liquid does not contain water but is pure oil (\*SLT \*NOSWC),  $S_l = S_o$ . This distinction has an effect on the exact formulas used. However, the definition of  $S_{org}$  is the same in both cases, that is, the critical oil in the gas/liquid(oil) system. Once the critical and connate saturations are known, the normalization of water for the oil-water system, and liquid for the liquid-gas system, are

For the oil-water table:

$$\begin{aligned} k_{rw}, \quad S'_w &= S'_{wcrit} + (S_w - S_{wcrit}) * \left( \frac{1 - S'_{oirw} - S'_{wcrit}}{1 - S_{oirw} - S_{wcrit}} \right) \\ k_{row}, \quad S''_w &= S''_{wcon} + (S_w - S_{wcon}) * \left( \frac{1 - S'_{orw} - S'_{wcon}}{1 - S_{orw} - S_{wcon}} \right) \\ P_{cow}, \quad S'''_w &= S'''_{wcon} + (S_w - S_{wcon}) * \left( \frac{1 - S'_{oirw} - S'_{wcon}}{1 - S_{oirw} - S_{wcon}} \right) \end{aligned}$$

$S_w$

$S'_w, S''_w, S'''_w$

$S_{wcrit}, S_{wcon}, S_{oirw}$  and  $S_{orw}$

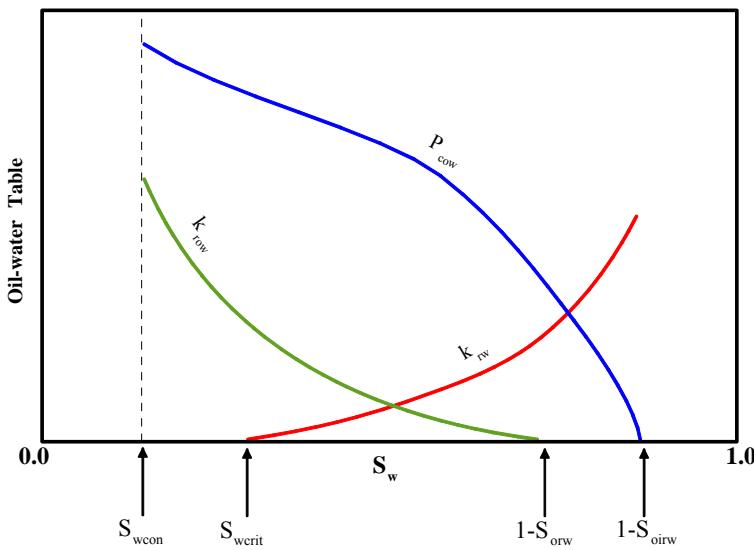
$S'_{wcrit}, S''_{wcon}, S'_{oirw}$  and  $S'_{orw}$

is the block's water saturation

are the saturations used to interpolate the original user entered tables with respect to  $k_{rw}$ ,  $k_{row}$  and  $P_{cow}$  respectively

are the user entered end point values (from the keywords for end point change or temperature dependence)

are the original table end point values (calculated directly from the relative permeability tables)

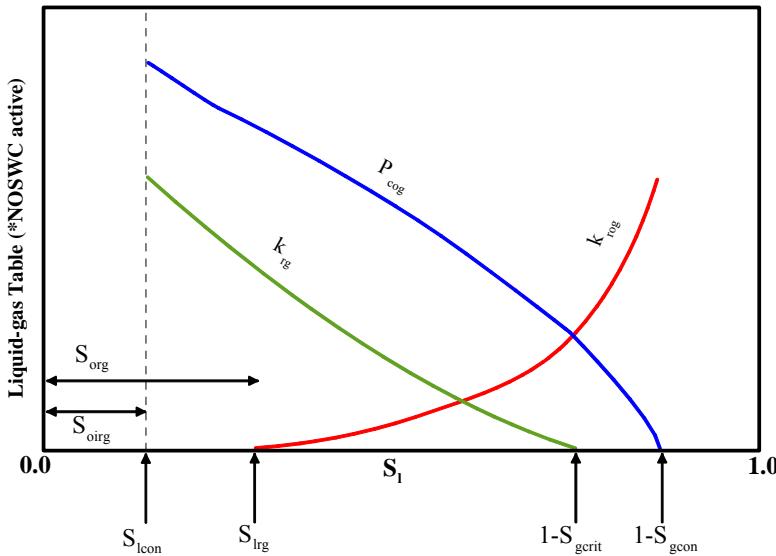


For the gas-liquid table:

$$k_{\text{rog}} \quad S'_1 = S'_{\text{lrg}} + (S_1 - S_{\text{lrg}}) * \left( \frac{1 - S'_{\text{lrg}} - S'_{\text{gcon}}}{1 - S_{\text{lrg}} - S_{\text{gcon}}} \right)$$

$$k_{\text{rg}} \quad S''_1 = S'_{\text{lcon}} + (S_1 - S_{\text{lcon}}) * \left( \frac{1 - S'_{\text{lcon}} - S'_{\text{gcrit}}}{1 - S_{\text{lcon}} - S_{\text{gcrit}}} \right)$$

$$P_{\text{cog}}, \quad S'''_1 = S'_{\text{lcon}} + (S_1 - S_{\text{lcon}}) \left( \frac{1 - S'_{\text{lcon}} - S'_{\text{gcon}}}{1 - S_{\text{lcon}} - S_{\text{gcon}}} \right)$$



$$\begin{aligned}
 S_l &= S_o + S_w \\
 S_{lrg} &= S_{wcon} + S_{org}, \text{ if the *NOSWC option is not used } S_{wc} \\
 S_{lrg} &= S_{org}, \text{ if the *NOSWC option is used}
 \end{aligned}$$

In the latter case  $S_{org}$  includes all residual liquid in the liquid-gas system.

$$S_l \\ S'_l, S''_l, S'''_l$$

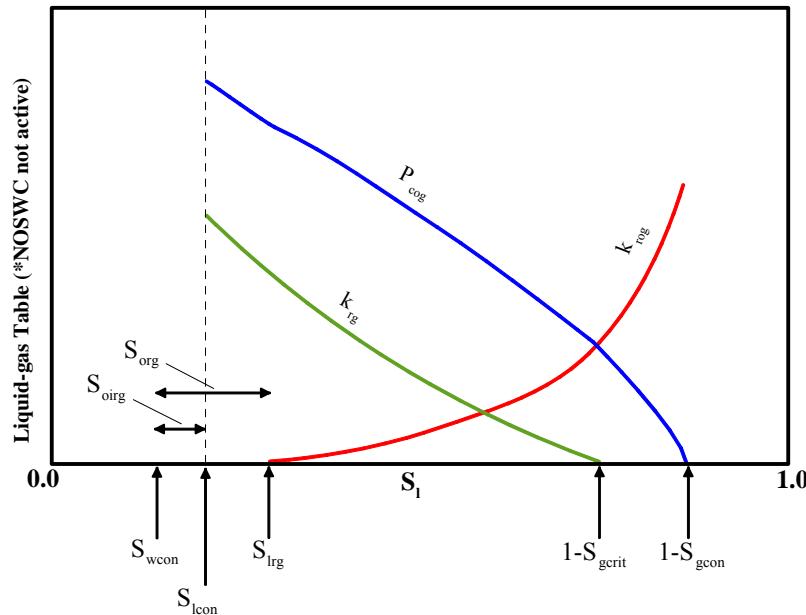
$S_{org}, S_{lcon}, S_{gcon}$  and  $S_{gerit}$

$S'_{org}, S'_{lcon}, S'_{gcon}$  and  $S'_{gerit}$

is the blocks liquid saturation  
are the saturations used to interpolate the original user entered tables with respect to  $k_{rog}$ ,  $k_{rg}$  and  $P_{cog}$  respectively  
are the user-entered grid block end point values (from the keywords for end point change or temperature dependence)  
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 that define the range of each curve.

$k_{rw}$	$S_{wcrit} \leq S_w \leq 1 - S_{oirw}$ ,	$k_{rw}$ is scaled by $S_{wcrit}$ and $S_{oirw}$
$k_{row}$	$S_{wcon} \leq S_w \leq 1 - S_{orw}$ ,	$k_{row}$ is scaled by $S_{wcon}$ and $S_{orw}$
$P_{cow}$	$S_{wcon} \leq S_w \leq 1 - S_{oirw}$ ,	$P_{cow}$ is scaled by $S_{wcon}$ and $S_{oirw}$
$k_{rog}$	$S_{lrg} \leq S_l \leq 1 - S_{gcon}$ ,	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 $S_{gcon}$ .
$k_{rg}$	$S_{lcon} \leq S_l \leq 1 - S_{gerit}$ ,	$k_{rg}$ is scaled by $S_{lcon}$ and $S_{gerit}$
$P_{cog}$	$S_{lcon} \leq S_l \leq 1 - S_{gcon}$ ,	$P_{cog}$ is scaled by $S_{lcon}$ and $S_{gcon}$



Then, normalized relative permeabilities and capillary pressures are obtained as functions of  $S'_w, S''_w, S'''_w$  and  $S'_l, S''_l, S'''_l$  via table look-ups. Finally, the relative permeabilities are multiplied by the calculated connate values to get the unnormalized result.

### End-point Saturation Modification Keywords

The keywords for modifying relative permeability end-points are shown below. A *Single Keyword* may be used by itself or after \*KRTEMTAB. A *Per-Block Keyword* is a grid property array.

<b>Symbol</b>	<b>Single Keyword</b>	<b>Per-Block Keyword</b>	<b>Description</b>
$S_{wcon}$	*SWCON,	*BSWCON,	Connate water saturation
$S_{wr}$	*SWR	*BSWR	
$S_{wcrit}$	*SWCRIT	*BSWCRIT	Critical water saturation
$S_{oirw}$	*SOIRW	*BSOIRW	Irreducible oil saturation to water
$S_{orw}$	*SORW	*BSORW	Residual oil saturation to water
$S_{gcon}$	*SGCON	*BSGCON	Connate gas saturation
$S_{gcrit}$	*SGR	*BSGR	Critical gas saturation
$S_{oирg}$	*SOIRG	*BSOIRG	Irreducible oil saturation to gas
$S_{org}$	*SORG	*BSORG	Residual oil saturation to gas
$S_{wирg}$	*SWIRG	*BSWIRG	Irreducible oil saturation to water
$S_{wrg}$	*SWRG	*BSWRG	Residual oil saturation to water

### Equality of Critical and Connate Saturations

End-point saturations can be divided into connate/critical pairs ( $S_{wcon}, S_{wcrit}$ ), ( $S_{oirw}, S_{orw}$ ), ( $S_{gcon}, S_{gcrit}$ ), ( $S_{oирg}, S_{org}$ ) and ( $S_{wирg}, S_{wrg}$ ). Except for the constraint connate  $\leq$  critical, in the most general case the two members of each pair are independent of each other. Often, however, connate = critical and it is the user's intent that this equality be preserved even after end-point modification options are applied.

To facilitate this, STARS keeps track of which end-point pairs are equal in the \*SWT and \*SLT tables, and then enforces equality during modification of one end-point by internally applying the same modification to the other. This equality enforcement is done for all the end-point modification options: single end-points, temperature dependence and per-block end-points.

For example, assume that  $S_{oirw} = S_{orw}$  is detected in the \*SWT table. If single end-point keyword \*SOIRW appears but keyword \*SORW does not, or if \*SORW appears but \*SOIRW does not, then both  $S_{oirw}$  and  $S_{orw}$  are assigned the new value. However, if both keywords appear then  $S_{oirw}$  and  $S_{orw}$  are assigned their separate new values, thus breaking their equality. The same process occurs for keywords \*SOIRW and \*SORW appearing in the list after \*KRTEMTAB, as well as for per-block keywords \*BSOIRW and \*BSORG.

Per-block end-point modification has an additional complication. When a per-block end-point keyword is entered, then all blocks use the per-block option for that end-point. This is facilitated by the fact that the end-point value of each block is initialized with the value from that block's rock type, which is then over-written with any input per-block values. If only one of a pair of per-block end-point keywords appears, the same data may be applied to the other end-point. However, this duplication is done only if there is at least one block for which (1) the end-points are equal for that block's rock type, and (2) the per-block value entered is different from that rock type's table value.

### Combination with Other Options

There are four options for overriding rock-fluid endpoints that have been specified in tables \*SWT and \*SLT:

1. Override with keywords \*SWR, etc.
2. Override with temperature dependence via \*KRTEMTAB. This will override values from keywords \*SWR, etc.
3. Interpolate between sets via \*INTCOMP and \*KRINTRP.
4. Override on a per block basis via \*BSWR, etc.

All these options work together. For example, you may specify that Swr has temperature dependence (\*KRTEMTAB), different values depending on the amount of surfactant in place (\*INTCOMP and \*KRINTRP) as well as a complex distribution in space (\*BSWR).

A per-block array like \*BSWR refers to the first \*KRTEMTAB temperature (if there is temperature dependence) and the first interpolation set in \*KRINTRP (if there is interpolation). It is the relative variation of the individual block's value from the "reference" value (first T and first interpolation set for the associated rock type) that is applied to the value calculated from the temperature and surfactant of interest.

The per-block option may be used also with oil-wet or intermediate wettability rock types. The interpretation of the relative permeability columns is applicable also to the keywords \*BKRWRO, \*BKROCW and \*BKRCGW.

Example:

Assume the following for quantity Swr:

```
*KRINTRP 1 *DTRAPW 0      ** No surfactant
    Value from *SWT is 0.2
    Value from *KRTEMTAB is 0.18 at 100 F and 0.28 at 300 F.
*KRINTRP 2 *DTRAPW 1e-3   ** Some surfactant
    Value from *SWT is 0.08
    Value from *KRTEMTAB is 0.06 at 100 F and 0.09 at 300 F.
    Value from *BSWR for block 1 is 0.216
```

We have interpolation with surfactant amount, temperature dependence and per-block overrides. At the time the first \*SWT is read, the table's  $S_{wr}$  and  $S_{orw}$  are saved and then the table's  $S_w$  entries are normalized. When the first \*KRTEMTAB is read,  $S_{wr} = 0.2$  is replaced with  $S_{wr} = 0.18$  since the \*SWT table is referenced to the lowest T. Then the same process is applied to the second interpolation set, with  $S_{wr} = 0.06$  as a result. When \*BSWR is read, the value 0.216, which is referenced to 100 F and no surfactant, is saved in separate storage  $S_{wr}$  for block #1.

The following is done to get a value of  $S_{wr}$  for  $T = 235$  F and some surfactant amount:

- a) Get  $S_{wr}$  at no surfactant by interpolating between 100 and 300 F

$$R = (235 - 100) / (300 - 100) = 0.675$$

$$S_{wr1} = (1-R)*0.18 + R*0.28 = 0.2475$$

Adjust the individual block's value with this relative variation from the  $S_{wr}$  reference value (at no surfactant and 100 F)

$$S_{wrA} = S_{wr1} * (0.216 / 0.18) = 0.297$$

Use this to get normalized block  $S_w$  from which the table lookup is done to get "no surfactant" value of normalized  $k_{rw}$  and  $k_{row}$ . Use other endpoint scaling factors to unnormalize  $k_{rw}$ , etc.

- b) Do the same for the 1e-3 surfactant amount

$$R = (235 - 100) / (300 - 100) = 0.675$$

$$S_{wr2} = (1-R)*0.06 + R*0.09 = 0.08025$$

$$S_{wrB} = S_{wr2} * (0.216 / 0.18) = 0.0963$$

- c) The values of  $k_{rw}$  and  $k_{row}$  for the two surfactant amounts are mixed.

This example has both T and component dependence. For T dependence alone, the algorithm is just part (a) above. For component dependence alone, the algorithm is equivalent to factor  $R = 0$ . Note that no scaling is done if the reference end point (first T and interpolation set) is zero. Also, it is the user's responsibility to ensure that end-point values after scaling are within physical ranges.

---

## Three-Phase Models

### Stone's Model I

Stone's Model I may be used to combine oil-water and liquid-gas two-phase relative permeabilities. This model assumes that (1)  $S_{gc} = 0$ , and (2)  $S_{om}$  varies between  $S_{orw}$  and  $S_{org}$ . This option is available only when  $S_{wc}$  is included in the liquid saturation (\*NOSWC absent).

The calculation of three-phase oil relative permeability is

$$k_{ro} = k_{row} * k_{rog} * S_{eo} / (k_{rocw} * S_{el} * (1 - S_{ew}))$$

$$S_{eo} = (S_o - S_{om}) / (1 - S_{wc} - S_{om})$$

$$S_{ew} = (S_w - S_{wc}) / (1 - S_{wc} - S_{om})$$

$$S_{el} = 1 - S_g / (1 - S_{wc} - S_{om})$$

where  $k_{rocw} = k_{row}(S_w = S_{wc}) = k_{rog}(S_g = 0)$  to ensure that  $k_{ro} = k_{row}$  when  $S_g = 0$  and that  $k_{ro} = k_{rog}$  when  $S_w = S_{wc}$ .

For the "minimal" value  $S_{om}$  of the oil saturation, STARS uses the linear function of  $S_g$  proposed by Fayers and Matthews (SPEJ April 1984, pp. 224-232):

$$S_{om}(S_g) = (1 - a(S_g)) * S_{orw} + a(S_g) * S_{org},$$

where

$$a(S_g) = S_g / (1 - S_{wcrit} - S_{org}).$$

### Stone's Model II (modified)

The relative permeability of water in the three-phase system is equal to the water relative permeability in the two-phase water-oil system, and is a function only of  $S_w$ . 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 only of  $S_g$ . The three-phase oil relative permeability is calculated using the modification of Settari and Aziz. (Aziz, K., and Settari, A., "Petroleum Reservoir Simulation," Applied Science Publishers Ltd., London, 1979).

Liquid Contains  $S_{wc}$  (default):

$$k_{ro} = k_{rocw} * ((k_{row}/k_{rocw} + k_{rw}) * (k_{rog}/k_{rocw} + k_{rg}) - k_{rw} - k_{rg})$$

where

$$k_{rocw} = k_{row}(S_w = S_{wc}) = k_{rog}(S_g = 0) \text{ to ensure that } k_{ro} = k_{row} \text{ when } S_g = 0 \text{ and that } k_{ro} = k_{rog} \text{ when } S_w = S_{wc}.$$

Liquid Does Not Contain  $S_{wc}$  (\*NOSWC):

$$k_{ro} = k_{romax} * ((k_{row}/k_{romax} + k_{rw}) * (k_{rog}/k_{romax} + k_{rg}) - k_{rw} - k_{rg})$$

where

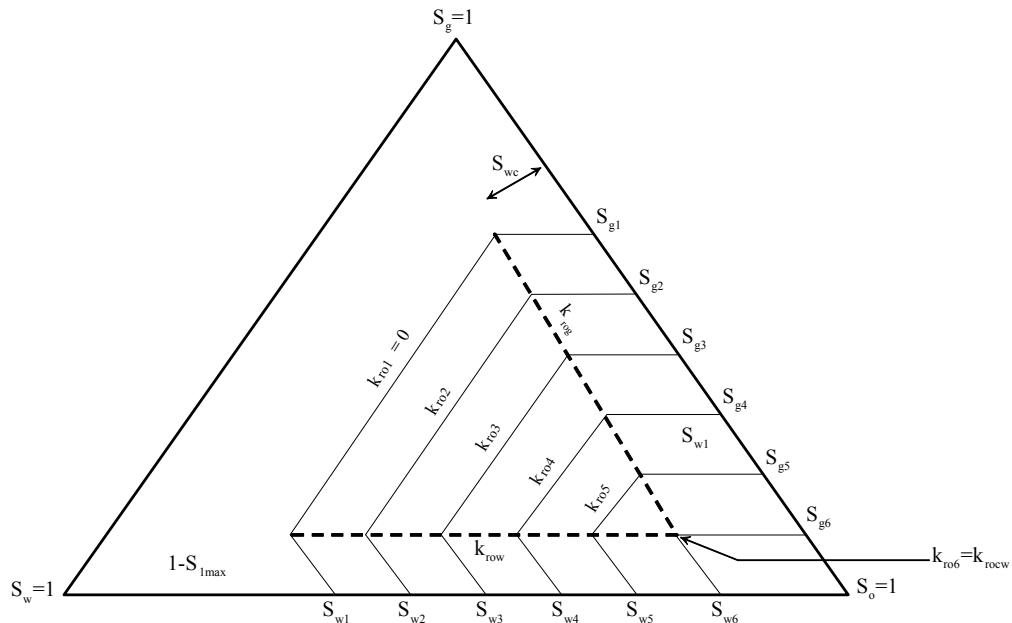
$$k_{romax} = k_{row}(S_w = 0) = k_{rog}(S_g = 0) \text{ to ensure that } k_{ro} = k_{row} \text{ when } S_g = 0 \text{ and that } k_{ro} = k_{rog} \text{ when } S_w = 0.$$

When plotted on a saturation ternary diagram like Figure 12, constant  $k_{ro}$  contours between the two-phase water-oil and gas-liquid data tend to be curved toward the  $S_o=0$  boundary. For some sets of two-phase data, the  $k_{ro}=0$  curve may cross the  $S_o=0$  boundary, resulting in a range of  $S_w$  values where  $k_{ro}>0$  when  $S_o = 0$ . Such a condition is not only unphysical, but it can reduce numerical stability. This condition is detected when the detailed rock-fluid data echo is written to the output file, but it is flagged only as a warning since the affected saturation range may not be encountered during timestepping. However, this condition is detected for individual blocks at the end of each timestep, and is flagged as a fatal error after a number of warnings. The equivalent condition is detected when  $k_{rw}$  is the middle phase.

### Baker's Linear Interpolation Model

The middle phase relative permeability is calculated from a linear interpolation scheme described by L. E. Baker in "Three-Phase Relative Permeability Correlations", SPE/DOE paper 17369.

Figure 12 shows a schematic of the geometrical construction for the method. The functions  $k_{row}(S_w)$  and  $k_{rog}(S_g)$  are placed on the ternary diagram of saturations, and points of equal oil relative permeability, that is,  $k_{row}(S_w) = k_{rog}(S_g)$ , are linked with a straight line to form a series of  $k_{ro}$  contours. This method works because  $k_{row}$  and  $k_{rog}$  have the same range (0 to  $k_{rocw}$ ).



**Figure 12: Geometric Construction of Linear Interpolation Method**

Note that at any  $(S_w, S_g)$  in the "interpolation" region  $k_{ro}$  depends only on  $k_{row}$  and  $k_{rog}$ . This is in contrast to Stone's II model where  $k_{ro}$  depends also on  $k_{rw}$  and  $k_{rg}$ .

---

## Wettability Options

There are five wettability options: water-wet, oil-wet and three models of intermediate wettability. The data requirements and formulas used are summarized here. Water-wet rock is the default. Only the water-wet option is available with set interpolation (\*INTCOMP).

These wettability options are static since they correspond to the original in-situ fluid composition and rock type. They should not be confused with the dynamic rock-fluid interpolation technique of modelling wettability as a function of composition or capillary number.

### Water Wet (\*WATWET)

This is the usual wettability choice, and is the default. This option assumes that the water phase is next to the rock, and oil is the middle phase. The three-phase relative permeability calculation is:

- a) Obtain  $k_{rw}$  and  $k_{row}$  from \*SWT, as a function of  $S_w$
- b)  $k_{rocw} = k_{row}(S_w=S_{wc})$
- c) Obtain  $k_{rg}$  and  $k_{rog}$  from \*SLT, as a function of  $S_g$
- d)  $k_{ro} = k_{rocw} * ((k_{row}/k_{rocw} + k_{rw}) * (k_{rog}/k_{rocw} + k_{rg}) - k_{rw} - k_{rg})$
- e)  $k_{rw}$  and  $k_{rg}$  are the same as the two-phase values

### Oil Wet (\*OILWET)

This option assumes that the oil phase is next to the rock, and water is the middle phase. The three-phase relative permeability calculation is:

- a) Obtain  $k_{rwo}$  and  $k_{ro}$  from \*SWT, as a function of  $S_o$
- b)  $k_{rwco} = k_{rwo}(S_o=S_{oc})$
- c) Obtain  $k_{rg}$  and  $k_{rwg}$  from \*SLT, as a function of  $S_g$
- d)  $k_{rw} = k_{rwco} * ((k_{rwo}/k_{rwco} + k_{ro}) * (k_{rwg}/k_{rwco} + k_{rg}) - k_{ro} - k_{rg})$
- e)  $k_{ro}$  and  $k_{rg}$  are the same as the two-phase values

### Intermediate Wet Model #1 (\*INTMED1)

This option assumes that one half of the pores are water-wet and the other half is oil-wet. The three-phase relative permeability calculation is:

- a) Obtain  $k_{rw(w)}$  and  $k_{row}$  from \*SWT, as a function of  $S_w$
- b)  $k_{rocw} = k_{row}(S_w=S_{wc})$
- c) Obtain  $k_{rg}$  and  $k_{rog}$  from \*SLT, as a function of  $S_g$
- d)  $k_{ro(w)} = k_{rocw} * ((k_{row}/k_{rocw} + k_{rw(w)}) * (k_{rog}/k_{rocw} + k_{rg}) - k_{rw(w)} - k_{rg})$
- e) Obtain  $k_{rwo}$  and  $k_{ro(o)}$  from \*SWT, as a function of  $S_o$
- f)  $k_{rwco} = k_{rwo}(S_o=S_{oc})$
- g) Obtain  $k_{rwg}$  from \*SLT, as a function of  $S_g$
- h)  $k_{rw(o)} = k_{rwco} * ((k_{rwo}/k_{rwco} + k_{ro(o)}) * (k_{rwg}/k_{rwco} + k_{rg}) - k_{ro(o)} - k_{rg})$

- i)  $k_{rw} = (k_{rw(w)} + k_{rw(o)}) / 2$
- j)  $k_{ro} = (k_{ro(w)} + k_{ro(o)}) / 2$
- k)  $k_{rg}$  is the same as the two-phase value

### Intermediate Wet Model #2 (\*INTMED2)

This option obtains  $k_{rw}$  assuming oil-wet, and obtains  $k_{ro}$  assuming water-wet. The three-phase relative permeability calculation is:

- a) to h) same as for Intermediate Wet Model #1
- i)  $k_{rw} = k_{rw(o)}$
- j)  $k_{ro} = k_{ro(w)}$
- k)  $k_{rg}$  is the same as the two-phase value

### Intermediate Wet Model #3 (\*INTMED3)

This option calculates both  $k_{rw}$  and  $k_{ro}$  from an application of Stone's model to two-phase  $k_{rwo}$  and  $k_{row}$  obtained by averaging water-wet and oil-wet values. The three-phase relative permeability calculation is:

- a) Obtain  $k_{rw(w)}$  and  $k_{row(w)}$  from \*SWT, as a function of  $S_w$
- b) Obtain  $k_{rwo(o)}$  and  $k_{ro(o)}$  from \*SWT, as a function of  $S_o$
- c)  $k_{rwo} = (k_{rw(w)} + k_{rwo(o)}) / 2$
- d)  $k_{row} = (k_{row(w)} + k_{ro(o)}) / 2$
- e)  $k_{rocw} = k_{row(w)}(S_w = S_{wc})$
- f) Obtain  $k_{rg}$  and  $k_{rog}$  from \*SLT, as a function of  $S_g$
- g)  $k_{ro} = k_{rocw} * ((k_{row} / k_{rocw} + k_{rwo}) * (k_{rog} / k_{rocw} + k_{rg}) - k_{rwo} - k_{rg})$
- h)  $k_{rwco} = k_{rwo}(S_o = S_{oc})$
- i) Obtain  $k_{rwg}$  from \*SLT, as a function of  $S_g$
- j)  $k_{rw} = k_{rwco} * ((k_{rwo} / k_{rwco} + k_{row}) * (k_{rwg} / k_{rwco} + k_{rg}) - k_{row} - k_{rg})$
- k)  $k_{rg}$  is the same as the two-phase value

### Summary of Meaning of \*SWT and \*SLT Tables

Option	$k_{rw}$ Table	$k_{row}$ Table	$k_{rog}$ Table
*WATWET	$k_{rw}$	$k_{row}$	$k_{rog}$
*OILWET	$k_{rwo}$	$k_{ro}$	$k_{rwg}$
*INTMED1	$k_{rw(w)}$ & $k_{rwo}$	$k_{row}$ & $k_{ro(o)}$	$k_{rog}$ & $k_{rwg}^*$
*INTMED2	"	"	"
*INTMED3	$k_{rw(w)}$ & $k_{rwo(o)}$	$k_{row(w)}$ & $k_{ro(o)}$	"

\*Unless overwritten by \*WATERGAS

---

## **Rock-Fluid Property Identifier (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.

---

## Rock Type Number for Rock-Fluid Data

\*RPT, \*KRTYPE, \*RTYPE,

\*KRTYPE\_VERT

### PURPOSE:

Define rock type number for rock-fluid data, and assign rock type number to grid blocks.

### FORMAT:

\*RPT *nrock* ( \*COPY *old\_nrock* ) ( *midcalc\_opt* ) ( *wet\_opt* )

### ARRAY:

\*KRTYPE or \*RTYPE

\*KRTYPE\_VERT

### DEFINITIONS:

*nrock*

Rock type number of the following rock-fluid data. The starting value is 1, and need not be specified explicitly.

Succeeding rock type numbers MUST be increasing by 1, etc., 1, 2, 3.

\*COPY *old\_nrock*

Initialize rock type number *nrock* with the data currently assigned to previously defined number *old\_nrock*. This is useful when different rock types are the same except for a few properties.

*midcalc\_opt*

Specify option for calculating the middle phase from the two 2-phase tables.

\*STONE2: Stone's second model (normalized). This is the default.

\*STONE1: Stone's first model (normalized). The \*NOSWC option for \*SLT is unavailable with \*STONE1.

\*LININTERP: Linear interpolation scheme described by L. E. Baker in "Three-Phase Relative Permeability Correlations", SPE/DOE paper 17369.

*wet\_opt*

Specify wettability option.

\*WATWET: Rock is water-wet and oil is the middle phase. This is the usual choice and the default.

\*OILWET: Rock is oil-wet and water is the middle phase.

\*INTMED1: Half the pores are water-wet and the other half are oil-wet.

\*INTMED2: Obtain  $k_{rw}$  assuming oil-wet and  $k_{ro}$  assuming water-wet.

\*INTMED3: Obtain  $k_{rw}$  and  $k_{ro}$  from an application of Stone's model to two-phase  $k_{rwo}$  and  $k_{row}$  obtained by averaging water-wet and oil-wet values.

## **\*KRTYPE, \*RTYPE**

Assign a rock-fluid rock type number to each grid block. The only rock type numbers allowed are 1 and those defined via \*RPT. Alias \*RTYPE can be used for compatibility with other CMG simulators.

## **\*KRTYPE\_VERT**

Assign a rock-fluid rock type number to each grid block for flow in the vertical direction. The only rock type numbers allowed are 1 and those defined via \*RPT.

If \*KRTYPE\_VERT is present then use \*KRTYPE to apply rock type numbers to flow in the horizontal (I and J) directions.

Subkeywords \*HORIZONTAL and \*VERTICAL are obsolete. Replace \*KRTYPE \*HORIZONTAL with \*KRTYPE and replace \*KRTYPE \*VERTICAL with \*KRTYPE\_VERT.

## **DEFAULTS:**

If \*KRTYPE is absent, \*KRTYPE \*CON 1 is assumed. If subkeyword \*IJK is used, then any non-discretized wellbore block that is not referred to explicitly will be assigned the value 1.

If a discretized wellbore grid is present in the data, an additional rock type is created with an *nrock* value equal to the maximum user value plus 1 and data corresponding to pipe flow. Each discretized wellbore block that is not explicitly assigned a value via \*KRTYPE is assigned to this internal rock type.

If middle-phase calculation specifier *midcalc\_opt* is absent, \*STONE2 is assumed.

If wettability specifier *wet\_opt* is absent, \*WATWET is assumed.

If \*KRTYPE\_VERT is absent, then rock types assigned via \*KRTYPE are applied to all flow directions. If subkeyword \*IJK is used, then any non-discretized wellbore block that is not referred to explicitly will be assigned the value 1.

## **CONDITIONS:**

Rock type number *nrock* is applied to the data following it until overwritten by another value. The only intermediate-wet option that can be used with \*LININTERP is \*INTMED2.

## **EXPLANATION:**

See “Multiple Sets of Rock-Fluid Data” at the beginning of this chapter for more details on the organization and use of keywords with multiple rock types.

### **Middle-Phase Option \*LININTERP**

This option requires that the wetting phase relative permeability entries in the \*SWT table be equal to the corresponding liquid relative permeability entries in the \*SLT table, between the critical saturations. If they are not, entries are inserted by interpolation to satisfy the condition. The expanded tables must fit within the allowed table dimensions.

For example, consider the water-wet case where the \*SWT table has entry  $k_{\text{row}} = 0.85$ . If the \*SLT table has an entry with  $k_{\text{rog}} = 0.85$ , then no action is taken. If there is no such  $k_{\text{rog}}$  entry, then all the columns in \*SLT ( $S_i$ ,  $k_{\text{rg}}$ , etc.) are interpolated to get a  $k_{\text{rog}} = 0.85$  entry. The same will be done to the columns in the \*SWT table to get a  $k_{\text{row}}$  entry equal to a  $k_{\text{rog}}$  entry entered as data.

The interpolation of \*LININTERP is not related to the inter-set interpolation of \*INTCOMP, but they can be used together.

### **Discretized Wellbores**

Discretized wellbore blocks usually need rock-fluid data that corresponds to pipe flow, that is, relative permeability curves that are straight lines with zero critical and connate saturations and unity end-points, and zero capillary pressures. A default is provided that corresponds to this usual case.

If a discretized wellbore grid is present in the data, an additional rock type is created with an  $n_{\text{rock}}$  value equal to the maximum user value plus 1 and data corresponding to pipe flow. Then, each discretized wellbore block that is not explicitly assigned a rock type value via \*KRTYPE is assigned to this internal rock type. In addition, explicit reference may be made to this pipe-flow rock type via the data entered for \*KRTYPE.

For example, assume that there are 8 user-specified rel perm rock types and that there is a discretized wellbore grid. In this case the pipe-flow rock type is created as rock type #9, and any wellbore blocks that are not assigned rock types via \*KRTYPE will be assigned to #9.

---

## **CounterCurrent Rock Type Data**

**\*KRTYPE\_CTRWAT,**

**\*KRTYPE\_CTROIL, \*KRTYPE\_CTRGAS**

### **PURPOSE:**

Specify rock-fluid sets to use to model countercurrent flow.

### **ARRAY:**

- \*KRTYPE\_CTRWAT**
- \*KRTYPE\_CTROIL**
- \*KRTYPE\_CTRGAS**

### **DEFINITIONS:**

#### **\*KRTYPE\_CTRWAT**

Assign a rock-fluid rock type number to each grid block for flow in the water countercurrent direction. The only rock type numbers allowed are 1 and those defined via \*RPT.

If \*KRTYPE\_CTRWAT is present then use \*KRTYPE to apply rock type numbers to flow in the cocurrent direction.

#### **\*KRTYPE\_CTROIL**

Assign a rock-fluid rock type number to each grid block for flow in the oil countercurrent direction. The only rock type numbers allowed are 1 and those defined via \*RPT.

If \*KRTYPE\_CTROIL is present then use \*KRTYPE to apply rock type numbers to flow in the cocurrent direction.

#### **\*KRTYPE\_CTRGAS**

Assign a rock-fluid rock type number to each grid block for flow in the gas countercurrent direction. The only rock type numbers allowed are 1 and those defined via \*RPT.

If \*KRTYPE\_CTRGAS is present then use \*KRTYPE to apply rock type numbers to flow in the cocurrent direction.

### **DEFAULTS:**

If \*KRTYPE\_CTRWAT, \*KRTYPE\_CTROIL, \*KRTYPE\_CTRGAS are absent, then rock types assigned via \*KRTYPE are applied to all flow directions. If subkeyword \*IJK is used, then any non-discretized wellbore block that is not referred to explicitly will be assigned the value 1.

## **EXPLANATION:**

### **Cocurrent and Countercurrent Relative Permeabilities**

Some processes (SAGD, VAPEX) involve the dynamic evolution of coupled cocurrent and countercurrent flows. Since the relative permeabilities for these flow conditions are experimentally different (typically countercurrent relative permeabilities are less than cocurrent relative permeabilities), a simulation needs a way to dynamically test and shift between cocurrent and countercurrent relative permeabilities in every region (every grid cell) during the simulation. These keywords implement this capability. See the discussion in Yuan et. al., CIM paper 2001-2002.

If coupled cocurrent-counterflow is to be modelled, additional rock types are created for each countercurrent flow situation (water countercurrent, oil countercurrent, gascountercurrent) which are used in conjunction with the standard cocurrent flow situation specified by the default \*KRTYPE keyword. Then, the simulator internally checks (for each grid block and at each time) the current flow situation and uses the appropriate combination of relative permeability curves.

These keywords allow you to specify different sets of countercurrent relative permeabilities in different regions. In addition, you may specify interpolation of cocurrent/countercurrent relative permeabilities as a function of a specified quantity, e.g., component composition via \*INTCOMP.

---

## Interpolation Component

\*INTCOMP

**PURPOSE:**

Indicate interpolation component.

**FORMAT:**

\*INTCOMP *comp\_name phase*

**DEFINITIONS:**

*comp\_name*

Quoted name of component upon whose composition the rock-fluid interpolation will depend.

*phase*

Keyword indicating the phase from which the component's composition will be taken:

\*WATER water (aqueous) mole fraction

\*OIL oil (oleic) mole fraction

\*GAS gas mole fraction

\*GLOBAL global mole fraction

\*MAX maximum of water, oil and gas mole fractions

**DEFAULTS:**

If \*INTCOMP is absent, interpolation will NOT be enabled.

**CONDITIONS:**

Unless \*IFTTABLE is present, it is assumed that the interpolation parameters \*DTRAPW and \*DTRAPN correspond to the mole fraction defined via \*INTCOMP.

**EXPLANATION:**

See overview in \*ROCKFLUID manual entry.

---

## Interfacial Tension

\*INTLIN, \*INTLOG, \*IFTTABLE

### PURPOSE:

Define interfacial tension and interpolation functions.

### FORMAT:

\*IFTTABLE

cift	sigift
:	:

-or-

\*IFTTABLE

*TEMP	temp
	cift
	sigift
	:

*TEMP	temp
	cift
	sigift
	:

etc.

-or-

\*IFTTABLE

( \*2CMPW | \*2CMPX | \*2CMPY ) 2conc

cift	sigift
:	:

( \*2CMPW | \*2CMPX | \*2CMPY ) 2conc

cift	sigift
:	:

etc.

\*INTLIN

\*INTLOG

### DEFINITIONS:

cift

Composition of component/phase given by \*INTCOMP. The sigift-versus-cift must have at least two entries. Put each entry on a new line. The maximum allowed number of entries for each table is 20.

sigift

Interfacial tension (dyne/cm).

\*TEMP temp

Temperature of sigift-versus-cift table (C | F). Temperature dependence is optional. The maximum allowed number of \*TEMP tables is 10.

2conc

Concentration value of second component affecting the sigift-versus-cift table for component \*INTCOMP. The additional concentration dependence is optional. Used with \*2CMPW, \*2CMPX or \*2CMPY the concentration refers to component NUMW in the aqueous phase, component NUMX in the oleic phase or component NUMY in the gaseous phase, respectively. The maximum allowed number of second-concentration tables is 10.

\*INTLIN

Linear interpolation is used when doing a table lookup of composition values cift. This is the default.

\*INTLOG

Logarithmic interpolation is used when doing a table lookup of composition values cift.

#### **DEFAULTS:**

If \*IFTTABLE is absent, then it is assumed that the interpolation parameters \*DTRAPW and \*DTRAPN correspond to the mole fraction itself defined via \*INTCOMP.

If the subkeyword \*TEMP is absent, the interfacial tension is assumed to be independent of temperature. \*TEMP can be absent for thermal as well as isothermal runs.

\*INTLIN is the default until overridden by \*INTLOG.

#### **EXPLANATION:**

See overview in \*ROCKFLUID manual entry.

---

**Basic Foam Interpolation Parameters** \*FMSURF, \*FMCAP, \*FMOIL,  
\*FMGCP, \*FMOMF, \*FMSALT, \*FMMOB, \*EPSURF, \*EPCAP, \*EPOIL, \*EPGCP, \*EPOMF,  
\*EPSALT, \*FLOIL, \*FLSALT

**PURPOSE:**

Assign basic foam interpolating parameters.

**FORMAT:**

*FMSURF	<i>fmsurf</i>
*FMCAP	<i>fmcap</i>
*FMOIL	<i>fmoil</i>
*FMGCP	<i>fmgcp</i>
*FMOMF	<i>fmomf</i>
*FMSALT	<i>fmsalt</i>
*FMMOB	<i>fmmob</i>
*EPSURF	<i>epsurf</i>
*EPCAP	<i>epcap</i>
*EPOIL	<i>epoil</i>
*EPGCP	<i>epgcp</i>
*EPOMF	<i>epomf</i>
*EPSALT	<i>epsalt</i>
*FLOIL	<i>floil</i>
*FLSALT	<i>flsalt</i>

**DEFINITIONS:**

*fmsurf*

Critical component mole fraction value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

*fmcap*

Reference rheology capillary number value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

*fmoil*

Critical oil saturation value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

*fmgcp*

Critical generation capillary number value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

*fmomf*

Critical oil mole fraction for component *numx* used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.

<i>fmsalt</i>	Critical salt mole fraction value (component <i>numw</i> ) used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.
<i>fmmob</i>	Reference foam mobility reduction factor used in dimensionless foam interpolation calculation. The minimum allowed value is 0, and the suggested maximum is 100,000.
<i>epsurf</i>	Exponent for composition contribution to dimensionless foam interpolation calculation. The allowed range is -4 to 4. The default is 0, which makes foam interpolation independent of composition.
<i>epcap</i>	Exponent for capillary number contribution to dimensionless foam interpolation calculation. The allowed range is -10 to 10. The default is 0, which makes foam interpolation independent of capillary number.
<i>epoil</i>	Exponent for oil saturation contribution to dimensionless foam interpolation calculation. The allowed range is 0 to 5. The default is 0, which makes foam interpolation independent of oil saturation.
<i>epgcp</i>	Exponent for generation cap. no. contribution to dimensionless foam interpolation calculation. The allowed range is -10 to 10. The default is 0, which makes foam interpolation independent of capillary number.
<i>epomf</i>	Exponent for oil mole fraction contribution to dimensionless foam interpolation calculation. The allowed range is 0 to 5. The default is 0, which makes foam interpolation independent of oil mole fraction of component <i>numx</i> .
<i>epsalt</i>	Exponent for salt contribution to dimensionless foam interpolation calculation. The allowed range is -4 to 4. The default is 0, which makes foam interpolation independent of composition of component <i>numw</i> .
<i>floil</i>	Lower oil saturation value used in dimensionless foam interpolation calculation. The allowed range is 0 to 1.
<i>flsalt</i>	Lower salt mole fraction value (component <i>numw</i> ) used in dimensionless foam interpolation calculation. The allowed range is 0 to 1

## DEFUALTS:

If all of these keywords are absent, then it is assumed that the interpolation parameters \*DTRAPW and \*DTRAPN correspond to the interfacial tension (capillary number) option defined via \*IFTTABLE if present, and the mole fraction defined via \*INTCOMP if not.

## CONDITIONS:

The foam interpolation option is possible only if both \*INTCOMP and \*IFTTABLE are present.

## EXPLANATION:

See overview in \*ROCKFLUID manual entry.

The basic foam interpolation option interpolates between sets of relative permeability curves via the dimensionless interpolation factor

$$FM = \frac{1}{1 + FMMOB * F1 * F2 * F3 * F4 * F5 * F6}$$

where

F1	=	( MOLE FRACTION(ICPREL) / FMSURF )	** EPSURF,
F2	=	(( FMOIL - OIL SATURATION ) / (FMOIL-FLOIL))	** EPOIL,
F3	=	( FMCAP / CAPILLARY NUMBER )	** EPCAP,
F4	=	(( FMGCP - CAPILLARY NUMBER ) / FMGCP)	** EPGCP,
F5	=	(( FMOMF - OIL MOLE FR.(NUMX) ) / FMOMF)	** EPOMF,
F6	=	(( MOLE FRACTION(NUMW) -FLSALT) / (FMSALT -FLSALT))	** EPSALT

The factor FM is an inverse mobility reduction factor which varies between FM = 1 (no foam) and FM << 0 (strongest foam).

The reference foam mobility reduction factor FMMOB is that achieved at measured values of surfactant concentration FMSURF, capillary number (flow rate) FMCAP above two times FMGCP, zero oil saturation, and zero oil mole fraction of component *numx*. The normal range of FMMOB is 5 to 100, depending on the strength of created foam.

The remaining factors account empirically for the effects of surfactant concentration, the positive effect of salt (component *numw*), the detrimental effect of oil and oil composition (component *numx*), and flow velocity (both generation and shear thinning effects) on foam mobility, essentially scaling FMMOB. Typical values are FMSURF = 0.00001 mole fraction, FMCAP = 0.0001, and FMOIL = 0.2, while EPSURF = 1, EPCAP = 0.5, and EPOIL = 1 are appropriate exponents. Additionally FMGCP = 1.0e-6 and FMOMF = 0.2, while EPGCP = 1 and EPOMF = 1. Setting an exponent value to zero disables the corresponding contribution.

The simplest application of the foam interpolation option is to rescale gas relative permeability, that is, from  $k_{rg}$  to  $FM \cdot k_{rg}$ . To account for increased gas trapping with foam, employ higher critical gas saturation in the input foam relative permeability curves.

---

## **Interpolation Set Number and Parameters**

\*KRINTRP, \*KRINTERP,  
\*DTRAPW, \*DTRAPN, \*WCRV, \*OCRV, \*GCRV, \*SCRV

### **PURPOSE:**

Indicate interpolation set number along with its value of interpolating parameter.

### **FORMAT:**

\*KRINTRP *nset* (\*COPY *old\_rock old\_set*) (*wet\_opt*)  
\*KRINTERP *nglobset* (\*COPY *old\_rock old\_set*)

*DTRAPW	<i>dtrapw</i>
*DTRAPN	<i>dtrapn</i>
*WCRV	<i>wcrv</i>
*OCRV	<i>ocrv</i>
*GCRV	<i>gcrv</i>
*SCRV	<i>scrv</i>

### **DEFINITIONS:**

#### **\*KRINTRP *nset***

Interpolation set number, local to the current rock-fluid rock type. Values start at 1 for each new rock type and increase by 1 for each additional interpolation set. For example, rock type #1 might have local set numbers 1 and 2 while rock type #2 might have local set numbers 1, 2 and 3.

#### **\*KRINTERP *nglobset***

Interpolation set number, global over all rock-fluid rock types. Values must start at 1 and increase by 1 for each additional interpolation set. For example, rock type #1 might have global set numbers 1 and 2 while rock type #2 might have global set numbers 3, 4 and 5. **This keyword is considered obsolete and \*KRINTRP should be used instead.**

#### **\*COPY *old\_rock old\_set***

Optional keyword that allows a rock-fluid data set to be defined incrementally from another previously defined set.

*old\_rock* is a rock type number previously defined via \*RPT. If rock type #1 is not explicitly defined via \*RPT then it is implied by default.

*old\_set* is a local interpolation set number previously defined via \*KRINTRP. To copy the single set of rock-fluid data from a rock type without \*KRINTRP or \*KRINTERP, use *old\_set* = 1. To copy a set defined via \*KRINTERP, see EXPLANATION below.

*wet\_opt*

Keyword specifying a wettability option for this interpolation set. Choose from the wettability subkeyword list described for keyword \*RPT.

*dtrapw*

Value of wetting phase interpolation parameter for current rock-fluid data set. At least one of *dtrapw* and *dtrapn* must be provided to enable interpolation. Physical meaning of *dtrapw* depends on interpolation option.

*dtrapn*

Value of non-wetting phase interpolation parameter for current rock-fluid data set. At least one of *dtrapw* and *dtrapn* must be provided to enable interpolation. Physical meaning of *dtrapn* depends on interpolation option.

*wcrv*

Curvature change parameter for water relative permeability.

*ocrv*

Curvature change parameter for oil relative permeability.

*gcrv*

Curvature change parameter for gas relative permeability.

*scrv*

Curvature change parameter for liquid relative permeability.

## DEFUALTS:

For a rock type, if \*KRINTRP and \*KRINTERP are absent then there is no rock-fluid interpolation, but for \*COPY purposes the data is accessible as interpolation set #1.

If *wet\_opt* is absent, the wettability option is assumed to be:

- nset = 1* what was specified for the rock type via \*RPT or its default,
- nset > 1* what was specified for *nset* = 1.

At least one of \*DTRAPW and \*DTRAPN must be present to enable interpolation. If only one is present, its value is applied to the absent keyword.

Each of \*WCRV, \*OCRV, \*GCRV and \*SCRV default to 1 if absent.

## CONDITIONS:

At least two sets of rock-fluid data must be present before interpolation can be done. The minimum requirement for interpolation inside a rock type is:

```

*INTCOMP . . . ** Interpolation component

*KRINTRP 1
*DTRAPW . . .
*SWT . . .
*SLT . . .

*KRINTRP 2 *COPY 1 1
*DTRAPW . . . ** Set #2, different *SWT only
*SWT . . .

```

### **EXPLANATION:**

See 'INTERPOLATION OF RELATIVE PERMEABILITY AND CAPILLARY PRESSURE' in this section's overview.

### **Example**

The following illustrative sketch of keywords shows four rock types, three of which have interpolation. Actual rock-fluid data (e.g., \*SWT and \*SLT) is denoted with "...".

```

*RPT 1
  *KRINTRP 1
  ... ** Base data for rock type #1, set #1

  *KRINTRP 2 *COPY 1 1
  ... ** Changes for rock type #1, set #2

  *KRINTRP 3 *COPY 1 1
  ... ** Changes for rock type #1, set #3

*RPT 2
... ** Base data for rock type #2 (set #1)

*RPT 3
  *KRINTRP 1 *COPY 2 1
  ... ** Changes for rock type #3, set #1

  *KRINTRP 2 *COPY 3 1
  ... ** Changes for rock type #3, set #2

  *KRINTRP 3 *COPY 3 2
  ... ** Changes for rock type #3, set #3

*RPT 4
  *KRINTRP 1 *COPY 1 2
  ... ** Changes for rock type #4, set #1

  *KRINTRP 2 *COPY 4 1
  ... ** Changes for rock type #4, set #2

  *KRINTRP 3 *COPY 4 1
  ... ** Changes for rock type #4, set #3

```

For rock type #1, set #1 is fully defined and sets #2 and #3 are defined incrementally from it. Rock type #2 has no interpolation. For rock type #3, set #1 is based on the data from rock type #2, set #2 is defined incrementally from set #1 and set #3 is defined incrementally from set #2. For rock type #4, set #1 is defined incrementally from set #2 of rock type #1 and sets #2 and #3 are defined incrementally from it.

### **Obsolete keyword \*KRINTERP**

Keyword \*KRINTERP uses global interpolation set number *nglobset* which is inconsistent with *old\_set* and can be difficult to convert. Also, \*KRINTERP made it difficult to mix interpolating and non-interpolating rock types (see example above). Therefore, this keyword is considered obsolete and supported for it will be removed in a future version.

To help convert from \*KRINTERP to \*KRINTRP, the following data fragment shows the global set numbers needed for the first three rock types of the above example.

```
*RPT 1
  *KRINTERP 1
  *KRINTERP 2 *COPY 1 1
  *KRINTERP 3 *COPY 1 1
*RPT 2
  *KRINTERP 4
*RPT 3
  *KRINTERP 5
  *KRINTERP 6 *COPY 3 1
  *KRINTERP 7 *COPY 3 2
```

---

## Water-Oil Relative Permeability Table

\*SWT

### PURPOSE:

Define the water-oil relative permeability table.

### TABLE:

\*SWT (\*SMOOTHEND {\*LINEAR | \*QUAD | \*CUBIC} ) (\*PCGW)  
{ S<sub>w</sub> k<sub>rw</sub> k<sub>row</sub> (P<sub>cgw</sub>) ( P<sub>cow</sub> ( P<sub>cowi</sub> ) ) }

### DEFINITIONS:

\*SMOOTHEND {\*LINEAR | \*QUAD | \*CUBIC}

Optional keyword indicating what type of interpolation is to be used for the table intervals where k<sub>rw</sub> and k<sub>row</sub> go from zero to non-zero. Use subkeyword \*LINEAR for linear interpolation, \*QUAD for quadratic interpolation and \*CUBIC for cubic interpolation. If \*SMOOTHEND is absent then \*LINEAR is used.

\*PCGW

Specifies that column 4 of the \*SWT table is gas/water capillary pressure P<sub>cgw</sub>.

S<sub>w</sub>

Water saturation. The allowed range is 0 to 1. S<sub>w</sub> table entries must be increasing. The minimum allowed difference between S<sub>w</sub> entries is 1e-5.

If a water zone is present the last table entry should have S<sub>w</sub> = 1, with k<sub>rw</sub> = 1 and k<sub>row</sub> = 0.

k<sub>rw</sub>

Relative permeability to water at S<sub>w</sub>. The first entry must be 0, and k<sub>rw</sub> entries must be non-decreasing to a maximum of 1. The last k<sub>rw</sub> must be greater than zero. S<sub>rw</sub> is obtained from the last entry with k<sub>rw</sub> = 0.

This column is interpreted as k<sub>rw</sub> for \*WATWET, k<sub>rwo</sub> for \*OILWET and both k<sub>rw</sub> and k<sub>rwo</sub> for intermediate wettability. See section **Wettability Options** at the beginning of this chapter.

k<sub>row</sub>

Relative permeability to oil at S<sub>w</sub>. The first entry must be greater than zero but not exceed 1, and k<sub>row</sub> entries must be non-increasing. The last entry must be zero. S<sub>row</sub> is obtained from the first entry with k<sub>row</sub> = 0.

This column is interpreted as k<sub>row</sub> for \*WATWET, k<sub>ro</sub> for \*OILWET and both k<sub>row</sub> and k<sub>ro</sub> for intermediate wettability. See section **Wettability Options** at the beginning of this chapter.

$P_{cgw}$

Gas-water capillary pressure  $P_g - P_w$  (kPa | psi) in column 4, if \*PCGW is present.  $P_{cgw}$  entries must be non-increasing (i.e., decreasing or equal) with increasing  $S_w$ . Generally,  $P_{cgw}$  should be larger than  $P_{cow}$ .

Use  $P_{cgw}$  to initialize a gas/water transition zone when \*TRANZONE is used in the Initial Conditions data section. Use  $P_{cgw}$  also to account for gas/water interfacial forces when oil phase disappears in a three-phase system.

$P_{cow}$

Water-oil capillary pressure  $P_o - P_w$  (kPa | psi) on the drainage curve.  $P_{cow}$  entries must be non-increasing (i.e., decreasing or equal) with increasing  $S_w$ . When this column is absent its entries are assumed to be zero and no capillary pressure effects are included in the simulation. This column is required if column  $P_{cowi}$  is entered. By default,  $P_{cow}$  will be used in determining the saturation distribution when the \*VERTICAL \*DEPTH\_AVE option is used.

$P_{cowi}$

Water-oil capillary pressure  $P_o - P_w$  (kPa | psi) on the imbibition curve.  $P_{cowi}$  entries must be non-increasing (i.e., decreasing or equal) with increasing  $S_w$ . If this column is absent then it is assumed that there is no hysteresis of capillary pressure. Column  $P_{cowi}$  cannot appear without column  $P_{cow}$ .

For \*WATWET,  $P_{cowi}$  entries must not exceed the corresponding drainage curve value and  $P_{cowi}$  curve must meet the drainage curve,  $P_{cow}$ , at the irreducible water saturation,  $S_{wc}$ . If  $P_{cowi}$  curve from lab is ended at maximum trapped oil saturation, the curve should be extended to the rest table entries with the equal  $P_{cowi}$  value at the maximum trapped oil saturation. STARS will determine the maximum trapped saturation by first  $P_{cowi}$  unchanging condition.

For \*OILWET, however,  $P_{cowi}$  values must not less than the corresponding drainage curve value and the  $P_{cowi}$  curve must meet the drainage curve at the irreducible oil saturation,  $S_{oc}$ . If  $P_{cowi}$  curve from lab starts at maximum trapped water saturation, the curve should be extended to the rest table entries with the equal  $P_{cowi}$  value at the maximum trapped water saturation. STARS will determine the maximum trapped saturation by first  $P_{cowi}$  changing condition.

## CONDITIONS:

At least one \*SWT table must be entered, and it must appear before \*SLT.

Entries must be in order of increasing water saturation.

The maximum number of rows allowed in this table is 100.

For the size of the mobile region  $1-S_{wcrit}-S_{orw}$ , the minimum allowed value is 0.02 and the minimum recommended value is 0.3. These conditions are applied for all temperatures, all interpolation sets and all per-block end-point values.

To enable the capillary pressure hysteresis option, some other primary keywords must be entered along with  $P_{cowi}$ . For further discussion of how to apply this feature, refer to the section “**Hysteresis Parameters**”.

This table must have either 3 columns ( $S_w \ k_{rw} \ k_{row}$ ), 4 columns ( $S_w \ k_{rw} \ k_{row} \ P_{cow}$ ) or 5 columns ( $S_w \ k_{rw} \ k_{row} \ P_{cow} \ P_{cowi}$ ).

### The \*LININTERP Option

This option requires that the wetting phase relative permeability entries in the \*SWT table be equal to the corresponding liquid relative permeability entries in the \*SLT table, between the critical saturations. If they are not, entries are inserted by interpolation to satisfy the condition. The expanded tables must fit within the allowed table dimensions.

For example, consider the water-wet case where the \*SWT table has entry  $k_{row} = 0.85$ . If the \*SLT table has an entry with  $k_{rog} = 0.85$ , then no action is taken. If there is no such  $k_{rog}$  entry, then all the columns in \*SLT ( $S_l, k_{rg}$ , etc.) are interpolated to get a  $k_{rog} = 0.85$  entry. The same will be done to the columns in the \*SWT table to get a  $k_{row}$  entry equal to a  $k_{rog}$  entry entered as data.

## Liquid-Gas Relative Permeability Table

\*SLT

### PURPOSE:

Define the liquid-gas relative permeability table.

### TABLE:

```
SLT (*NOSWC) (smooth)
{ Sl krg krog ( Pcog ( Pcogi ) ) }
-or-
*SLT (*NOSWC) (smooth) *WATERGAS
{ Sl krg krog krwg ( Pcog ( Pcogi ) ) }
smooth = *SMOOTHEND { *LINEAR | *QUAD | *CUBIC }
```

### DEFINITIONS:

#### \*NOSWC

Table liquid saturation  $S_l$  does not contain connate water saturation  $S_{wc}$ , and therefore is all oil. The \*STONE1 option is unavailable with \*NOSWC.

#### \*SMOOTHEND { \*LINEAR | \*QUAD | \*CUBIC }

Optional keyword indicating what type of interpolation is to be used for the table intervals where  $k_{rg}$  and  $k_{rog}$  ( $k_{rwg}$ ) go from zero to non-zero. Use subkeyword \*LINEAR for linear interpolation, \*QUAD for quadratic interpolation and \*CUBIC for cubic interpolation. If \*SMOOTHEND is absent then \*LINEAR is used.

#### \*WATERGAS

Flag to indicate that the  $k_{rwg}$  column is expected. This keyword is needed only for specifying  $k_{rwg}$  different from  $k_{row}$  for an intermediate wettability option. See section **Wettability Options** at the beginning of this chapter.

#### $S_l$

Liquid saturation. The allowed range is 0 to 1.  $S_l$  table entries must be increasing. The minimum allowed difference between  $S_l$  entries is 1e-5.

The last  $S_l$  value should be 1. If  $S_l = 1$  is not present, this entry is added automatically.

#### $k_{rg}$

Relative permeability to gas at  $S_l$ . The first entry must be greater than zero but not exceed 1, and  $k_{rg}$  entries must be non-increasing. The last entry must be 0.  $S_{gc}$  is obtained from the first entry with  $k_{rg} = 0$ .

$k_{rog}$	Relative permeability to oil (and wetting water) at $S_l = S_o + S_w$ . The first entry must be 0, and table entries must be non-decreasing. The last entry must be greater than zero but not exceed 1. $S_{lc}$ is obtained from the last entry with $k_{rog} = 0$ . The last $k_{rog}$ entry must be equal to the first $k_{row}$ entry in the *SWT table ( $k_{row}(S_{wc})$ without *NOSWC, $k_{row(o)}$ with *NOSWC).
	This column is interpreted as $k_{rog}$ for *WATWET and $k_{rwg}$ for *OILWET. For intermediate wettability it is interpreted as both $k_{rog}$ and $k_{rwg}$ unless $k_{rwg}$ is specified explicitly via *WATERGAS. See section <b>Wettability Options</b> at the beginning of this chapter.
$k_{rwg}$	Relative permeability to water (and wetting oil) at $S_l = S_w + S_o$ . The first entry must be 0, and table entries must be non-decreasing. The last entry must be greater than zero, but not exceed 1. $S_{lc}$ is obtained from the last entry with $k_{rwg} = 0$ . The last $k_{rwg}$ entry must be equal to $k_{rwo}(S_{oc})$ . If the last $S_l$ is not 1, another table entry to this effect is added. See *WATERGAS, above.
$P_{cog}$	Gas-oil capillary pressure $P_g - P_o$ (kPa   psi) on the drainage curve. Entries must be non-increasing (i.e., decreasing or equal) with increasing $S_l$ . By default, $P_{cog}$ will determine the saturation distributions when the *VERTICAL *DEPTH_AVE option is used. If column $P_{cog}$ is absent then it is assumed that the entries are zero and there is no gas-oil capillary pressure effect. This column is required if column $P_{cogi}$ is entered.
$P_{cogi}$	Gas-oil capillary pressure $P_g - P_o$ (kPa   psi) on the imbibition curve. $P_{cogi}$ entries must be non-increasing (i.e., decreasing or equal) with increasing $S_l$ and must not exceed the corresponding drainage curve value. The imbibition curve must meet the drainage curve at the connate liquid saturation, $S_{lc}$ . If column $P_{cogi}$ is absent then capillary pressure has no hysteresis effects. If $P_{cogi}$ curve from lab is ended at maximum trapped gas saturation, the curve should be extended to the rest table entries with the equal $P_{cogi}$ value at the maximum trapped gas saturation. STARS will determine the maximum trapped gas saturation by first $P_{cogi}$ unchanging condition.

## DEFUALTS:

If \*NOSWC is absent, it is assumed that liquid saturation  $S_l$  does contain  $S_{wc}$ .

If \*WATERGAS is absent, it is assumed that the  $k_{rwg}$  table is identical to the  $k_{rog}$  table. Thus, you can use \*SLT without \*WATERGAS to define  $k_{rwg}$  when using \*OILWET, etc.

## **CONDITIONS:**

This table must be entered at least once, even if gas is never present, and it must occur after \*SWT, since an endpoint check uses information from \*SWT.

For the size of the mobile region  $1-S_{gcrit}-S_{lrg}$ , the minimum allowed value is 0.02 and the minimum recommended value is 0.3. These conditions are applied for all temperatures, all interpolation sets and all per-block end-point values.

The maximum number of rows allowed in this table is 100.

To enable the gas-oil capillary pressure hysteresis option, some other primary keywords must be entered along with  $P_{cogi}$ . For further discussion of how to apply this feature, refer to the section "Hysteresis Parameters".

Without \*WATERGAS this table must have either 3 columns ( $S_l$   $k_{rg}$   $k_{rog}$ ), 4 columns ( $S_l$   $k_{rg}$   $k_{rog}$   $P_{cog}$ ) or 5 columns ( $S_l$   $k_{rg}$   $k_{rog}$   $P_{cog}$   $P_{cogi}$ ). With \*WATERGAS this table must have either 4 columns ( $S_l$   $k_{rg}$   $k_{rog}$   $k_{rwg}$ ), 5 columns ( $S_l$   $k_{rg}$   $k_{rog}$   $k_{rwg}$   $P_{cog}$ ) or 6 columns ( $S_l$   $k_{rg}$   $k_{rog}$   $k_{rwg}$   $P_{cog}$   $P_{cogi}$ ).

## **EXPLANATION:**

When \*NOSWC option is not used,  $k_{row}$  entries of \*SWT before  $S_{wc}$  must be equal to  $k_{row}(S_{wc})$ , since Stone's models assumes that the endpoint value is  $k_{row}(S_{wc})$ . In this case, the only reason to have table entries for  $S_w < S_{wc}$  is for  $P_{cow}$ . When \*NOSWC is used, this restriction is lifted.

See the section in this chapter's introduction entitled "CRITICAL AND CONNATE SATURATIONS, SCALE-UP FACTORS, AND NORMALIZATION".

---

## **Hysteresis Parameters (Optional)**

\*HYS\_KRO, \*HYS\_KRW, \*HYS\_KRG,  
\*HYS\_PCOW, \*HYS\_PCOG, \*HYS\_LEVEL, \*HYS\_TOLW, \*HYS\_REVW, \*HYS\_TOLG,  
\*HYS\_REVG, \*HYS\_DRAINW, \*HYS\_IMBIBW, \*HYS\_DRAING, \*HYS\_IMBIBG

### **PURPOSE:**

These keywords enable the relative permeability and capillary pressure hysteresis option and signal entry of hysteresis parameters.

### **FORMAT:**

Defined for current interpolation set:

*HYS_KRO	*CARLSON *SOTMAX ( <i>sotmax</i> ) -or- *KILLOUGH *HYEXO ( <i>hyexo</i> ) ( *HYEXW ( <i>hyexw</i> ) ) *SWTI ( <i>S<sub>wi</sub></i> <i>k<sub>ri_nonwet</sub></i> ( <i>k<sub>ri_wet</sub></i> ) ) -or- *BBM *ENWI ( <i>enwi</i> ) *ENWD ( <i>enwd</i> ) *EWTI ( <i>ewti</i> ) *EWTD ( <i>ewtd</i> ) *RNW ( <i>rnw</i> ) *RWT ( <i>rwt</i> ) *SWTI ( <i>S<sub>wi</sub></i> <i>k<sub>ri_nonwet</sub></i> ( <i>k<sub>ri_wet</sub></i> ) )
*HYS_KRW	*CARLSON *SWTMAX ( <i>swtmax</i> ) -or- *KILLOUGH *HYEXW ( <i>hyexw</i> ) ( *HYEXO ( <i>hyexo</i> ) ) *SWTI { <i>S<sub>wi</sub></i> <i>k<sub>ri_nonwet</sub></i> ( <i>k<sub>ri_wet</sub></i> ) } -or- *BBM *ENWI ( <i>enwi</i> ) *ENWD ( <i>enwd</i> ) *EWTI ( <i>ewti</i> ) *EWTD ( <i>ewtd</i> ) *RNW ( <i>rnw</i> ) *RWT ( <i>rwt</i> ) *SWTI ( <i>S<sub>wi</sub></i> <i>k<sub>ri_nonwet</sub></i> ( <i>k<sub>ri_wet</sub></i> ) )
*HYS_KRG	*CARLSON *SGTMAX ( <i>sgtmax</i> ) -or- *KILLOUGH *HYEXG ( <i>hyexg</i> ) *SLTI { <i>S<sub>li</sub></i> <i>k<sub>rgi</sub></i> } -or- *BBM *ENGI ( <i>engi</i> ) *ENGD ( <i>engd</i> ) *RNG ( <i>rng</i> ) *SLTI { <i>S<sub>li</sub></i> <i>k<sub>rgi</sub></i> }
*HYS_PCOW	(*EPSOW <i>epsow</i> )
*HYS_PCOG	(*EPSOG <i>epsog</i> )

Defined for relative permeability and capillary pressure with all interpolation sets:

*HYS_TOLW	<i>tolhyl</i>
*HYS_REVW	<i>tolrel</i>
*HYS_TOLG	<i>tolhyg</i>
*HYS_REVG	<i>tolreg</i>

Defined for capillary pressure with all interpolation sets:

*HYS_LEVEL	<i>nlevel</i>
*HYS_DRAINW	*HYS_IMBIBW   *HYS_DRAINW_2ND
*HYS_DRAING	*HYS_IMBIBG   *HYS_DRAING_2ND

## DEFINITIONS:

### \*HYS\_KRO

With **water** being the wetting phase (\*WATWET), \*HYS\_KRO enables hysteretic effect on the **oil** (non-wetting phase) relative permeability for all three methods and optionally on the **water** (wetting phase) relative permeability for Killough and BBM method.

### \*HYS\_KRW

With **oil** being the wetting phase (\*OILWET), \*HYS\_KRW enables hysteretic effect on the **water** (non-wetting phase) relative permeability for all three methods and optionally on the **oil** (wetting phase) relative permeability for Killough and BBM method.

### \*HYS\_KRG

With **gas** always being treated as the non-wetting phase, \*HYS\_KRG enables hysteretic effect on the **gas** relative permeability for all three methods.

### \*CARLSON

Depending on the rock wettability, the method developed by Carlson is employed in the treatment of **non-wetting** phase relative permeability hysteresis.

### \*KILLOUGH

Depending on the rock wettability, the method developed by Killough is employed in the treatment of **non-wetting** phase, and optionally **wetting** phase, relative permeability hysteresis.

### \*BBM

Depending on the rock wettability, the method developed by Beattie, Boberg and McNab is employed in the treatment of **non-wetting** phase, and optionally **wetting** phase, relative permeability hysteresis.

### \*SOTMAX (*sotmax*)

For the Carlson method only, *sotmax* is the maximum trapped oil (non-wetting phase) saturation of the imbibition curve (See Figure HY1(a)). This is the endpoint of the imbibition curve which breaks off from the oil relative permeability drainage curve at the maximum possible  $k_{row}$  (at  $S_o = 1.0 - S_{wc}$ ).

This value is used to evaluate the shape and path of all scanning curves which leave the drainage curve at any saturation reversal. *sotmax* must be greater than the residual oil saturation  $S_{orw}$  and less than  $(1.0 - S_{wc})$ .

#### \*SWTMAX (*swtmax*)

For the Carlson method only, *swtmax* is the maximum trapped water (non-wetting phase) saturation of the imbibition curve. This is the endpoint of the imbibition curve which breaks off from the water relative permeability drainage curve at the maximum possible  $k_{rwo}$  (at  $S_w = 1.0 - S_{oc}$ ). This value is used to evaluate the shape and path of all scanning curves which leave the drainage curve at any saturation reversal. *swtmax* must be greater than the critical water saturation  $S_{wr}$  and less than  $(1.0 - S_{oc})$ . (See Figure HY4).

#### \*SGTMAX (*sgtmax*)

For the Carlson method only, *sgtmax* is the maximum trapped gas saturation of the imbibition curve. This is the endpoint of the imbibition curve which breaks off from the gas relative permeability drainage curve at the maximum possible  $k_{rg}$  (at  $S_g = 1.0 - S_{lc}$ ). This value is used to evaluate the shape and path of all scanning curves which leave the drainage curve at any saturation reversal. *sgtmax* must be greater than the critical gas saturation  $S_{gc}$  and less than  $(1.0 - S_{lc})$ . A larger value of *sgtmax* will result in a steeper imbibition curve which may potentially cause numerical convergence difficulties.

#### \*HYEXO (*hyexo*)

Used by the Killough method, this dimensionless real number determines the position and curvature of the oil relative permeability scanning curves. For oil non-wetting, the larger the *hyexo* value, the further the scanning curve will be away from the drainage curve (see (EQ h.6)). For oil wet case, increasing *hyexo* tends to make the scanning curves closer to the drainage curve (see (EQ h.12) and EQ (h.13)).

#### \*HYEXW (*hyexw*)

Used by the Killough method, this dimensionless real number determines the position and curvature of the water relative permeability scanning curves. For water non-wetting, the larger the *hyexw* value, the further the scanning curve will be away from the drainage curve. For water wet case, increasing *hyexw* tends to make the scanning curves closer to the drainage curve (see (EQ h.12) and (EQ h.13)).

#### \*HYEXG (*hyexg*)

Used by the Killough method, this dimensionless real number determines the position and curvature of the gas relative permeability scanning curves. The larger the *hyexg* value, the further the scanning curve will be away from the drainage curve.

( \*ENWI *enwi* ) ( \*ENWD *enwd* ) ( \*EWTI *ewti* ) ( \*EWTD *ewtd* )

Used by the BBM method, these dimensionless scanning curve exponents determine how rapidly the non-wetting and/or wetting phase relative permeability scanning curves approach the bounding drainage and imbibition curve after a saturation reversal. *enwi* and *enwd* are designated for non-wetting phase and *ewti* and *ewtd* for wetting phase.

For water wet, when  $S_w$  is increasing, the larger value of *enwi* and *ewti* makes the scanning curve move rapidly toward the imbibition bounding curve (see (EQ h.9) and (EQ h.14)). Rising *enwd* and *ewtd* tends to make the scanning curves closer to the drainage bounding curve when  $S_w$  is decreasing.

For oil wet case, when  $S_w$  is increasing, the larger value of *enwi* and *ewti* makes the scanning curve move rapidly toward the drainage bounding curve. Rising *enwd* and *ewtd* tends to make the scanning curves closer to the imbibition bounding curve when  $S_w$  is decreasing.

( \*ENGI *engi* ) ( \*ENGD *engd* )

Used by the BBM method, these dimensionless scanning curve exponents determine the position and curvature of the gas phase relative permeability scanning curves. The larger the *engi* value, the further the scanning curve will be away from the drainage curve when  $S_{li}$  is increasing. Rising *engd* tends to make the scanning curves closer to the drainage curve when  $S_{li}$  is decreasing.

( \*RNW *rnw* ) ( \*RWT *rwt* )

Used by the BBM method, these dimensionless real numbers determine which curve the simulation will start on for the wetting and non-wetting phase relative permeabilities. If *rnw* ( *rwt* ) = 1.0, the run will start on drainage curve for non-wetting phase (wetting phase) relative permeability. If *rnw* ( *rwt* ) = 0.0, the run will starts on imbibition curve for non-wetting phase (wetting phase) relative permeability. If  $0.0 < rnw$  ( *rwt* )  $< 1.0$ , the run will starts on scanning curve for non-wetting phase (wetting phase) relative permeability.

( \*RNG *rng* )

Used by the BBM method, this dimensionless real number determines which curve the simulation starts on for the gas relative permeability. If *rng* = 1.0, the run will starts on drainage curve. If *rng* = 0.0, the run will starts on imbibition curve. If  $0.0 < rng < 1.0$ , the run will starts on scanning curve.

#### \*SWTI

Indicates the start of the water-oil relative permeability imbibition table.

$S_{wi}$

A column of real numbers represents the water saturation. The allowed range is 0 to 1.  $S_{wi}$  table entries must be increasing. The minimum allowed difference between  $S_{wi}$  entries is 1e-5.

If \*KILLOUGH is invoked, for a water wet system, the first entry must equal to the connate water saturation  $S_{wc}$  from the drainage table and the last entry,  $S_{wi\_last}$ , defines the maximum trapped oil saturation by  $sotmax = 1 - S_{wi\_last}$ ; for an oil wet system, the first entry defines the maximum trapped water saturation,  $swtmax$ , and the last entry must equal to  $1 - S_{orw}$ .

If \*BBM is invoked, the first entry must equal to the connate water saturation  $S_{wc}$  from the drainage table and the last entry must equal to  $1 - S_{orw}$ .

#### $k_{ri\_nonwet}$

A column of real numbers represents the imbibition relative permeability to **non-wetting liquid phase** at  $S_{wi}$ . This column is interpreted as  $k_{rowi}$  (imbibition relative permeability to oil) for \*WATWET,  $k_{rwi}$  (imbibition relative permeability to water) for \*OILWET.

If \*KILLOUGH is invoked, for a **water wet** system, the first entry (of  $k_{rowi}$ ) must equal to the  $k_{row}$  endpoint value in the drainage table, that is, the value at the connate water saturation.  $k_{rowi}$  entries must be smaller than the corresponding drainage table value. The last  $k_{rowi}$  must equal to zero. For an **oil wet** system, water is the non wetting phase. The  $k_{rwi}$  first entry must be 0, and  $k_{rwi}$  entries must be smaller than the corresponding drainage table value. The last  $k_{rwi}$  must equal to the endpoint value in the drainage table.

If \*BBM is invoked,  $k_{rowi}$  or  $k_{rwi}$  must have the same endpoints as its counterpart of the drainage table. For **water wet** system,  $k_{rowi}$  entries must be smaller than the corresponding drainage table value; for **oil wet** system,  $k_{rwi}$  entries must be smaller than the corresponding drainage table value.

#### $(k_{ri\_wet})$

A column of real numbers represents the imbibition relative permeability to the wetting liquid phase. This column is interpreted as  $k_{rwi}$  (imbibition relative permeability to water) for \*WATWET,  $k_{rowi}$  (imbibition relative permeability to oil) for \*OILWET. If this column is absent, then there will be no hysteresis for the wetting phase relative permeability.

If \*KILLOUGH is invoked, for **water wet** system, the first entry must be 0, and  $k_{rwi}$  entries must be larger than the corresponding drainage table value. The last  $k_{rwi}$  must be greater than zero and smaller than the endpoint value in the drainage table. For **oil wet** system, the  $k_{rowi}$  first entry must be greater than zero and smaller than the endpoint value at  $S_{wc}$  in the drainage table.  $k_{rowi}$  entries must be larger than the corresponding drainage table value. The last  $k_{rowi}$  must equal to zero.

If \*BBM is invoked,  $k_{rwi}$  or  $k_{rowi}$  must have the same endpoints as its counterpart of the drainage table. For **water wet** system,  $k_{rwi}$  entries must be larger than the corresponding drainage table value; for **oil wet** system,  $k_{rowi}$  entries must be larger than the corresponding drainage table value.

**\*SLTI**

Indicates the start of the liquid-gas relative permeability imbibition table.

 **$S_{li}$** 

A column of real numbers represents the liquid saturation. The allowed range is 0 to 1.  $S_{li}$  table entries must be increasing. The minimum allowed difference between  $S_{li}$  entries is 1e-5.

If \*KILLOUGH is invoked, the first entry must equal to the connate liquid saturation obtained from the drainage table and the last entry,  $S_{li\_last}$ , defines the maximum trapped gas saturation by  $sgtmax = 1 - S_{li\_last}$ .

If \*BBM is invoked, the first entry must equal to the connate liquid saturation obtained from the drainage table and the last entry must equal to  $1 - S_{gcrit}$ .

 **$k_{rgi}$** 

A column of real numbers represents the imbibition relative permeability to gas.  $k_{rgi}$  entries must be non-increasing.

If \*KILLOUGH is invoked, the first entry must equal to the endpoint value in the drainage table, that is, the value at connate liquid saturation.  $k_{rgi}$  entries must be smaller than the corresponding drainage table value. The last  $k_{rgi}$  must equal to zero.

If \*BBM is invoked,  $k_{rgi}$  must have the same endpoints as  $k_{rg}$  of the drainage table and  $k_{rgi}$  entries must be smaller than the corresponding drainage table value.

**\*HYS\_PCOW**

Denotes that hysteresis effect on the oil-water capillary pressure is modeled.

**\*EPSOW *epsow***

Dimensionless real number which determines the transition between the imbibition and drainage curves for oil-water capillary pressure. Typical values of *epsow* should generally satisfy the expression:  $0.05 \leq \text{epsow} \leq 0.1$ . Values smaller than 0.05 tend to cause numerical convergence problems.

**\*HYS\_PCOG**

Denotes that hysteresis effect on the oil-gas capillary pressure is modeled.

**\*EPSOG *epsog***

Dimensionless real number which determines the transition between the imbibition and drainage curves for oil-gas capillary pressure. Typical values of *epsog* should generally satisfy the expression:  $0.05 \leq \text{epsog} \leq 0.1$ . Values smaller than 0.05 tend to cause numerical convergence problem.

**\*HYS\_TOLW *tolhyl***

Saturation tolerance by which the water (or oil) saturation must exceed the critical values (endpoints), e.g.  $|S_w - S_{wc}| > tolhyl$ , before hysteresis calculations are performed.

**\*HYS\_REVW** *tolrel*

Saturation tolerance by which the water (or oil) saturation must exceed the historical maximum saturation, e.g.  $|S_o - S_{ohmax}| > tolrel$ , before hysteresis calculations are performed.

**\*HYS\_TOLG** *tolhyg*

Saturation tolerance by which the gas saturation must exceed the critical values before hysteresis calculations are performed.

**\*HYS\_REVG** *tolreg*

Saturation tolerance by which the gas saturation must exceed the historical maximum saturation before hysteresis calculations are performed.

**\*HYS\_LEVEL**

Number of levels of scanning curves for capillary pressure hysteresis.  
Allowed values are 1 or 2.

**\*HYS\_DRAINW | \*HYS\_IMBIBW | HYS\_DRAINW\_2ND**

Indicates which water-oil capillary pressure curve to use for vertical equilibrium initialization (\*VERTICAL \*DEPTH\_AVE) and for the onset capillary pressure. \*HYS\_DRAINW denotes the drainage curve and \*HYS\_IMBIBW denotes the imbibition curve. Similar to HYS\_DRAINW, \*HYS\_DRAINW\_2ND denotes that the simulation starts on the drainage curve but has completed primary drainage. At most one of these subkeywords may be entered. If none is entered then \*HYS\_DRAINW is assumed.

**\*HYS\_DRAING | \*HYS\_IMBIBG | HYS\_DRAING\_2ND**

Indicates which gas-oil capillary pressure curve to use for vertical equilibrium initialization (\*VERTICAL \*DEPTH\_AVE) and for the onset capillary pressure. \*HYS\_DRAING denotes the drainage curve and \*HYS\_IMBIBG denotes the imbibition curve. Similar to HYS\_DRAING, \*HYS\_DRAING\_2ND denotes that the simulation starts on the drainage curve but has completed primary drainage. At most one of these subkeywords may be entered. If none is entered then \*HYS\_DRAING is assumed.

**DEFUALTS:**

For water wet system (\*WATWET):

If \*HYS\_KRO is not entered then there is no hysteresis effect on non-wetting oil relative permeability and wetting water relative permeability for this interpolation set.

If \*HYS\_KRO \*CARLSON \*SOTMAX is entered without the number sotmax following it, then

$$sotmax = S_{orw} + 0.5 * (1.0 - S_{orw} - S_{wc})$$

If \*HYS\_KRO \*KILLOUGH is entered without \*HYEXO, the relative permeability interpolation (EQ h.5) will be used in oil relative permeability computations. Otherwise, the saturation interpolation (EQ h.6) will be used. The input of \*HYEXW is mandatory if hysteresis of wetting phase is intended.

If \*HYS\_KRO \*KILLOUGH \*HYEXO \*HYEXW is entered without the real numbers following, then

$$hyexo = hyexw = 1.0$$

If \*HYS\_KRO \*BBM is entered, any missing subkeyword and its associated number will be defaulted to 1.0.

If \*SWTI table contains only two columns, the second column represents  $k_{rowi}$ , and therefore there will be no hysteretic effect on wetting water phase relative permeability for this interpolation set.

For oil wet system (\*OILWET):

If \*HYS\_KRW is not entered then there is no hysteretic effect on non-wetting water relative permeability and wetting oil relative permeability for this interpolation set.

If \*HYS\_KRW \*CARLSON \*SWTMAX is entered without the number *swtmax* following it, then

$$swtmax = S_{wc} + 0.5 * (1.0 - S_{orw} - S_{wc})$$

If \*HYS\_KRW \*KILLOUGH is entered without \*HYEXW, the relative permeability interpolation will be used in water relative permeability computations. Otherwise, the saturation interpolation will be used. The input of \*HYEXO is mandatory if hysteresis of wetting phase is intended.

If \*HYS\_KRW \*KILLOUGH \*HYEXO \*HYEXW is entered without the real numbers following, then

$$hyexo = hyexw = 1.0$$

If \*HYS\_KRW \*BBM is entered, any missing subkeyword and its associated number will be defaulted to 1.0.

If \*SWTI table contains only two columns, the second column represents  $k_{rwi}$ , and therefore there will be no hysteretic effect on wetting oil relative permeability for this interpolation set.

\*SWTI table are mandatory input for Killough and BBM method. There is no default.

If \*HYS\_KRG is not entered then there is no hysteretic effect on gas relative permeability for this interpolation set. If \*HYS\_KRG \*CARLSON \*SGTMAX is entered without the number *sgtmax* following it, then

$$sgtmax = S_{gr} + 0.5 * (1.0 - S_{lr} - S_{gr})$$

If \*HYS\_KRG \*KILLOUGH is entered without \*HYEXG, the relative permeability interpolation will be used in gas relative permeability computations. Otherwise, the saturation interpolation will be used.

If \*HYS\_KRG \*KILLOUGH \*HYEXG is entered without the number following it, then

$$hyexg = 1.0$$

If \*HYS\_KRG \*BBM is entered without the subkeywords and numbers following it, then

$$engi = engd = rng = 1.0$$

\*SLTI table is mandatory input for Killough and BBM method. There is no default.

If \*HYS\_PCOW is not entered then there is no hysteretic effect on oil-water capillary pressure for this interpolation set. If \*HYS\_PCOW is entered without the subkeyword \*EPSOW *epsow*, then *epsow* = 0.1 is assumed. To enable oil-water capillary pressure hysteresis, column P<sub>cowi</sub> must be entered via table keyword \*SWT for this interpolation set.

If \*HYS\_PCOG is not entered then there is no hysteretic effect on oil-gas capillary pressure for this interpolation set. If \*HYS\_PCOG is entered without the subkeyword \*EPSOG *epsog*, then *epsog* = 0.1 is assumed. To enable the oil-gas capillary pressure hysteresis, column P<sub>cogi</sub> must be entered via table keyword \*SLT for this interpolation set.

If \*HYS\_LEVEL is not entered, then \*HYS\_LEVEL 1 is assumed.

If \*HYS\_TOLW is not entered, then \*HYS\_TOLW 1.0e-6 is assumed.

If \*HYS\_REVW is not entered, then \*HYS\_REVW 1.0e-6 is assumed.

If \*HYS\_TOLG is not entered, then \*HYS\_TOLG 1.0e-6 is assumed.

If \*HYS\_REVG is not entered, then \*HYS\_REVG 1.0e-6 is assumed.

If none of \*HYS\_DRAINW, \*HYS\_IMBIBW or HYS\_DRAINW\_2ND is entered, \*HYS\_DRAINW is assumed. However, this may be overridden by HYS\_DRAINW\_2ND if P<sub>cow</sub> and P<sub>cowi</sub> share the same end points.

If none of \*HYS\_DRAING, \*HYS\_IMBIBG or HYS\_DRAING\_2ND is entered, \*HYS\_DRAING is assumed. However, this may be overridden by HYS\_DRAING\_2ND if P<sub>cog</sub> and P<sub>cogi</sub> share the same end points.

### CONDITIONS:

The relative permeability hysteresis option can only be applied to the system that a strong rock wettability exists (either \*WATWET or \*OILWET). These keywords must be in the Rock-Fluid Data keyword group, after the drainage water-oil and the liquid-gas saturation tables (\*SWT and \*SLT) have been entered. It is possible to apply hysteresis to either the relative permeabilities, capillary pressure or both simultaneously. The oil-water capillary pressure hysteresis may be applied to either \*WATWET or \*OILWET case. The gas phase is always treated as the non-wetting phase. The hysteresis option may be applied in combination with other features, such as endpoint over-riding and temperature dependence, but extra caution should be exercised to ensure the consistency of input data to avoid numerical difficulties. Similarly, the combination of hysteresis and table interpolation may introduce new uncertainties if the table and the hysteresis parameters of the two adjacent sets are so much different.

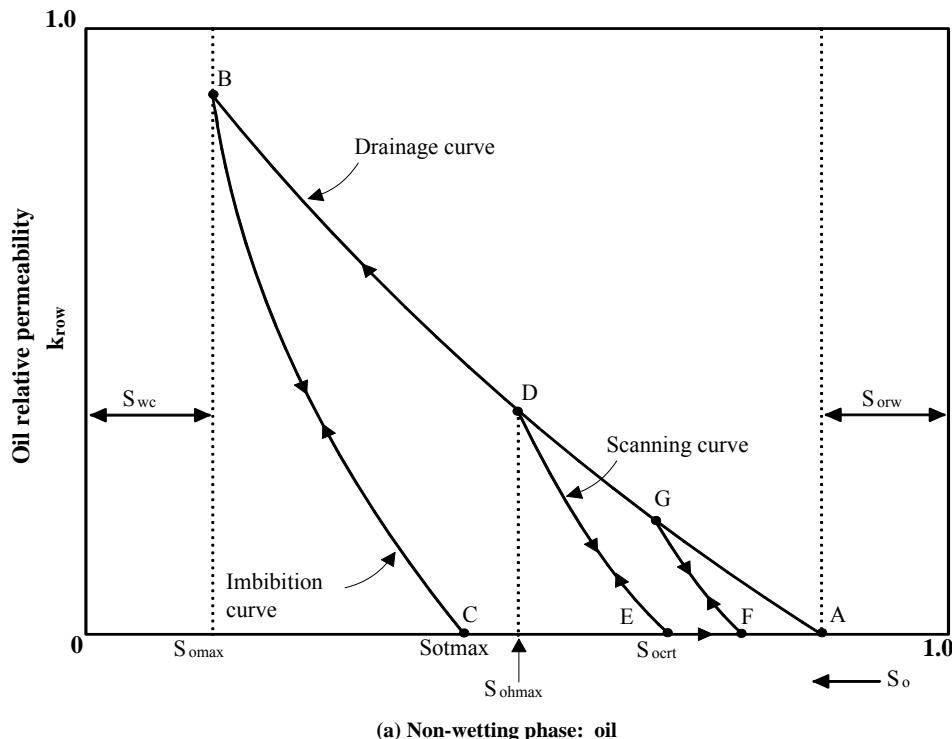
## EXPLANATION:

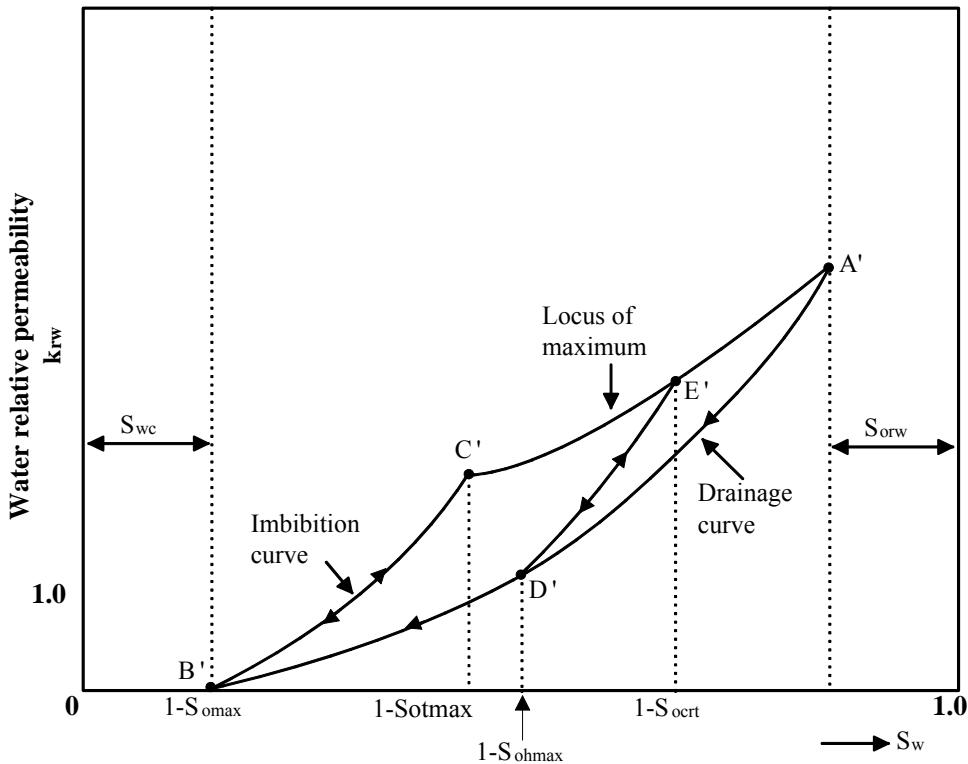
The relative permeability and capillary pressure hysteresis option allows to simulate the history-dependency of saturation functions when saturation changes are not unidirectional. Three methods are available to model non-wetting phase relative permeability hysteresis: Carlson, Killough and BBM and two for wetting phase, Killough and BBM. For a three phase system, the hysteretic values are first obtained independently from both water-oil and liquid-gas systems, these values are then utilized in the calculation of the middle phase (oil for water wet and water for oil wet) relative permeability.

### Non-wetting Phase Relative Permeability Hysteresis

#### The Carlson and Killough method

Following Carlson's (\*CARLSON) approach, (Carlson, F.M., "Simulation of Relative Permeability Hysteresis to the Non-wetting Phase", Paper SPE 10157), Figure HY1(a) illustrates the general nature of hysteresis in the oil relative permeability assuming oil is the non-wetting phase. Analysis and numerical treatment for other non-wetting phases is analogous to oil phase presented here.





(b) Wetting phase: water

*Figure HY1: Hysteretic characteristics of wetting and non-wetting phase relative permeability, Carlson(non-wetting phase only) and Killough method.*

If oil saturation increases monotonically from  $S_{orw}$  (point A) to the maximum oil saturation  $S_{omax} = 1.0 - S_{wc}$  (point B), the drainage curve AB will be followed. If oil saturation then decreases from B all the way to C, the imbibition curve is used. If the drainage or imbibition process is reversed at some point between on AB, the relative permeability will be obtained from a scanning curve.

Suppose a drainage process is reversed at some intermediate oil saturation  $S_{ohmax}$  (point D), a scanning curve DE is created. The end points of a scanning curve are the trapped oil saturation ( $S_{oer}$ ) and the historical maximum oil saturation ( $S_{ohmax}$ ) reached at the moment in the run.

For any state on the scanning curve DE, change back to drainage will stay on the same scanning curve until  $S_{ohmax}$  is reached. When the state returns to the drainage curve at D, if drainage continues, the state will follow DB, until imbibition again succeeds.

Another situation may arise when oil saturation decreases at the state of point E. This could happen if oil phase is burnt or dissolved. Then at a point F to the right of E, a subsequent drainage process would result in a scan upward to the drainage curve at point G.

The Carlson (\*CARLSON) method needs to update the historical maximum oil saturation ( $S_{ohmax}$ ) for each grid cell during the simulation. If the oil saturation equals or exceeds the historical maximum,  $S_{ohmax}$ , the drainage curve will be used to determine the value of the oil relative permeability. On the other hand, if the oil saturation in a grid cell falls below  $S_{ohmax}$ , a scanning curve will be employed. In constructing the scanning curve, the approach is based on the assumption that the scanning relative permeability is equal to the drainage relative permeability evaluated at the free oil saturation,  $S_{of}$ , that is:

$$k_{row}^{scan}(S_o) = k_{row}^{drain}(S_{of}) \quad (\text{EQ h.1})$$

where the free oil saturation  $S_{of}$  is obtained from the following equation:

$$S_{of} = S_{orw} + 0.5 \left[ (S_o - S_{ocrt}) + \sqrt{(S_o - S_{ocrt})^2 + \frac{4(S_o - S_{ocrt})}{C}} \right] \quad (\text{EQ h.2})$$

In (EQ h.2),

$S_o$ : Grid cell oil saturation;

$S_{orw}$ : Residual oil saturation for the drainage curve;

$S_{ocrt}$ : Trapped oil saturation calculated from

$$S_{ocrt} = S_{orw} + \frac{S_{ohmax} - S_{orw}}{1 + C (S_{ohmax} - S_{orw})} \quad (\text{EQ h.3})$$

$C$ : Land constant calculated from

$$C = \frac{S_{omax} - sotmax}{(S_{omax} - S_{orw})(sotmax - S_{orw})} \quad (\text{EQ h.4})$$

$S_{ohmax}$ : Historical maximum oil saturation;

$sotmax$ : Inputted maximum trapped oil saturation of the imbibition curve.

The scanning curves constructed by the Carlson method retain a geometrical simplicity since the only hysteretic parameter inputted is  $sotmax$ .

The Killough (\*KILLOUGH) method (Killough, J. E., "Reservoir Simulation with History-Dependent Saturation Functions", SPEJ, Feb. 1976, 37-48) renders more user control on the formation of the scanning curves. Similar to the Carlson's, it uses the same formula, (EQ h.3) and (EQ h.4) to compute the trapped saturation  $S_{ocrt}$ , but the relative permeability on the scanning curve is calculated by either a relative permeability interpolation

$$k_{row}^{scan}(S_o) = k_{row}^{drain}(S_{ohmax}) \times \frac{k_{row}^{imbib}(\bar{S}_o)}{k_{row}^{drain}(S_{omax})} \quad (\text{EQ h.5})$$

or a saturation interpolation

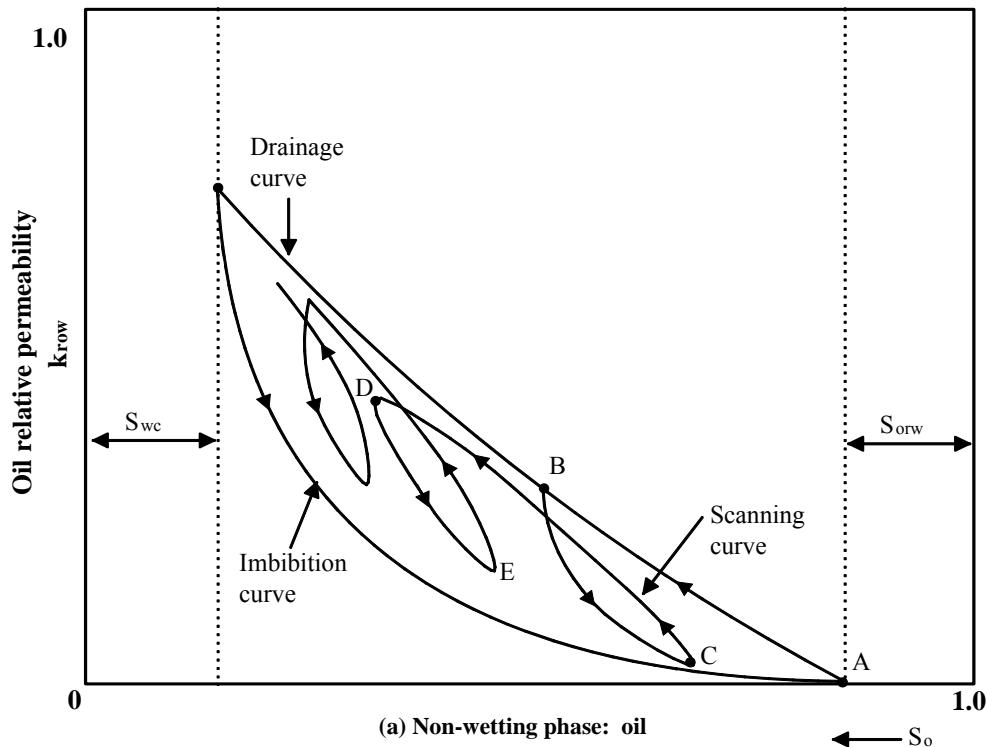
$$k_{row}^{scan}(S_o) = k_{row}^{drain}(S_{ohmax}) \times \left( \frac{S_o - S_{ocrt}}{S_{ohmax} - S_{ocrt}} \right)^{(\text{hyexo})} \quad (\text{EQ h.6})$$

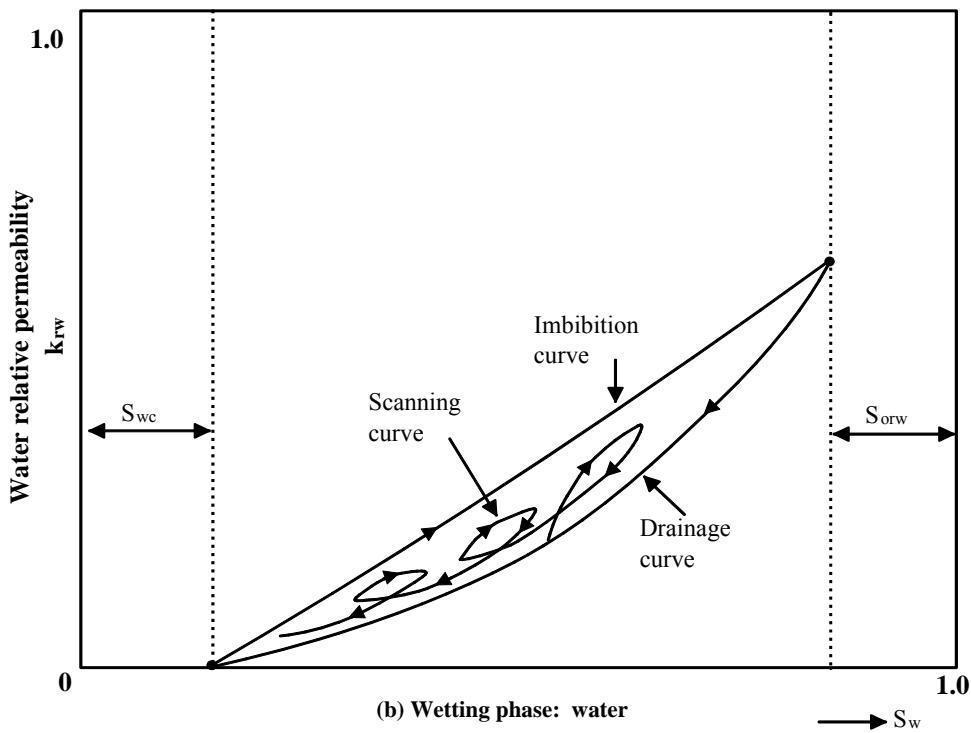
where  $k_{row}^{drain}$  and  $k_{row}^{imbib}$  are the relative permeability values on the drainage and imbibition curve and the normalized oil saturation  $\bar{S}_o$  in (EQ h.5) is computed from

$$\bar{S}_o = \frac{(S_o - S_{ocrt}) \times (S_{omax} - sotmax)}{(S_{ohmax} - S_{ocrt})} + sotmax \quad (\text{EQ h.7})$$

### The BBM method

Based on a similar observation as the previous section, the BBM method constructs the non-wetting phase relative permeability scanning curves differently from the Carlson and the Killough method (For details, see Beattie, C.I., Boberg, T.C. and McNab, G.S., "Reservoir Simulation of Cyclic Steam Stimulation in the Cold Lake Oil Sands", SPE 18752). Figure HY2(a) shows the history of oil relative permeability vs. water saturation by tracking a grid block over several injection-production cycles. As the oil saturation increases (or water saturation decreases) initially from point A to B,  $k_{row}$  follows the drainage curve. At point B the saturation reversal occurs,  $k_{row}$  moves along a scanning curve BC toward the imbibition bounding curve. When second reversal occurs at point C, a new scanning curve CD is trailed until another reversal takes place in the later cycles.





**Figure HY2:** Hysteretic characteristics of wetting and non-wetting phase relative permeability, BBM method.

According to the BBM method, using a normalized water saturation and oil relative permeability

$$\bar{S}_w = \frac{(S_w - S_{wc})}{(1 - S_{wc} - S_{orw})}; \quad \bar{k}_{row} = \frac{k_{row}(S_w)}{k_{row}(S_{wc})}, \quad (\text{EQ h.8})$$

hysteretic relative permeabilities to oil on the scanning curve are computed by the following formulae:

When  $S_w$  increasing:

$$\bar{k}_{row}^{\text{scan}} = \bar{k}_{row}^{\text{imbib}} + (\text{rnw}) \times \left( \frac{1 - \bar{S}_w}{1 - (\bar{S}_w)_p} \right)^{(\text{enwi})} \times (\bar{k}_{row}^{\text{drain}} - \bar{k}_{row}^{\text{imbib}}) \quad (\text{EQ h.9})$$

When  $S_w$  decreasing:

$$\bar{k}_{row}^{\text{scan}} = \bar{k}_{row}^{\text{drain}} - (1 - \text{rnw}) \times \left( \frac{\bar{S}_w}{(\bar{S}_w)_p} \right)^{(\text{enwd})} \times (\bar{k}_{row}^{\text{drain}} - \bar{k}_{row}^{\text{imbib}}) \quad (\text{EQ h.10})$$

In (EQ h.9) and (EQ h.10), the subscript 'p' denotes the values at the point where the last saturation reversal occurred and

$$r_{nw} = \frac{(\bar{k}_{row}^{scan})_p - (\bar{k}_{row}^{imbib})_p}{(\bar{k}_{row}^{drain})_p - (\bar{k}_{row}^{imbib})_p} \quad (\text{EQ h.11})$$

Once the normalized values are known, the relative permeability can easily be recovered from (EQ h.8).

### **Wetting Phase Relative Permeability Hysteresis**

The wetting phase relative permeability generally exhibits a much less dependence on the change of flow directions than the non-wetting phase. However, it has been observed that for some processes, such as cyclic steam, the wetting phase relative permeability hysteresis plays an important role in achieving a satisfactory history match.

### **The Killough method**

In association with its non-wetting phase counterpart, Figure HY1(b) shows a sketch of hysteretic relative permeability profile for the wetting phase. The curve A'B' represents the user-inputted water (wetting phase) relative permeability drainage curve, and B'C' represents the user-inputted water relative permeability imbibition curve. With reference to the analysis of the non-wetting phase hysteresis, as water saturation decreases from A' to B', the drainage curve A'B' is followed; then the imbibition curve B'C' is followed as the water saturation increases from  $S_{wc}$  until reaching the maximum water saturation, 1-sotmax, on the curve. For any reversal occurred at an intermediate saturation on A'B', a scanning curve, shown as D'E', is generated starting at  $1-S_{ohmax}$  and ending at  $1-S_{ocrt}$ . The maximum reachable  $k_{rw}$  on the scanning curve is predicted by an interpolation between the imbibition curve and drainage curve

$$k_{rw}^{scan}(1-S_{ocrt}) = k_{rw}^{drain}(1-S_{ocrt}) + [k_{rw}^{imbib}(1-sotmax) - k_{rw}^{drain}(1-sotmax)] \times \left( \frac{S_{ocrt} - S_{orw}}{sotmax - S_{orw}} \right)^{(hyexw)} \quad (\text{EQ h.12})$$

The user-specified parameter,  $hyexw$ , is used to allow a closer fit with inputted data.

At any given  $S_w$ , the wetting phase relative permeability on the scanning curve is then calculated from

$$k_{rw}^{scan}(S_w) = k_{rw}^{drain}(1-S_{ohmax}) + \left[ \frac{k_{rw}^{imbib}(1-\bar{S}_o)}{k_{rw}^{imbib}(1-sotmax)} \right] \left[ \frac{k_{rw}^{imbib}(1-\bar{S}_o)}{k_{rw}^{imbib}(1-sotmax)} \right] \times [k_{rw}^{scan}(1-S_{ocrt}) - k_{rw}^{drain}(1-S_{ohmax})] \quad (\text{EQ h.13})$$

where  $\bar{S}_o$  is the normalized oil saturation from (EQ h.7).

## The BBM method

Figure HY2(b) depicts the relative permeability hysteretic feature of the wetting phase, the water in this case. Very similar to the treatment for the non-wetting phase, the hysteretic relative permeabilities to water on the scanning curve are computed by the following formulae:

When  $S_w$  increasing:

$$\bar{k}_{rw}^{scan} = \bar{k}_{rw}^{imbib} - (rwt) \times \left( \frac{1 - \bar{S}_w}{1 - (\bar{S}_w)_p} \right)^{(ewti)} \times (\bar{k}_{rw}^{imbib} - \bar{k}_{rw}^{drain}) \quad (EQ\ h.14)$$

When  $S_w$  decreasing:

$$\bar{k}_{rw}^{scan} = \bar{k}_{rw}^{drain} + (1 - rwt) \times \left( \frac{\bar{S}_w}{(\bar{S}_w)_p} \right)^{(ewtd)} \times (\bar{k}_{rw}^{imbib} - \bar{k}_{rw}^{drain}) \quad (EQ\ h.15)$$

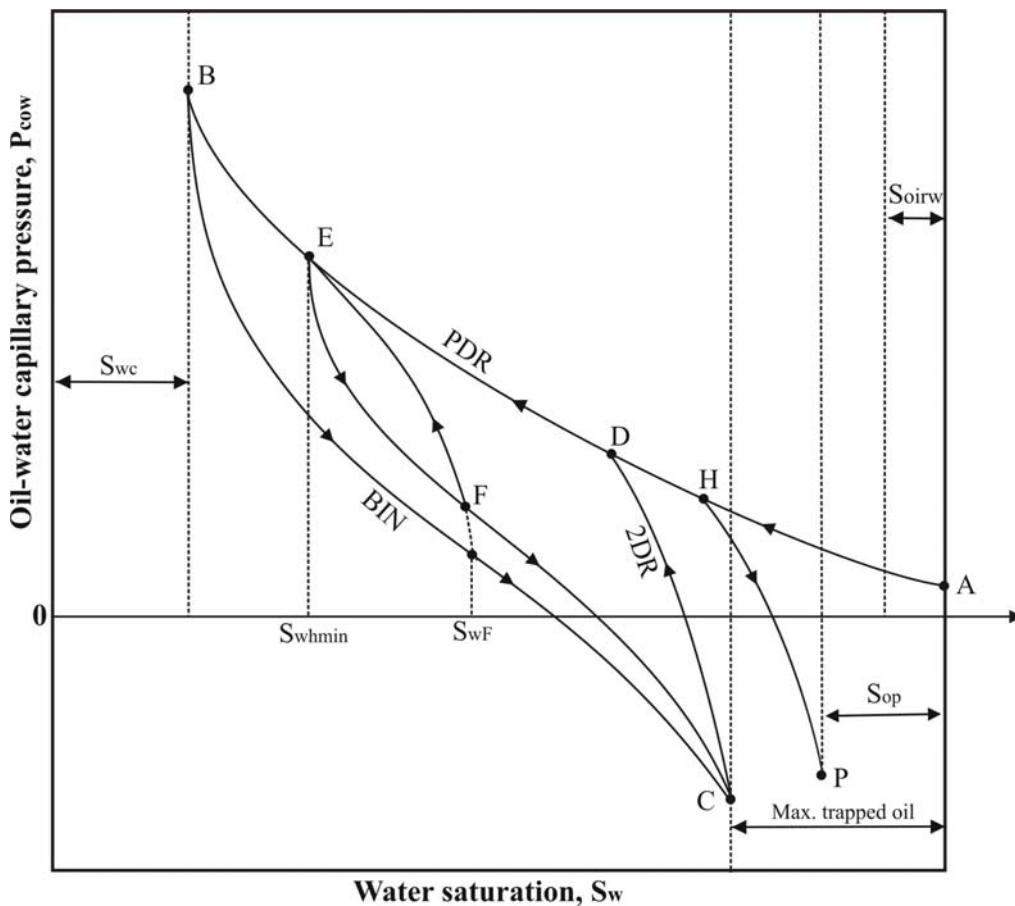
In (EQ h.14) and (EQ h.15), the subscript 'p' again denotes the values at the point where the last saturation reversal occurred and

$$rwt = \frac{(\bar{k}_{rw}^{imbib})_p - (\bar{k}_{rw}^{scan})_p}{(\bar{k}_{rw}^{imbib})_p - (\bar{k}_{rw}^{drain})_p} \quad (EQ\ h.16)$$

## Capillary Pressure Hysteresis

The method used in STARS to model the capillary pressure hysteresis is similar to the one suggested by Killough (Killough, J. E., "Reservoir Simulation with History-Dependent Functions", Soc. Pet. Eng. J., Feb. 1976, Trans. AIME, Vol. 261, pp. 37-48). A brief illustration of the method will be focused on an oil-water system and can be applied to a liquid-gas system straightforwardly.

Consider an oil-water system with the capillary pressure,  $P_{cow}$ , versus water saturation,  $S_w$ , behavior given in Figure HY3. The three curves A-B, B-C and C-D-B are referred to as the primary drainage curve (PDR), the bounding imbibition curve (BIM) and the secondary drainage curve (2DR). These curves are obtained when displacement in either direction is carried out completely to the residual value of saturation.



**Figure HY3: Oil-water capillary pressure bounding and scanning curves**

Suppose the process has already completed the primary drainage and the water saturation starts increasing at intermediate water saturation  $S_{whmin}$  (point E) on 2DR curve,  $P_{cow}$  will follow an imbibition scanning curve EC. For a given water saturation,  $S_w$ , the capillary pressure on the scanning imbibition curve EC is calculated by a weighted average of the values from secondary drainage 2DR and bounding imbibition BIN curve

$$P_{cow}^{scan}(S_w) = P_{cow}^{2DR}(S_w) - F(S_w) \times [P_{cow}^{2DR}(S_w) - P_{cow}^{BIN}(S_w)] \quad (\text{EQ h.17})$$

with the weighting function  $F(S_w)$  defined as

$$F(S_w) = \frac{\frac{1}{S_w - S_{wh\min} + \varepsilon} - \frac{1}{\varepsilon}}{\frac{1}{S_{w\max} - S_{wh\min} + \varepsilon} - \frac{1}{\varepsilon}} \quad (\text{EQ h.18})$$

where

- $S_{wh\min}$ : Historical minimum water saturation in the drainage;
- $S_{w\max}$ : Maximum water saturation attainable:  $S_{w\max} = 1.0 - (\text{Maximum trapped oil saturation})$ ;
- $\varepsilon$ : Entered curvature parameter ( $= \text{epsow}$ ).

If the imbibition scanning curve started at point E experiences a second saturation reversal at point F ( $S_w$  decreasing), a new drainage scanning curve FE will be followed which generally is not a re-traverse of EF:

$$P_{cow}^{\text{scen}}(S_w) = P_{cow}^{\text{BIM}}(S_w) + G(S_w) \times \left[ P_{cow}^{2\text{DR}}(S_w) - P_{cow}^{\text{BIM}}(S_w) \right] \quad (\text{EQ h.19})$$

with the weighting function  $G(S_w)$  defined as

$$G(S_w) = \frac{\frac{1}{S_{wf} - S_w + \varepsilon} - \frac{1}{\varepsilon}}{\frac{1}{S_{wf} - S_{wh\min} + \varepsilon} - \frac{1}{\varepsilon}} \quad (\text{EQ h.20})$$

The unknown saturation  $S_{wf}$  (see Figure HY3) is determined from the condition that the two scanning curves meet at point F. In STARS, the keyword \*HYS\_LEVEL controls the level of scanning curve to be employed. The default is 1, that is, the drainage scanning curve will re-traverse the imbibition scanning curves and vice versa.

The explanations presented above for saturation reversals in the drainage process can be applied likewise to the case which an imbibition process is reversed. Similar equations for calculating the scanning capillary pressure can be derived (see Aziz, K. and Settari, A., *Petroleum Reservoir Simulation*, Applied Science Publishers, London, 1979, pp. 395-399).

As mentioned, the scanning curves constructed by the above equations is valid only when the system has previously undergone a complete drainage displacement. If a reversal occurs at point H on the primary drainage curve PDR, rather than following a scanning curve back to the maximum trapped oil saturation, some new residual  $S_{op}$  has to be determined and then the scanning curve HP is constructed. In STARS, user can specify if the primary drainage has completed by the keyword \*HYS\_DRAINW or HYS\_DRAINW\_2ND for all simulation grids. STARS will also check the drainage and imbibition capillary pressure input to determine the state of the grids belonging to that saturation table set. During the simulation run, STARS detects internally whether a grid cell has finished the primary drainage and then employ different formula to calculate scanning capillary pressure.

## Relative Permeability and Capillary Pressure Hysteresis for Oil Wet Reservoir

As a graphical representation of the input data and the hysteretic characteristics, Figure HY4 depicts oil-water capillary pressure and water relative permeability for an oil wet rock. The top graph shows the capillary pressure defined as  $P_{cow} = P_o - P_w$  versus water saturation (**non wetting phase**) behavior in an oil-water system. In comparison with Figure HY3 for a water wet system, the three curves A-B, B-C and C-D-B are the primary drainage curve (PDR), the bounding imbibition curve (BIM) and the secondary drainage curve (2DR), respectively. The PDR (or 2DR) and BIM curves resemble the fourth and fifth column of the \*SWT table.

Similar to a water wet system, these curves are obtained when displacement in either direction is carried out completely to the residual value of saturation. The graph on the bottom is the corresponding  $k_{rwo}$  versus water saturation curves of the same system assuming water relative permeability hysteresis is also considered in the simulation. The two graphs have been drawn adjacently aimed at a clear visualization of the coordination for all inputted curves. Here are the major rules to follow when entering these curves for oil wet rocks.

1. The primary drainage capillary pressure (A-B) should be all negative and the imbibition capillary pressure should not be smaller in value than the drainage curve at the same table entry;
2. The drainage curve and the imbibition curve for capillary pressure and relative permeability should meet at Soc;
3. The maximum trapped water saturation  $S_{wtrap}$  from the imbibition relative permeability curve and the imbibition capillary pressure curve should be identical if hysteresis option is applied for the both.

Hysteretic values of capillary pressure and relative permeability will be determined in the same fashion as the water wet rock.

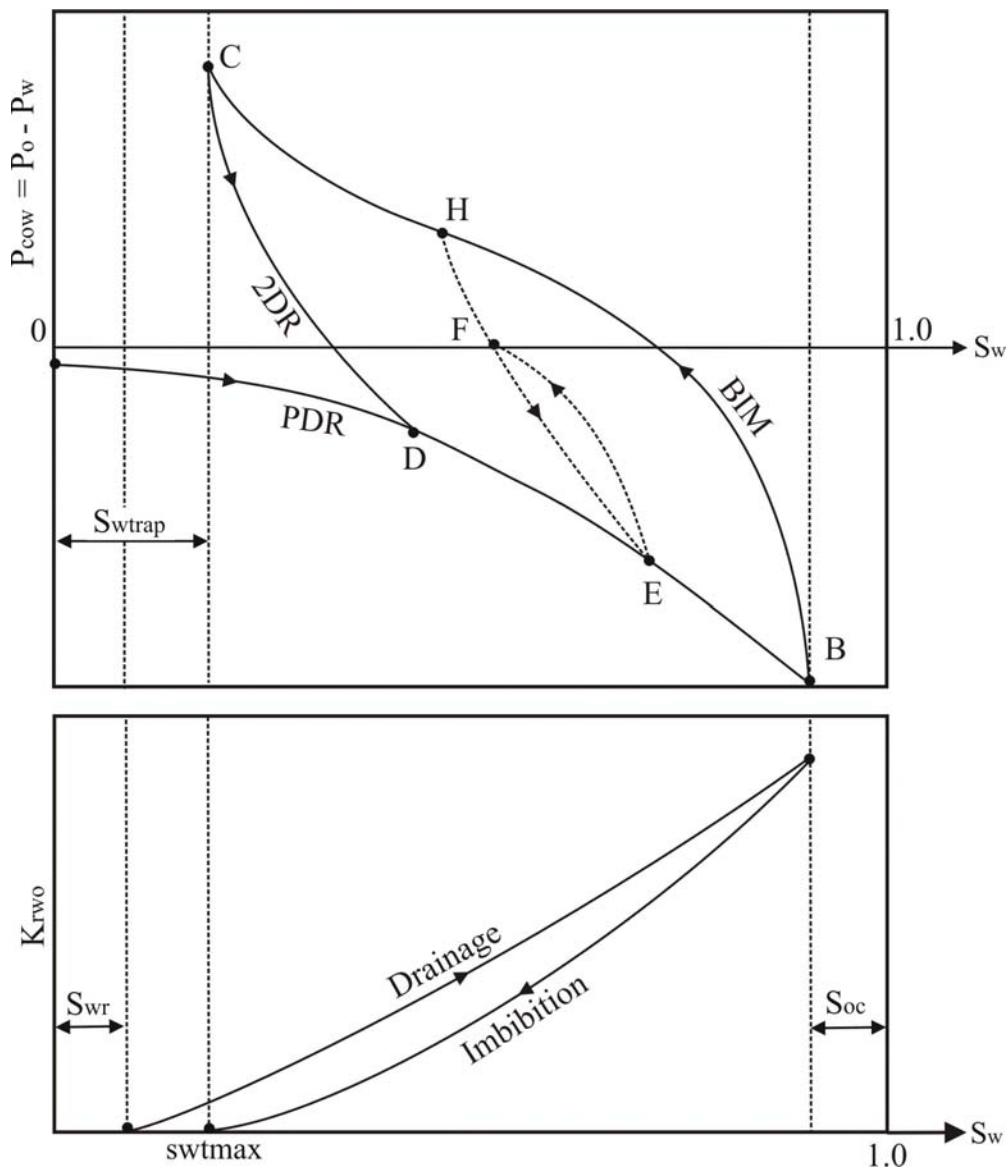


Figure HY4: Oil-water capillary pressure and water relative permeability curves for oil-wet rocks.

### EXAMPLE:

For a water wet system, the following keywords will activate the Carlson method for the hysteretic effect on the non-wetting oil relative permeability, the gas relative permeability and the water-oil capillary pressure hysteresis:

*HYS_KRO	*CARLSON	*SOTMAX 0.45	**-- Maximum trapped oil saturation
*HYS_KRG	*CARLSON	*SGTMAX 0.40	**-- Maximum trapped gas saturation
*HYS_Pcow	*EPSOW	0.10	**-- $P_{cowi}$ entered via *SWT table

For a water wet system, the following keywords will activate the Killough method for the relative permeability hysteresis to non-wetting oil phase, wetting water phase and gas phase:

```
*HYS_KRO *KILLOUGH *HYEXO 2.0 *HYEXW 2.5
*SWTI
** Swi      krowi    krwi
  0.13     1.0      0.0
.
.
.
  0.4595   0.0      0.024

*HYS_KRG  *KILLOUGH *HYEXG 1.5
*SLTI
** Sli      krgi
  0.200    0.17
.
.
.
  0.52      0.0
```

For an oil wet system, the following keywords will activate the BBM method for the relative permeability hysteresis to non-wetting water phase, wetting oil phase:

```
*HYS_KRW  *BBM
  *EWTD 13.0 *EWTD 13.0 *ENWI 10.0 *ENWD 3.0  *RNW 1.0 *RWT 0.5
*SWTI
** swti    krwi    krowi
  0.2      0.0      0.9
  0.32     0.01     int
  0.56     0.16     int
  0.68     0.35     int
  0.8      0.9      0.0
```

### Output of Hysteretic Relative Permeability

To help user to verify the input data and observe the relative permeability behavior, two special history outputs can be applied. The following keywords in Input/Output Control section will record the history of both wetting (water or oil) and non-wetting phase (oil or water) relative permeability vs. water saturation for grid block (i1, j1, k1):

```
outsrf special blockvar rpwt i1,j1,k1
               blockvar rpnw i1,j1,k1
               blockvar sw   i1,j1,k1
```

rpwt and rpnw are for the wetting phase and non-wetting phase, respectively. Plotting the curves with Results Graph, user should be able to check if the scanning curves behave appropriately, such as whether the scanning values are confined in the bounding curves, and fine-tune the input parameters accordingly.

---

## **Relative Permeability Endpoints**

\*SWR, \*SWCON, \*SWCRIT, \*SORW,  
\*SOIRW, \*SGR, \*SGCON, \*SORG, \*SOIRG, \*SWRG, \*SWIRG, \*KRWRO, \*KRWIRO,  
\*KROCW, \*KRGCW, \*PCWEND, \*PCGEND, \*PCGWEND

### **PURPOSE:**

Overwrite critical and connate saturations and endpoints from the tables.

### **FORMAT:**

*SWR or *SWCON	S <sub>wcon</sub>
*SWCRIT	S <sub>wcrit</sub>
*SORW	S <sub>orw</sub>
*SOIRW	S <sub>oirw</sub>
*SGR	S <sub>gr</sub>
*SGCON	S <sub>gcon</sub>
*SORG	S <sub>org</sub>
*SOIRG	S <sub>oирг</sub>
*SWRG	S <sub>wrg</sub>
*SWIRG	S <sub>wирг</sub>
*KRWRO	k <sub>rwro</sub>
*KRWIRO	k <sub>rwiro</sub>
*KROCW	k <sub>rocw</sub>
*KRGCW	k <sub>rgew</sub>
*PCWEND	pcwend
*PCGEND	pcgend
*PCGWEND	pcgwend

### **DEFINITIONS:**

S<sub>wcon</sub>

Irreducible water saturation. Sometimes known as S<sub>wc</sub>, but the keyword used is \*SWR or \*SWCON. The allowed range is 0 to 1.

S<sub>wcrit</sub>

Critical water saturation. The allowed range is 0 to 1.

S<sub>orw</sub>

Residual oil saturation for water injection. The allowed range is 0 to 1.

S<sub>oirw</sub>

Irreducible oil saturation for water injection. The allowed range is 0 to 1

S<sub>gr</sub>

Critical gas saturation. The allowed range is 0 to 1

S<sub>gcon</sub>

Connate gas saturation. The allowed range is 0 to 1.

$S_{org}$	Residual oil saturation for gas injection. The allowed range is 0 to 1. Corresponds to $S_{lrg} - S_{wc}$ if liquid contains $S_{wc}$ .
$S_{oirg}$	Irreducible oil saturation for gas injection. The allowed range is 0 to 1. Corresponds to $S_{lc} - S_{wc}$ if liquid contains $S_{wc}$ .
$S_{wrg}$	Residual water saturation for gas injection. The allowed range is 0 to 1. Corresponds to $S_{lrg} - S_{oc}$ if liquid contains $S_{oc}$ . This quantity is used only when the system is oil-wet or intermediate-wet.
$S_{wirg}$	Irreducible water saturation for gas injection. The allowed range is 0 to 1. Corresponds to $S_{lc} - S_{oc}$ if liquid contains $S_{oc}$ . This quantity is used only when the system is oil-wet or intermediate-wet.
$k_{rwro}$	Relative permeability to water at $S_w = 1 - S_{orw}$ in the water/oil table. The allowed range is 1.0e-9 to 1. Keyword *KRWRO is considered obsolete and *KRWIRO should be used instead. Any *KRWRO data is converted internally to *KRWIRO and is reported as $k_{rwiro}$ . Use of *KRWRO makes it possible to scale the $k_{rw}$ curve such that $k_{rwiro}$ exceeds 1 which is not a recommended practice. Note that $k_{rwro}$ may be different from $k_{rwiro}$ when $S_{orw} > S_{irw}$ .
$k_{rwiro}$	Relative permeability to water at $S_w = 1 - S_{oirw}$ in the water/oil table. The allowed range is 1.0e-9 to 1.
$k_{rocw}$	Relative permeability to oil at connate water and zero gas saturation. The allowed range is 1.0e-9 to 1.
$k_{rgcw}$	Relative permeability to gas at connate liquid. The allowed range is 1.0e-9 to 1.
$pcwend$	Maximum value of water-oil capillary pressure, usually an endpoint value (kPa   psi   kPa). This quantity may be specified only when water-oil capillary pressure was entered via table *SWT. The sign of $pcwend$ should match that of the *SWT table endpoint, e.g., negative when *OILWET is used.

**pcgend**

Maximum value of gas-liquid capillary pressure, usually an endpoint value (kPa | psi | kPa). This quantity may be specified only when gas-liquid capillary pressure was entered via table \*SLT.

**pcgwend**

Maximum value of gas-water capillary pressure, usually an endpoint value (kPa | psi | kPa). This quantity may be specified only when gas-water capillary pressure was entered via table \*SWT subkeyword \*PCGW.

#### **DEFAULTS:**

These keywords should be used only to overwrite the critical and connate saturations and endpoints that were found in the tables entered via \*SWT and \*SLT.

#### **CONDITIONS:**

To be effective, these keywords must occur after \*SWT and \*SLT. Any of these keywords found before the corresponding table will be ignored.

For the size of the mobile regions  $1-S_{wcrit}-S_{orw}$  and  $1-S_{gcrit}-S_{lrg}$ , the minimum allowed value is 0.02 and the minimum recommended value is 0.3.

#### **EXPLANATION:**

The result of applying these endpoint modifications to the tables entered via \*SWT and \*SLT will be reported in the output file in the data summary section. Information about endpoints scaling is at the beginning of this section "CRITICAL AND CONNATE SATURATIONS, SCALE-UP FACTORS, AND NORMALIZATION".

## Relative Permeability Temperature Dependence

\*KRTEMTAB

### PURPOSE:

Specify temperature dependence for critical saturations and endpoints.

### FORMAT:

```
*KRTEMTAB key(1) ... key(n)
      { T val(1) ... val(n) }
```

### DEFINITIONS:

#### \*KRTEMTAB

A list of relative permeability endpoint keywords must follow, along with a table of corresponding endpoint values versus temperature.

#### key(i)

Any keywords from the RELATIVE PERMEABILITY ENDPOINT keyword group (\*SWR, \*SWCRIT, \*SORW, \*SOIRW, \*SGR, \*SGCON, \*SORG, \*SOIRG, \*SWRG, \*SWIRG, \*KRWIRO, \*KRWRO, \*KROCW, \*KRGCW, \*PCWEND, \*PCGEND) in any order. A maximum of 10 keywords are allowed.

#### T

Temperature table entry (C | F). There must be at least 2 entries, and all temperature entries must be evenly spaced. The maximum allowed number of temperature entries is 10. The first T entry is the reference temperature for data entered via keywords \*SWT, \*SLT, \*SWR, etc. and \*BSWR, etc.

#### val(i)

Table values corresponding to keyword key(i).

### DEFAULTS:

If \*KRTEMTAB is absent, no temperature dependence is assumed.

If \*KRTEMTAB is present, then for each endpoint keyword absent from the table the corresponding quantity will be independent of temperature.

### CONDITIONS:

This keyword must occur after \*SWT and \*SLT.

For the size of the mobile regions  $1 - S_{w\text{crit}} - S_{o\text{rw}}$  and  $1 - S_{g\text{crit}} - S_{l\text{rg}}$ , the minimum allowed value is 0.02 and the minimum recommended value is 0.3.

### EXPLANATION:

See the section in this chapter's introduction entitled "CRITICAL AND CONNATE SATURATIONS, SCALE-UP FACTORS, AND NORMALIZATION". See also Appendix D.6.

---

**Rock-Fluid Scaling for Each Block** \*BSWR, \*BSWCON, \*BSWCRT,  
\*BSORW, \*BSOIRW, \*BSGR, \*BSGCON, \*BSORG, \*BSOIRG, \*BSWRG, \*BSWIRG,  
\*BKRWRO, \*BKRWIRO, \*BKROCW, \*BKRGCW, \*BPCWMAX, \*BPCGMAX

**PURPOSE:**

Specify end points of rock-fluid tables for each grid block.

**ARRAY:**

\*BSWR or \*BSWCON  
\*BSWCRT  
\*BSORW  
\*BSOIRW  
\*BSGR  
\*BSGCON  
\*BSORG  
\*BSOIRG  
\*BSWRG  
\*BSWIRG  
\*BKRWRO  
\*BKRWIRO  
\*BKROCW  
\*BKRGCW  
\*BPCWMAX  
\*BPCGMAX

**DEFINITIONS:**

\*BSWR or \*BSWCON

Connate water saturation. The allowed range is 0 to 1.

\*BSWCRT

Critical water saturation. The allowed range is 0 to 1.

\*BSORW

Residual oil saturation to water. The allowed range is 0 to 1.

\*BSOIRW

Irreducible oil saturation to water. The allowed range is 0 to 1.

\*BSGR

Critical gas saturation. The allowed range is 0 to 1.

\*BSGCON

Connate gas saturation. The allowed range is 0 to 1.

\*BSORG

Residual oil saturation to gas. The allowed range is 0 to 1.

<b>*BSOIRG</b>	Irreducible oil saturation to gas. The allowed range is 0 to 1.
<b>*BSWRG</b>	Residual water saturation to gas. The allowed range is 0 to 1. This quantity is used only for blocks whose system is oil-wet or intermediate-wet.
<b>*BSWIRG</b>	Irreducible water saturation to gas. The allowed range is 0 to 1. This quantity is used only for blocks whose system is oil-wet or intermediate-wet.
<b>*BKRWRO</b>	Relative permeability to water at residual oil and zero gas saturation. The allowed range is 1.0e-9 to 1. Keyword *BKRWRO is obsolete and *BKRWIRO should be used instead. Any *BKRWRO data is converted internally to *BKRWIRO and is reported as $k_{rwiro}$ . Use of *BKRWRO makes it possible to scale the $k_{rw}$ curve such that $k_{rwiro}$ exceeds 1 which is not a recommended practice. Note that $k_{rwro}$ may be different from $k_{rwiro}$ when $S_{orw} > S_{oirw}$ .
<b>*BKRWIRO</b>	Relative permeability to water at irreducible oil and zero gas saturation. The allowed range is 1.0e-9 to 1.
<b>*BKROCW</b>	Relative permeability to oil at connate water and zero gas saturation. The allowed range is 1.0e-9 to 1.
<b>*BKRGCW</b>	Relative permeability to gas at connate liquid. The allowed range is 1.0e-9 to 1.
<b>*BPCWMAX</b>	Maximum value of water-oil capillary pressure, usually an endpoint value (kPa   psi   kPa).
<b>*BPCGMAX</b>	Maximum value of gas-oil capillary pressure, usually an endpoint value (kPa   psi   kPa).

#### **DEFAULTS:**

Each quantity may be defaulted independently. For example, the defaulting and assignment of  $S_{wr}$  via \*SWT, \*SWR or \*BSWR does not affect the defaulting and assignment of  $S_{gr}$  via \*SLT, \*SGR or \*BSGR.

For each keyword absent, the corresponding quantity used by a block comes from the rock type assigned to that block, and the quantity will change according to rock type re-assignments via \*KRTYPE in recurrent data.

Upon the first appearance of each of these keywords, each block is seeded with the associated rock type value for that quantity. Subsequently, each block's default is the value assigned up to that point in the simulation.

### **CONDITIONS:**

These keywords, if present, must appear after all other keywords that define or modify endpoints of the rock-fluid tables for all rock types (\*SWT, \*SLT, \*KRTEMTAB, all modifiers such as \*SWR).

For the size of the mobile regions  $1-S_{wcrit}-S_{orw}$  and  $1-S_{gcrit}-S_{lrg}$ , the minimum allowed value is 0.02 and the minimum recommended value is 0.3.

These keywords may appear also in the Well and Recurrent Data section.

### **EXPLANATION:**

All these keywords are grid arrays, and all array reading option subkeywords are valid.

See the section in this chapter's introduction entitled "CRITICAL AND CONNATE SATURATIONS, SCALE-UP FACTORS, AND NORMALIZATION".

### **Rock Type Versus Individual Block Data**

There are two ways to specify an endpoint scaling quantity for a given grid block: (1) through the rock type associated with the block, or (2) directly via one of the above "block" keywords. If this "block" keyword is absent then the quantity is obtained from the rock type during the entire simulation.

However, once the "block" keyword for a quantity is detected then that quantity is obtained FOR ALL BLOCKS from the "block" array at all times after that. This switch over from rock type to block is facilitated by the fact that each entry in the "block" array is seeded from the associated rock type in effect when the keyword first appears. However, it also means that thereafter in the run the quantity in question is never obtained from a block's rock type, even if the rock type changes via \*KRTYPE. It is not possible to switch from "block" array back to rock type. Any subsequent changes to the quantity must be done via the "block" keyword.

### **Inheritance by Refined Grids**

Inheritance of values defined by these keywords from parent to child blocks is performed as in recurrent data, that is, immediately upon reading the keyword data. See **Find-Grid Inheritance** in the Well and Recurrent Data chapter.

---

## Capillary Pressure Third-Phase Correction (Optional)

\*PC\_3RD\_PHASE, \*EPCOW, \*EPCGO, \*EXPCW, \*EXPCG

### PURPOSE:

Specify third-phase correction for capillary pressure.

### FORMAT:

\*PC\_3RD\_PHASE ( \*EPCOW  $\varepsilon_w$  | \*EXPCW  $n_w$  | \*EPCGO  $\varepsilon_g$  | \*EXPCG  $n_g$  )

### DEFINITIONS:

\*EPCOW  $\varepsilon_w$

Specify curvature parameter  $\varepsilon_w$  for correcting gas/oil capillary pressure with respect to water saturation (see Eq. CP4 and CP5).

\*EXPCW  $n_w$

Specify curve fitting exponent  $n_w$  for correcting gas/oil capillary pressure with respect to water saturation (see Eq. CP4 and CP5).

\*EPCGO  $\varepsilon_g$

Specify curvature parameter  $\varepsilon_g$  for correcting oil/water capillary pressure with respect to gas saturation (see Eq. CP7).

\*EXPCG  $n_g$

Specify curve fitting exponent  $n_g$  for correcting oil/water capillary pressure with respect to gas saturation (see Eq. CP7).

### DEFAULTS:

If \*PC\_3RD\_PHASE is absent, no third-phase correction is made for any capillary pressure.

If \*PC\_3RD\_PHASE is present but a secondary keyword is absent, the third-phase correction is done with the following assumption for the missing subkeyword(s).

Keyword absent	Assumption
*EPCOW	$\varepsilon_w = 10$
*EXPCW	$n_w = 1$
*EPCGO	$\varepsilon_g = 10$
*EXPCG	$n_g = 1$

### CONDITIONS:

Sometimes the capillary pressure between two fluid phases is influenced significantly by the presence of a third fluid phase (gas for oil/water capillary pressure and water for gas/oil capillary pressure). You can account for this by using keyword \*PC\_3RD\_PHASE to enable the third-phase correction for capillary pressure. This option also helps smooth the capillary pressure response when the middle fluid phase, e.g., oil phase, disappears and reappears. Consider applying this option in the following situations:

1. Determine how much influence the third phase saturation will have on capillary pressures corresponding to the conventional two-phase assumption;
2. The physical process will cause middle phase to come and go. For example, oil phase appears in an initially water/gas system. Also, oil phase can disappear and reappear due to vaporization/condensation or chemical reactions.

Keyword \*PC\_3RD\_PHASE may be used together with rock-fluid set interpolation (keyword (\*KRINTRP), as long as each rock-fluid set satisfies the conditions of \*PC\_3RD\_PHASE.

\*PC\_3RD\_PHASE is allowed only for water-wet system (\*WATWET).

### **EXPLANATION:**

#### **Conventional Two-phase Capillary Pressure Scheme**

At any given saturation state, capillary pressures are determined from the following approach

$$p_{cow} = f(s_w) \quad \text{and} \quad p_{cgo} = f(s_l) = f(1 - s_g) \quad (\text{Eq. CP1})$$

This approach assumes that  $P_{cow}$ , the oil/water capillary pressure, depends only on the water saturation (wetting phase) and  $P_{cgo}$ , the gas/oil capillary pressure, depends only on the gas phase saturation (non-wetting phase). It has been observed experimentally that capillary pressures in a three-phase system are different from their two-phase counterparts. In other words, the makeup of the non-wetting phase will influence  $P_{cow}$  and the makeup of the wetting phase will influence  $P_{cgo}$ .

#### **Three-phase Capillary Pressures Through Third-Phase Correction**

In contrast to the two-phase approach (Eq. CP1), three-phase capillary pressures account for the effect of the third phase

$$p_{cow3} = f(s_w, s_g) \quad \text{and} \quad p_{cgo3} = f(s_g, s_w) \quad (\text{Eq. CP2})$$

Here  $P_{cow3}$  and  $P_{cgo3}$  denote three-phase oil/water and gas/oil capillary pressure, respectively, and are obtained by modifying their two-phase counterparts  $P_{cow}$  and  $P_{cgo}$  with respect to the two-phase gas/water capillary  $P_{cgw}$ . The following two sections detail the method of  $s_g$  correction on  $P_{cow}$  and  $s_w$  correction on  $P_{cgo}$ .

#### **$P_{cgo3}$ : water saturation correction for two-phase $P_{cgo}$**

First, specify the gas-water two-phase capillary pressure  $P_{cgw}$  as a function of water saturation in the \*SWT table via subkeyword \*PCGW.  $P_{cgw}$  is depicted graphically in Figure CP1 along with the two-phase  $P_{cgo}$  vs.  $s_l = 1 - s_g$ .

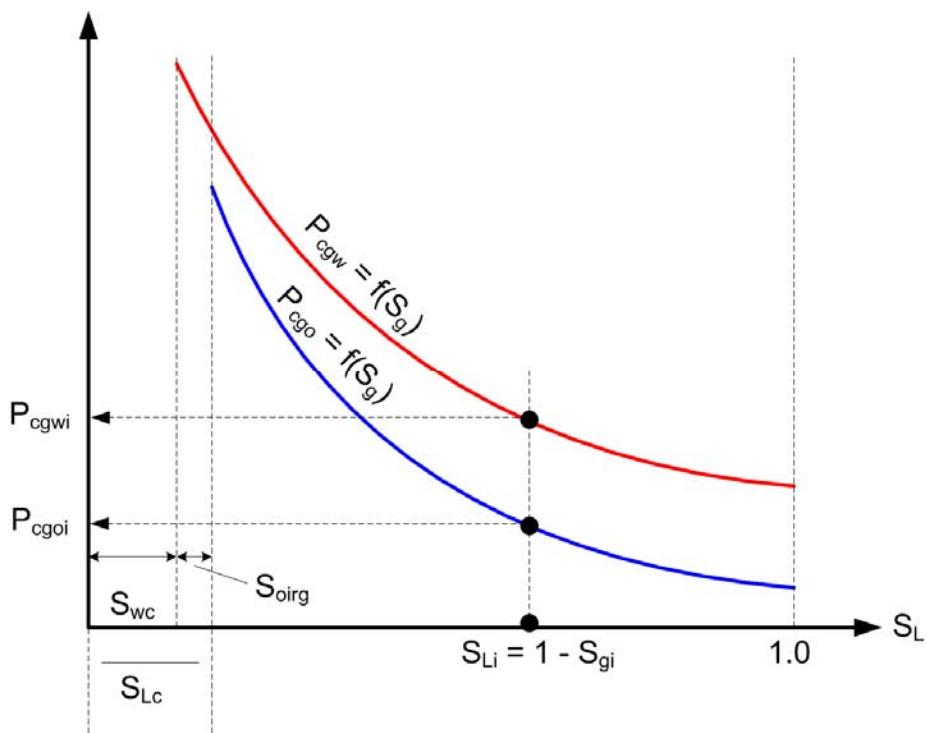


Figure CP1:  $P_{cgo}$  and  $P_{cgw}$  two-phase capillary pressures and third phase  $s_w$  correction

The following discussion outlines the motivation and justification for the  $s_w$  correction for two-phase  $P_{cgo}$ . Assume that connate water  $s_{wc}$  is contained in the core when capillary pressure  $P_{cgo}$  is measured. At liquid saturation  $s_{li} = 1 - s_{gi}$  (Figure CP1) where there are three mobile phases (i.e.,  $s_{wi} > s_{wc}$ ,  $s_{oi} > 0$  and  $s_{gi} > 0$ ), the two-phase capillary pressures are  $P_{cgoi}$  and  $P_{cgwi}$ , respectively. When  $s_{wi} = s_{wc}$  and so only two phases are mobile,  $P_{cgo}$  is sufficient and the correction is unnecessary.

Now imagine a gas-oil two-phase system in which the oil phase ( $s_{oi}=1-s_{wc}-s_{gi}$ ) is replaced gradually by water phase. Consider the following two limiting cases. At  $s_{wi} = s_{wc}$  the three-phase gas-oil capillary pressure  $P_{cgo3}$  is equal to its two-phase counterpart  $P_{cgo}$ , i.e.,  $P_{cgo3} = P_{cgoi}$ . At the other limit  $s_{wi} = 1 - s_{gi}$  (oil phase is replaced completely by water phase) we have  $P_{cgo3} = P_{cgwi}$ . At any state between these limits (arbitrary  $s_{wi}$  at  $s_{gi}$ ) the three-phase gas-oil capillary pressure is calculated by interpolation formula

$$P_{cgo3}(s_{gi}, s_{wi}) = P_{cgoi} + G(s_{wi}) \times [P_{cgwi} - P_{cgoi}] \quad (\text{Eq. CP3})$$

where weighting function  $G(s_{wi})$  is, for \*NOSWC in effect,

$$G(s_{wi}) = \left( \frac{\frac{1}{s_{wi} + \varepsilon_w} - \frac{1}{\varepsilon_w}}{\frac{1}{1 - s_{gi} + \varepsilon_w} - \frac{1}{\varepsilon_w}} \right)^{n_w} \quad (\text{Eq. CP4})$$

or, for \*NOSWC not in effect,

$$G(s_{wi}) = \left( \frac{\frac{1}{s_{wi} - s_{wc} + \varepsilon_w} - \frac{1}{\varepsilon_w}}{\frac{1}{1 - s_{gi} - s_{wc} + \varepsilon_w} - \frac{1}{\varepsilon_w}} \right)^{n_w} \quad (\text{Eq. CP5})$$

In these formulas

$s_{wi}$ :	Grid water saturation at the time;
$s_{gi}$ :	Grid gas saturation at the time;
$P_{cg0i}$ :	The gas-oil two-phase capillary pressure at $s_{gi}$ ;
$P_{cgwi}$ :	The gas-water two-phase capillary pressure at same $s_{gi}$ ;
$\varepsilon_w$ :	User defined curvature fitting parameter;
$n_w$ :	User defined curve fitting exponential.

Weighting function  $G(s_{wi})$  satisfies  $0 \leq G(s_{wi}) \leq 1$  and

$$\begin{aligned} G(s_{wi}) &= 0 \rightarrow P_{cg03} = P_{cg0} \quad \text{at } s_{wi} = 0 \text{ or } s_{wi} = s_{wc}; \\ G(s_{wi}) &= 1 \rightarrow P_{cg03} = P_{cgw} \quad \text{at } s_{wi} = 1 - s_{gi}. \end{aligned}$$

For any  $s_{wi}$  which satisfies  $s_{wc} \leq s_{wi} \leq 1 - s_{gi}$ , the three-phase gas-oil capillary pressure falls between its two two-phase counterparts:  $P_{cg0}(s_{gi}) \leq P_{cg03}(s_{gi}, s_{wi}) \leq P_{cgw}(s_{gi})$  and satisfies the general expression  $p_{cgw} = p_{cow} + p_{cg0}$ .

The above interpolation scheme achieves two objectives: (1) it accounts for the water saturation effect on the gas-oil capillary pressure, and (2) it guarantees a smooth transition of capillary pressure from a gas-oil two-phase state to a gas-water two-phase state. User-defined curve fitting parameters  $\varepsilon_w$  and  $n_w$  add extra flexibility for run control and history matching.

An example of a three-phase  $P_{cg03}$  surface is sketched in Figure CP2. The  $P_{cg0}$  curve is assumed to have \*NOSWC in effect. The surface is concave upward and its curvature is influenced by the value of  $\varepsilon_w$ .

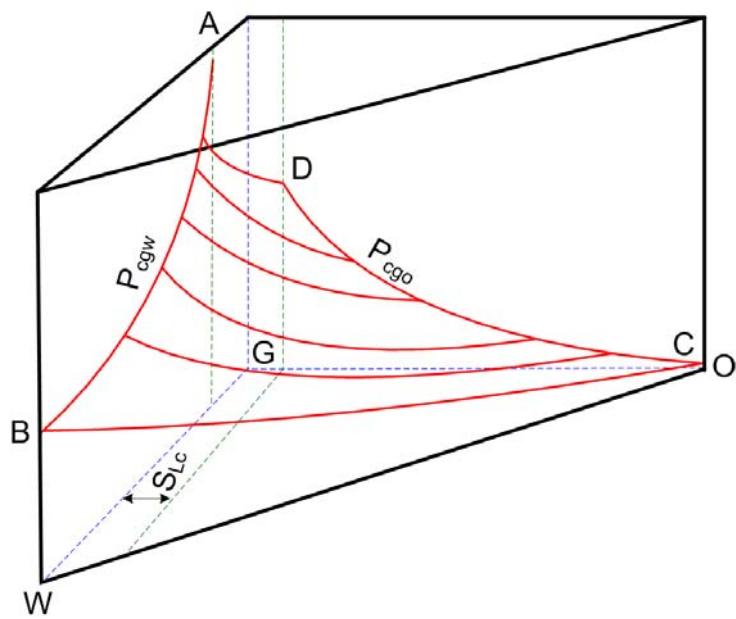
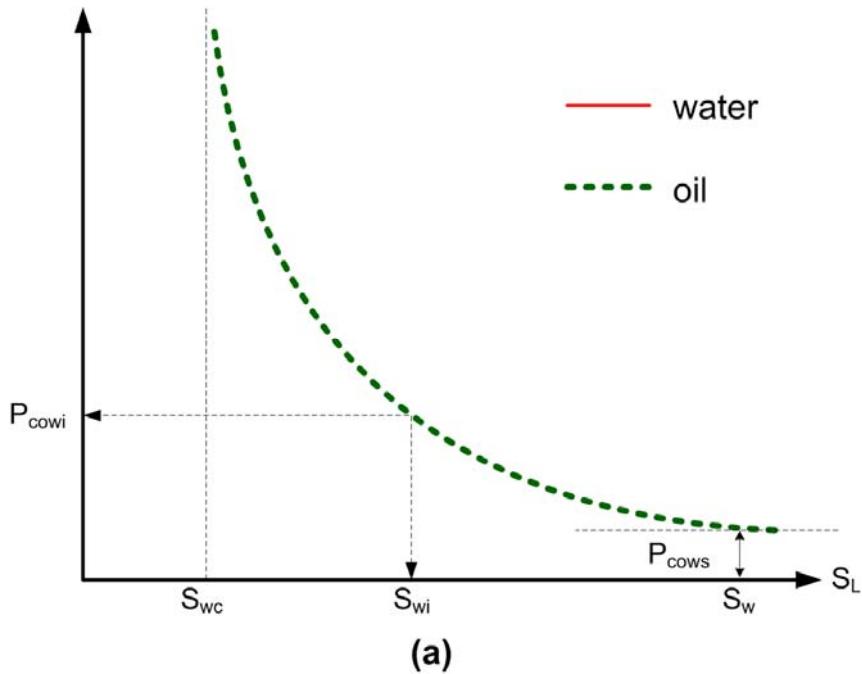


Figure CP2:  $P_{cg3}$  surface

$P_{cow3}$ : gas saturation correction for two-phase  $P_{cow}$



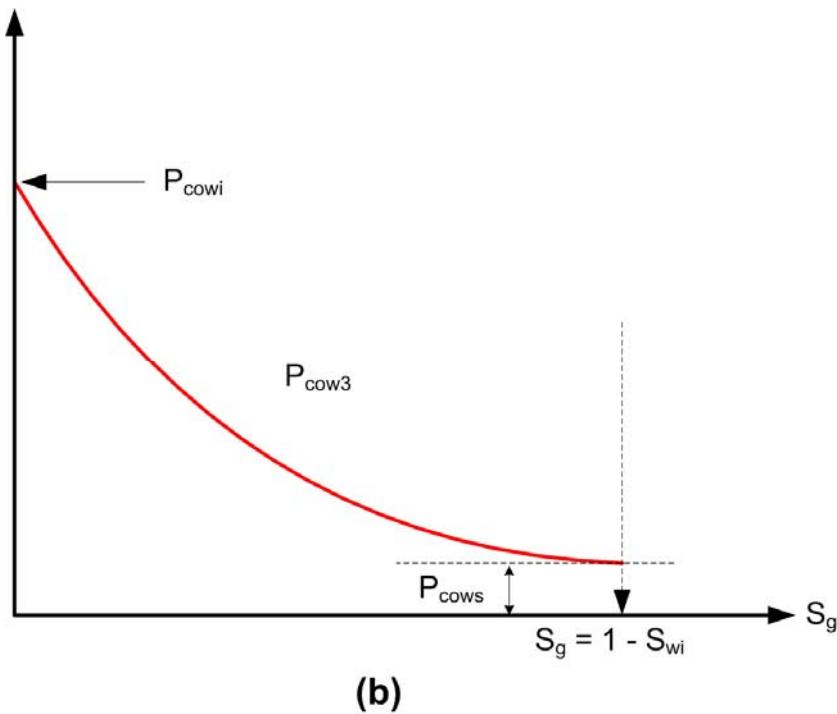


Figure CP3: Oil-water two-phase capillary pressure and the third-phase  $s_g$  correction

The gas saturation correction for two-phase  $P_{cow}$  is structured differently from  $s_w$  on  $P_{cgo}$ . For any given water saturation  $s_{wi}$  (Figure CP3a) where there are three mobile phases ( $s_{wi}, s_{oi}, s_{gi} > 0$ ), the two-phase capillary pressure is  $P_{cowi}$ . When  $s_{gi} = 0$  and so only two phases are mobile,  $P_{cow}$  is sufficient and the correction is not needed.

Now imagine that oil phase in this system ( $s_{oi} = 1 - s_{wi}$ ) is replaced gradually by gas phase. Again there are two limiting cases. At  $s_{gi} = 0$  the three-phase oil-water capillary pressure  $P_{cow3}$  is identical to the two-phase oil-water capillary  $P_{cow}$ , i.e.,  $P_{cow3} = P_{cowi}$ . At the other limit  $s_{gi} = 1 - s_{wi}$  the gas phase replaces the oil phase completely and the two-phase oil-water capillary pressure  $P_{cow}$  approaches its threshold value  $P_{cows}$ . This situation is similar physically to oil invading a water zone. At any arbitrary  $s_{gi}$  (and  $s_{wi}$ ) the three-phase oil-water capillary pressure is obtained by interpolation formula

$$P_{cow3}(s_{wi}, s_{gi}) = P_{cows} + F(s_{gi}) \times [P_{cowi} - P_{cows}] \quad (\text{Eq. CP6})$$

where weighting function  $F(s_{gi})$  is

$$F(s_{gi}) = \left[ \frac{\frac{1}{\frac{1 - s_{wi} - s_{gi} + \varepsilon_g}{1 - s_{wi}} - \frac{1}{\varepsilon_g}}} {\frac{1}{\frac{1 - s_{wi} + \varepsilon_g}{1 - s_{wi}} - \frac{1}{\varepsilon_g}}} \right]^{n_g} \quad (\text{Eq. CP7})$$

and

$s_{wi}$ :	Grid water saturation at the time;
$s_{gi}$ :	Grid gas saturation at the time;
$P_{cowi}$ :	The oil-water two-phase capillary pressure at $s_{wi}$ ;
$P_{cows}$ :	The threshold value of oil-water two-phase capillary pressure;
$\varepsilon_g$ :	User defined curvature fitting parameter;
$n_g$ :	User defined curve fitting exponential.

Function  $F(S_{gi})$  satisfies:

$$0 \leq F(s_{gi}) \leq 1$$

and

$$F(s_{gi}) = 0 \rightarrow P_{cow3} = P_{cows} \quad \text{at } s_{gi} = 1 - s_{wi};$$

$$F(s_{gi}) = 1 \rightarrow P_{cow3} = P_{cowi} \quad \text{at } s_{gi} = 0.$$

For any  $s_{gi}$  satisfying  $0 \leq s_{gi} \leq 1 - s_{wi}$ , the three-phase oil-water capillary pressure is smaller than its two-phase counterpart, that is  $P_{cow3}(s_{wi}, s_{gi}) < P_{cowi}(s_{wi})$ .

An example of a three-phase  $P_{cow3}$  surface is drawn in Figure CP4. The surface is concave upward. The slope of curve AB and the curvature of the surface are influenced by the value of  $\varepsilon_g$ .

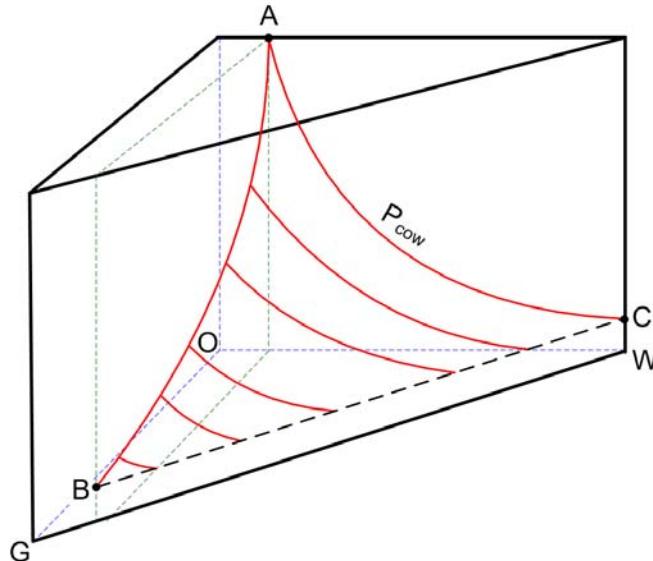


Figure CP4:  $P_{cow3}$  surface

### Application of Third-phase Capillary Pressure Correction

The following are several applications of the capillary pressure third-phase correction along with the  $P_{cgw}$  curve.

- i) ***Initially two-phase water-gas system, then gas condenses*** The reservoir is initialized as a two-phase water-gas system but oil phase (condensed HC gas) may appear and disappear later in the simulation. If \*VERTICAL \*DEPTH\_AVE is used, specify \*TRANZONE along with \*DWOC = \*DGOC. If a gas/water transition zone is intended, specify  $P_{cgw}$  via \*SWT table and subkeyword \*PCGW. If other capillary pressures are important when oil phase is present, specify  $P_{cow}$  and/or  $P_{cgo}$  in saturation tables (their defaults are zero). Use the third-phase correction option \*PC\_3RD\_PHASE to avoid discontinuity in capillary pressures as oil phase appears and disappears.
- ii) ***Initially three phase but oil disappears*** The reservoir is initially in three phases with or without transition zones. To simulate the interfacial behavior between gas and water when oil disappears, specify the  $P_{cgw}$  curve along with  $P_{cow}$  and/or  $P_{cgo}$ . Use the third-phase correction option \*PC\_3RD\_PHASE to avoid discontinuity in capillary pressures as oil phase appears and disappears.
- iii) ***The gas-water capillary of local importance*** If a gas-water two-phase state exists at a number of important locations, simulate the interfacial behavior between gas and water by specifying the  $P_{cgw}$  curve along with  $P_{cow}$  and/or  $P_{cgo}$ . Use the third-phase correction option \*PC\_3RD\_PHASE to avoid discontinuity in capillary pressures as oil phase appears and disappears.
- iv) ***The third phase correction is believed to be important*** If two-phase capillary pressures  $P_{cow}$  and  $P_{cgo}$  alone are not sufficient to represent the interfacial behavior, use the third-phase correction feature as a tool. In this case, specify the  $P_{cgw}$  curve along with  $P_{cow}$  and  $P_{cgo}$  and adjust the parameters of \*PC\_3RD\_PHASE to mimic the reservoir physics.

#### EXAMPLE:

The following data enables the third-phase correction feature to ensure a smooth transition of capillary pressure if oil phase appears or disappears.

```

*ROCKFLUID
*SWT      *PCGW
**        sw       krw      krow     pcgw      pcow
          0.3210   0.0000   1.0000   29.2      14.1
          0.4560   0.0018   0.7080   29.1      10.5
          .
          0.8011   0.1130   0.0130   6.1       4.1
          0.8200   0.1200   0.0000   4.9       2.9
*PC_3RD_PHASE *EPCOW 10  *EPCGO 10  *EXPCW 4  *EXPCG 4
*SLT
**        sl       krg      krog     pcgo
          0.5010   0.0380   0.0000   15.1
          .
          0.9500   0.0000   1.0000   1.55

*PC_3RD_PHASE *EPCOW 10  *EPCGO 10  *EXPCW 4  *EXPCG 4

```

---

## Subdomain Reinfiltration (Optional)

\*SD\_REINF

### PURPOSE:

Modify capillary continuity for inter-Subdomain interblock connections.

### ARRAY:

\*SD\_REINF

### DEFAULTS:

If there are inter-Subdomain interblock connections but \*SD\_REINF is absent, full matrix/matrix capillary continuity is assumed in those connections (\*SD\_REINF values = 0).

### CONDITIONS:

Keyword \*SD\_REINF is meaningful and effective only for inter-Subdomain connections (see keyword \*TRANSD).

Array qualifiers \*FRACTURE and \*MATRIX are not allowed.

### EXPLANATION:

Keyword \*TRANSD in the Reservoir Description section creates the SUBDOMAIN-DK inter-Subdomain connection between matrix stacks in vertically adjacent spatial blocks. This allows fluid flow (e.g., gravity drainage) to occur directly between matrix stacks across the horizontal fracture plane.

Normally there is full capillary continuity across an inter-Subdomain connection. Keyword \*SD\_REINF allows you to reduce capillary continuity across this connection, from partial to complete discontinuity.

When \*SD\_REINF is defined, as far as the capillarity is concerned the bottommost matrix subblock in a matrix stack “stands in” for the fracture in modeling flow from matrix to matrix grid. It is as if fluid flows from the upper matrix subblock the horizontal fracture and then re-infiltrates from the fracture into the underlying matrix subblock. This process works best when \*SUBDOMAIN subkeyword \*FRACVOL specifies a smaller thickness for the bottommost subblock which will be used to “stand in” for the horizontal fracture.

Normally the reinfiltration model is controlled by two values of \*SD\_REINF: 1.0 turns it on, while 0.0 turns it off. However, intermediate values of \*SD\_REINF may be used to account for limited reinfiltration occurring together with gravity drainage.

\*SD\_REINF specifies some level of capillary discontinuity by modifying the bottommost subblock’s  $P_{cow}$  and  $P_{cog}$ . Let  $F_{reinf}$  be the value entered via \*SD\_REINF for a particular block. The capillary pressures of that spatial block’s bottommost subdomain block are adjusted as follows.

$$P_{cow} = P_{cow} \cdot (1 - F_{reinf})$$

$$P_{cog} = P_{cog} \cdot (1 - F_{reinf})$$

For details of \*SD\_REINF, see "Simulation Gravity Drainage and Reinfiltration with a Subdomain-Dual Permeability Hybrid Fracture Model", SPE 106191, by Barry Rubin,

### **Examples:**

The following is an example of \*TRANSD and \*SD\_REINF usage:

```
**In Reservoir Description section, turn on direct gravity
**drainage between subdomains for all grids
*SUBDOMAIN 4
*TRANSD *CON 1.0
:
**In Rock-Fluid Data section, introduce capillary
**discontinuity in Inter-Subdomain flow for all grids
*SD_REINF CON 1.0
```

The suggested range of values for SD\_REINF is:

	<b>SI</b>	<b>Field</b>	<b>Lab</b>
min	0.0	0.0	0.0
max	1.0	1.0	1.0

---

## **Effective Molecular Diffusion Coefficients**

**\*DIFFI\_WAT,**

**\*DIFFJ\_WAT, \*DIFFK\_WAT, \*DIFFI\_OIL, \*DIFFJ\_OIL, \*DIFFK\_OIL, \*DIFFI\_GAS,  
\*DIFFJ\_GAS, \*DIFFK\_GAS**

### **PURPOSE:**

Enter effective molecular diffusion coefficients for the desired component and phase.

### **ARRAY:**

<b>*DIFFI_WAT</b>	<i>comp_name</i>
<b>*DIFFJ_WAT</b>	<i>comp_name</i>
<b>*DIFFK_WAT</b>	<i>comp_name</i>
<b>*DIFFI_OIL</b>	<i>comp_name</i>
<b>*DIFFJ_OIL</b>	<i>comp_name</i>
<b>*DIFFK_OIL</b>	<i>comp_name</i>
<b>*DIFFI_GAS</b>	<i>comp_name</i>
<b>*DIFFJ_GAS</b>	<i>comp_name</i>
<b>*DIFFK_GAS</b>	<i>comp_name</i>

**\*TORTU ( \*INCPORSAT | \*NOPORSAT)**

### **DEFINITIONS:**

**\*DIFFI\_WAT, \*DIFFJ\_WAT, \*DIFFK\_WAT**

Effective molecular diffusion coefficients ( $\text{m}^2/\text{day}$  |  $\text{ft}^2/\text{day}$ ) of *comp\_name* in the water phase for the I, J and K directions. Array reading option \*EQUALSI is allowed for the J and K directions.

**\*DIFFI\_OIL, \*DIFFJ\_OIL, \*DIFFK\_OIL**

Effective molecular diffusion coefficients ( $\text{m}^2/\text{day}$  |  $\text{ft}^2/\text{day}$ ) of *comp\_name* in the oil phase for the I, J and K directions. Array reading option \*EQUALSI is allowed for the J and K directions.

**\*DIFFI\_GAS, \*DIFFJ\_GAS, \*DIFFK\_GAS**

Effective molecular diffusion coefficients ( $\text{m}^2/\text{day}$  |  $\text{ft}^2/\text{day}$ ) of *comp\_name* in the gas phase for the I, J and K directions. Array reading option \*EQUALSI is allowed for the J and K directions.

*comp\_name*

Quoted component name. The component must be found in the specified phase, as determined by \*MODEL and the K value data entered.

**\*TORTU ( \*INCPORSAT | \*NOPORSAT )**

Choose the option for tortuosity (formation resistivity) which defines the molecular diffusion coefficients entered as data. If \*INCPORSAT is selected (the default) then the coefficient includes the factor  $\phi S_j$ , that is, the data entered is  $\phi S_j D_{ij}^* / F_{jk}$  (see below). If \*NOPORSAT is selected then factor  $\phi S_j$  is not included so the data entered is  $D_{ij}^* / F_{jk}$ , to which the current value of  $\phi S_j$  is applied when the diffusion amount is calculated.

A single \*TORTU selection applies to all components and phases.

### DEFAULTS:

If there is no molecular diffusion keyword for a component/phase combination, then there is no molecular diffusion of that component in that phase.

If keyword \*TORTU is not present then \*TORTU \*INCPORSAT is assumed.

### CONDITIONS:

For each component/phase combination specified, all three directions must be specified.

The molecular diffusion option may not be used together with the total dispersion option (keywords \*DISPI\_WAT, etc.).

The dependence of molecular diffusion on temperature and viscosity depends upon the use of keyword \* MOLDIFF\_DEP.

### EXPLANATION:

The propagation of injected tracers and chemicals employed in EOR processes are influenced by the tortuous flow paths and (random) heterogeneities of the porous media in which they flow. Normally this contribution to dispersion - the broadening and spreading of concentration fronts - dominates that due to molecular diffusion. Because of this, much useful information on porous media structure can be gained from the analysis of dispersion.

Dispersion is affected by transmissibility multipliers specified by keywords \*TRANSI, etc. Therefore, if the transmissibility multiplier for a given pair of adjacent blocks is decreased, then both the convective flow and dispersive flow will decrease accordingly.

In some circumstances these coefficients may be regarded as adjustable parameters that need to be tuned to give acceptable results. Indeed, laboratory values may not correspond to what is needed for large grid blocks used in reservoir simulation.

The flux  $J_{ijk}$  of component i in phase j in direction k due to diffusion is given by:

$$J_{ijk} = -(\varphi S_j D^*_{ij} / F_{jk}) \nabla_k (\rho_j X_{i,j})$$

where

$\varphi, S_j$  = porosity, saturation of phase j  
 $D^*_{ij}$  = molecular diffusion coefficient of component i in phase j, specified via keywords \*DIFFI\_WAT, etc., and \* MOLDIFF\_DEP.

$F_{jk}$  = tortuosity for phase j in direction k  
 $\nabla_k (\rho_j X_{i,j})$  = concentration gradient of component i in phase j in direction k

The coefficients entered here are *effective* because they include the effect of formation tortuosity and optionally  $\varphi S_j$ .

Tortuosity is defined as the ratio of the true path length traveled by a particle flowing through the medium to the macroscopic distance traveled.

When molecular diffusion is being modelled it may be necessary to run in fully implicit mode (\*AIM \*OFF in Numerical Control).

## Examples

```
** Molecular diffusion
*DIFFI_OIL 'C3H8' *CON 0.03      ** C3H8 in oil phase
*DIFFJ_OIL 'C3H8' *CON 0.03
*DIFFK_OIL 'C3H8' *CON 0.03
*DIFFI_OIL 'C7H16' *CON 0.02      ** C7H16 in oil phase
*DIFFJ_OIL 'C7H16' *CON 0.02
*DIFFK_OIL 'C7H16' *CON 0.02
*DIFFI_GAS 'C3H8' *CON 0.05      ** C3H8 in gas phase
*DIFFJ_GAS 'C3H8' *CON 0.05
*DIFFK_GAS 'C3H8' *CON 0.05
```

---

## Temperature and Viscosity Dependence of Diffusion

\*MOLDIFF\_DEP

### PURPOSE:

Specify temperature and viscosity dependence of molecular diffusion for the desired component and phase.

### ARRAY:

\*MOLDIFF\_DEP *comp\_name phase* ( \*TDEP *T<sub>ref</sub>*) ( \*VISDEP  $\mu_{ref}\beta$  )

### DEFINITIONS:

*comp\_name phase*

The dependence is applied to the component with name *comp\_name* (in quotes) in the fluid phase indicated by *phase* (see table below). The component must be found in that phase, as determined by \*MODEL and the K value data entered.

<i>phase</i>	Fluid phase
*WATER	water (aqueous)
*OIL	oil (oleic)
*GAS	gas (gaseous)

\*TDEP *T<sub>ref</sub>*

Temperature dependence is applied to molecular diffusion by multiplying by ratio ( $T/T_{refabs}$ ), where T is the current temperature in absolute degrees and  $T_{refabs}$  is the reference temperature  $T_{ref}$  (C | F) converted to absolute degrees.

\*VISDEP  $\mu_{ref}\beta$

Viscosity dependence is applied to molecular diffusion by multiplying by ratio  $(\mu_{ref}/\mu)^\beta$ , where  $\mu$  is the current *phase* viscosity and  $\mu_{ref}$  is the reference *phase* viscosity (cp). Note that  $\beta = 0$  will defeat viscosity dependence.

### DEFAULTS:

If \*TDEP is absent for a diffusing component/phase combination, then the corresponding molecular diffusion coefficient does not change with temperature.

If \*VISDEP is absent for a diffusing component/phase combination, then the corresponding molecular diffusion coefficient does not change with viscosity.

### CONDITIONS:

Keyword \*MOLDIFF\_DEP is effective only if molecular diffusion data has been specified for the same component/phase combination via keywords \*DIFFI\_WAT, etc.

Keyword \*MOLDIFF\_DEP requires that \*TORTU \*NOPORSAT be used.

## **EXPLANATION:**

This keyword lets you specify dependence of molecular diffusion on temperature and viscosity in the form

$$\mathbf{D}^*_{ij} \text{ proportioned to } T / \mu^\beta$$

where  $T$  is temperature in absolute degrees and  $\mu$  is viscosity.

Let  $\mathbf{D}^u_{ij}$  be molecular diffusion coefficients entered by the user via keywords \*DIFFI\_WAT, etc. When calculating diffusion at a block's local conditions, the coefficient  $\mathbf{D}^*_{ij}$  described in that keyword's EXPLANATION is affected by \*MOLDIFF\_DEP in the following ways.

*TDEP	*VISDEP	$\mathbf{D}^*_{ij}$ is assigned value
Absent	Absent	$\mathbf{D}^u_{ij}$
Present	Absent	$\mathbf{D}^u_{ij} \cdot (T/T_{refabs})$
Absent	Present	$\mathbf{D}^u_{ij} \cdot (\mu_{ref}/\mu)^\beta$
Present	Present	$\mathbf{D}^u_{ij} \cdot (T/T_{refabs}) \cdot (\mu_{ref}/\mu)^\beta$

Some references are

$$\beta = 1$$

Stokes-Einstein

$$\beta = 1$$

Wilke, C.R., Chang, P, "Correlation of Diffusion Coefficients in Dilute Solutions", A.I.Ch.E. Journal, June 1955, P 264-270.

$$\beta = 0.46$$

Hayduk, W.; Castaneda, R.; Bromfield, H.; Perras, R.R., 1973. Diffusivities of Propane in Normal Paraffin, Chlorobenzene and Butanol Solvents, AIChEJ. 19, pp 859-861.

$$\beta = 0.545$$

Das, S.K. and Butler, R.M., 1996. "Diffusion Coefficients of Propane and Butane in Peace River Bitumen", Canadian Journal of Chemical Engineering, pp985-991, Vol. 74, Dec.

---

**Mechanical Dispersivity** \*MDSPI\_WAT, \*MDSPJ\_WAT, \*MDSPK\_WAT,  
\*MDSPI\_OIL, \*MDSPJ\_OIL, \*MDSPK\_OIL, \*MDSPI\_GAS, \*MDSPJ\_GAS, \*MDSPK\_GAS

**PURPOSE:**

Enter mechanical (convective) dispersivity for the desired phase.

**ARRAY:**

\*MDSPI\_WAT  
\*MDSPJ\_WAT  
\*MDSPK\_WAT  
\*MDSPI\_OIL  
\*MDSPJ\_OIL  
\*MDSPK\_OIL  
\*MDSPI\_GAS  
\*MDSPJ\_GAS  
\*MDSPK\_GAS

**DEFINITIONS:**

\*MDSPI\_WAT, \*MDSPJ\_WAT, \*MDSPK\_WAT

Mechanical dispersivity (m | ft) in the water phase for the I, J and K directions.  
Array reading option \*EQUALSI is allowed for the J and K directions.

\*MDSPI\_OIL, \*MDSPJ\_OIL, \*MDSPK\_OIL

Mechanical dispersivity (m | ft) in the oil phase for the I, J and K directions.  
Array reading option \*EQUALSI is allowed for the J and K directions.

\*MDSPI\_GAS, \*MDSPJ\_GAS, \*MDSPK\_GAS

Mechanical dispersivity (m | ft) in the gas phase for the I, J and K directions.  
Array reading option \*EQUALSI is allowed for the J and K directions.

**DEFAULTS:**

If there is no mechanical dispersivity keyword for a phase, then there is no mechanical dispersivity in that phase.

**CONDITIONS:**

For each reservoir rock region specified, all three directions must be specified.

The mechanical dispersivity option may not be used together with the total dispersion option flagged for use by keywords \*DISPI\_WAT, etc.

**EXPLANATION:**

The propagation of injected tracers and chemicals employed in EOR processes are influenced by the tortuous flow paths and (random) heterogeneities of the porous media in which they flow. Normally this contribution to dispersion - the broadening and spreading of concentration fronts - dominates that due to molecular diffusion. Because of this, much useful information on porous media structure can be gained from the analysis of dispersion.

The above keywords allow dispersion coefficients to depend on region, direction and phase. For simulation, the input dispersion coefficients should be viewed as the true physical dispersion coefficients minus the numerical dispersion introduced by truncation error.

When mechanical dispersion is being modelled it may be necessary to run in fully implicit mode (\*AIM \*OFF in Numerical Control).

Dispersion is affected by transmissibility multipliers specified by keywords \*TRANSI, etc. Therefore, if the transmissibility multiplier for a given pair of adjacent blocks is decreased, then both the convective flow and dispersive flow will decrease accordingly.

### Theory

Dispersion is the mixing of fluids caused by diffusion, local velocity gradients, locally heterogeneous streamline lengths, and mechanical mixing (Lake, 1989). The mechanical dispersive flux  $J_{ijk}$  of component i in phase j in direction k is given by:

$$J_{ijk} = -\varphi S_j \alpha_{jk} |u_j| \nabla_k (\rho_j X_{i,j})$$

where

$\varphi, S_j$	= porosity, saturation of phase j
$\alpha_{jk}$	= dispersivity for phase j in direction k
$ u_j $	= magnitude of interstitial velocity of phase j
$\nabla_k (\rho_j X_{i,j})$	= concentration gradient of component i in phase j in direction k

### Longitudinal and Transverse Dispersivity

With regard to the issue of longitudinal versus transverse dispersivity, it is assumed implicitly that the predominant flow is in one grid direction throughout the simulation. In tensor terms, the dispersion tensor is diagonal. The longitudinal value can be assigned to the predominant flow direction, and the transverse value can be assigned to the other two directions.

However, the flexibility of data entry makes it possible to use variations of this strategy as well as completely different strategies.

### Examples

```
** Mechanical dispersivity
*MDSPI_OIL  *CON 0.03  ** All components in oil phase
*MDSPJ_OIL  *CON 0.03
*MDSPK_OIL  *CON 0.03
*MDSPI_GAS  *CON 0.05  ** All components in gas phase
*MDSPJ_GAS  *CON 0.05
*MDSPK_GAS  *CON 0.05
```

---

## Total Dispersion Coefficients      \*DISPI\_WAT, \*DISPJ\_WAT, \*DISPK\_WAT, \*DISPI\_OIL, \*DISPJ\_OIL, \*DISPK\_OIL, \*DISPI\_GAS, \*DISPJ\_GAS, \*DISPK\_GAS

### PURPOSE:

Enter total dispersion coefficients for the desired component and phase.

### ARRAY:

*DISPI_WAT	<i>comp_name</i>
*DISPJ_WAT	<i>comp_name</i>
*DISPK_WAT	<i>comp_name</i>
*DISPI_OIL	<i>comp_name</i>
*DISPJ_OIL	<i>comp_name</i>
*DISPK_OIL	<i>comp_name</i>
*DISPI_GAS	<i>comp_name</i>
*DISPJ_GAS	<i>comp_name</i>
*DISPK_GAS	<i>comp_name</i>

### DEFINITIONS:

#### \*DISPI\_WAT, \*DISPJ\_WAT, \*DISPK\_WAT

Effective total dispersion coefficients ( $\text{m}^2/\text{day}$  |  $\text{ft}^2/\text{day}$ ) of *comp\_name* in the water phase for the I, J and K directions. Array reading option \*EQUALSI is allowed for the J and K directions.

#### \*DISPI\_OIL, \*DISPJ\_OIL, \*DISPK\_OIL

Effective total dispersion coefficients ( $\text{m}^2/\text{day}$  |  $\text{ft}^2/\text{day}$ ) of *comp\_name* in the oil phase for the I, J and K directions. Array reading option \*EQUALSI is allowed for the J and K directions.

#### \*DISPI\_GAS, \*DISPJ\_GAS, \*DISPK\_GAS

Effective total dispersion coefficients ( $\text{m}^2/\text{day}$  |  $\text{ft}^2/\text{day}$ ) of *comp\_name* in the gas phase for the I, J and K directions. Array reading option \*EQUALSI is allowed for the J and K directions.

#### *comp\_name*

Quoted component name. The component must be found in the specified phase, as determined by \*MODEL and the K value data entered.

### DEFAULTS:

For each component/phase combination, if there is no total dispersion keyword then the total dispersion is determined by the corresponding molecular diffusion and mechanical dispersivity keywords.

## CONDITIONS:

For each component/phase combination specified, all three directions must be specified.

The total dispersion option may not be used together with the molecular diffusion option (keywords \*DIFFI\_WAT, etc.) or the mechanical dispersivity option (keywords \*MDSPI\_WAT, etc.).

## EXPLANATION:

The propagation of injected tracers and chemicals employed in EOR processes are influenced by the tortuous flow paths and (random) heterogeneities of the porous media in which they flow. Normally this contribution to dispersion - the broadening and spreading of concentration fronts - dominates that due to molecular diffusion. Because of this, much useful information on porous media structure can be gained from the analysis of dispersion.

Most generally, dispersion coefficients are found to be region and direction dependent, as well as being different in different phases for species which partition in multiple phases. This variation can be captured with the allowed flexible input options. For simulation, the input dispersion coefficients should be viewed as the true physical dispersion coefficients minus the numerical dispersion introduced by truncation error.

The total dispersive flux  $J_{ijk}$  of component i in phase j in direction k is given by:

$$J_{ijk} = -\mathbf{D}_{ijk} \nabla_k (\rho_j x_{ij})$$

where

$$\begin{aligned}\mathbf{D}_{ijk} &= \text{total dispersion coefficient of component } i \text{ in phase } j \text{ in direction } k \\ \nabla_k (\rho_j x_{ij}) &= \text{concentration gradient of component } i \text{ in phase } j \text{ in direction } k\end{aligned}$$

Total dispersion is made up of two parts: effective molecular diffusion (which is component and phase dependent), and mechanical dispersion (which is a property of the reservoir rock), as follows. Dispersion is affected by transmissibility multipliers specified by keywords \*TRANSI, etc. Therefore, if the transmissibility multiplier for a given pair of adjacent blocks is decreased, then both the convective flow and dispersive flow will decrease accordingly.

## Examples

```
** Total dispersion with natural fracture grid option
** Component "COMB WAT" in water phase
*DISPI_WAT 'COMB WAT' *MATRIX *CON 1e-4
*DISPJ_WAT 'COMB WAT' *MATRIX *EQUALSI
*DISPK_WAT 'COMB WAT' *MATRIX *EQUALSI
*DISPI_WAT 'COMB WAT' *FRACTURE *CON 1e-2
*DISPJ_WAT 'COMB WAT' *FRACTURE *EQUALSI
*DISPK_WAT 'COMB WAT' *FRACTURE *EQUALSI
** COMPONENT "OXYGEN" IN GAS PHASE
*DISPI_GAS 'OXYGEN' *MATRIX *CON 3e-4
*DISPJ_GAS 'OXYGEN' *MATRIX *EQUALSI
*DISPK_GAS 'OXYGEN' *MATRIX *EQUALSI
*DISPI_GAS 'OXYGEN' *FRACTURE *CON 3e-2
*DISPJ_GAS 'OXYGEN' *FRACTURE *EQUALSI
*DISPK_GAS 'OXYGEN' *FRACTURE *EQUALSI
```

## Adsorbing Component Functions

\*ADSCOMP, \*ADSLANG,

\*ADSTABLE

### PURPOSE:

Assign composition and temperature dependence of component adsorption.

### FORMAT:

```
*ADSCOMP comp_name phase_des
*ADSLANG tad1 tad2 tad3
-or-
*ADSLANG *TEMP
{ tads tad1 tad2 tad3 }
-or-
*ADSLANG ( *2CMPW | *2CMPX | *2CMPY )
{ 2cads tad1 tad2 tad3 }
-or-
*ADSTABLE
{ cpt adt }
-or-
*ADSTABLE
{ *TEMP tads
{ cpt adt } }
-or-
*ADSTABLE
{ ( *2CMPW | *2CMPX | *2CMPY ) 2cads
{ cpt adt } }
```

### DEFINITIONS:

*comp\_name*

Quoted name of component to which the following adsorption function will apply.

*phase\_des*

Phase from which the adsorbing component's composition dependence will be taken:

'WATER'	water (aqueous) mole fraction
'OIL'	oil (oleic) mole fraction
'GAS'	gas mole fraction
'GLOBAL'	global mole fraction
'MAX'	maximum of water, oil and gas mole fractions

\*ADSLANG

Denotes that composition dependence is specified via Langmuir isotherm coefficients. If \*TEMP is present, enter a table of coefficients versus T (maximum of 30 entries).

*tad1*

First parameter in the Langmuir expression for the adsorption isotherm (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>). It must be non-negative.

*tad2*

Second parameter in the Langmuir expression for the adsorption isotherm associated with salt effects (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>). It must be non-negative.

At present this coefficient is not used. Enter 0.

*tad3*

Third parameter in the Langmuir expression for the adsorption isotherm. It must be no less than 1e-15.

*tads*

Temperature of the isothermal adsorption data (Langmuir or table).

*2cads*

Concentration value of second component affecting the adsorption data (Langmuir or table), for component \*INTCOMP. This additional concentration dependence is optional. The identity and phase of the second component is determined by the choice of keyword in the following table:

\*2CMPW - component number NUMW in aqueous phase

\*2CMPX - component number NUMX in oleic phase

\*2CMPY - component number NUMY in gaseous phase

**\*ADSTABLE**

Denotes that composition dependence is specified via a table of adsorption versus composition (maximum of 30 entries). If \*TEMP is present, enter the table for different values of temperature *tads* (maximum of 30).

*cpt*

Mole fraction of *comp\_name* in *phase\_des*. The allowed range is 0 to 1. Table entries *cpt* must increase by at least 1e-10.

*adt*

Adsorbed moles per unit pore volume at composition *cpt* (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>). Table entries *adt* must increase by more than 1e-10.

## **DEFAULTS:**

If subkeyword \*TEMP is absent, the adsorption is assumed to be independent of temperature. If none of the subkeywords \*2CMPW, \*2CMPX or \*2CMPY is present, adsorption is independent of additional components.

## **CONDITIONS:**

\*ADSCOMP and either \*ADSLANG or \*ADSTABLE must be present for each adsorbing component.

\*ADSCOMP must appear only once for each adsorbing component.

## **EXPLANATION:**

The Langmuir adsorption isotherm gives the adsorbed moles of component MM per unit pore volume as

$$ad = \frac{(tad1 + tad2 * xnacl) * ca}{(1 + tad3 * ca)}$$

where xnacl is the salinity of the brine, and ca is the mole fraction of *comp\_name* in *phase\_des*. At high concentrations (large ca) the maximum adsorption is  $(tad1 + tad2 * xnacl)/tad3$ .

### **Single-Component Phase**

If *phase\_des* indicates a single phase (WATER, OIL or GAS) then *comp\_name* must not be the only component found in that phase. The adsorption model varies that component's mole fraction to achieve a balance of that component's moles in both the reference phase and the adsorbed phase. When *comp\_des* is the only component in *phase\_des*, its mole fraction is constant at 1 and the adsorption model cannot work properly.

Table "PHASE DISTRIBUTION OF COMPONENTS" in the text output file reports on the component-phase distribution resulting from the \*MODEL and K-value data entered. If *phase\_des* is a single-component phase according to this table, then an error message is issued when the offending \*ADSCOMP keyword is read and the run stops before the table is printed. In order to see this table you must temporarily remove the offending adsorption data.

### **Phase Disappearance**

The amount of adsorption depends only on phase mole fraction and not the amount of that phase present. This model is intended for an injected component which achieves a modest maximum mole fraction in a phase that is always present, such as injected chemical. Therefore, if *phase\_des* indicates a single phase (WATER, OIL or GAS) then that phase must not disappear or become small.

---

## **Rock-Dependent Adsorption Data**

**\*ADSPHBLK, \*PORFT, \*RRFT, \*ADSTYPE**

**\*ADSROCK, \*ADMAXT, \*ADRT,**

### **PURPOSE:**

Assign rock (permeability) dependence of adsorption data for component/phase indicated via \*ADSCOMP.

### **FORMAT:**

**\*ADSROCK *nrock***  
**\*ADMAXT *admaxt***  
**\*ADRT *adrt***  
**\*ADSPHBLK *phase\_des***  
**\*PORFT *porft***  
**\*RRFT *rrft***

### **ARRAY:**

**\*ADSTYPE**

### **DEFINITIONS:**

**\*ADSROCK *nrock***

Specify the current rock type number for the keywords \*ADMAXT, \*ADRT, \*PORFT, \*RRFT and \*ADSPHBLK. The default is 1. This keyword is necessary only if you have multiple adsorption rock types.

***admaxt***

Maximum adsorption capacity (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>). It must be positive.

***adrt***

Residual adsorption level (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>). The allowed range is from 0 to *admaxt*.

A zero value implies that adsorption is completely reversible, while *adrt* = *admaxt* denotes completely irreversible adsorption.

***phase\_des***

Over-rides the default phase to which the resistance factor calculation is applied

**\*W** water (aqueous) phase  
**\*O** oil (oleic) phase  
**\*G** gas (gaseous) phase  
**\*ALL** all phases

Normally, the resistance factor is applied only to the fluid phase which is the source of the adsorbing component. \*ADSPHBLK makes it possible to apply the resistance factor of another or all fluid phases.

*porft*

Accessible pore volume or fraction of available pore volume. The allowed range is from 0 to 1.

It can be viewed also as one minus the fraction of pore volume that is inaccessible to the component.

*rrft*

Residual resistance factor for the adsorbing component. It must be greater than or equal to 1. The default is 1.

**\*ADSTYPE**

Assign multiple adsorption rock type numbers to grid blocks.

### **DEFUALTS:**

The following defaults apply to all adsorbing components in all adsorbing rock types.

- \*ADSROCK 1
- \*ADMAXT 0 (no adsorption)
- \*ADRT 0 (completely reversible adsorption))
- \*PORFT 1 (no inaccessible pore volume)
- \*RRFT 1 (no resistance effect)
- \*ADSTYPE \*CON 1

If keyword \*ADSPHBLK is absent, then the resistance factor is applied to the phase which is the source of the adsorbing component.

### **CONDITIONS:**

\*ADMAXT is required for adsorption.

Keywords \*ADMAXT, \*ADRT, \*ADSPHBLK, \*PORFT and \*RRFT apply only to the current component specified by \*ADSCOMP and the current rock type specified by \*ADSROCK.

### **EXPLANATION:**

Adsorption properties such as component retention, residual resistance factor, inaccessible pore volume and desorption level depend upon the formation permeability. Reservoir heterogeneities can cause these properties to vary significantly within a reservoir. Therefore, equilibrium adsorption is a function of location as well as component concentration and temperature.

This is accounted for by scaling the adsorption obtained from local concentration and temperature conditions by the factor

$$\text{ADMAXT}(I) / \text{AD}_{\max,T_1}$$

where  $\text{ADMAXT}(I)$  is the maximum adsorption capacity at grid block I, and  $\text{AD}_{\max,T_1}$  is the maximum possible adsorption obtainable from the adsorption isotherm of the first input temperature, that is, first *tads* in keyword \*ADSLANG or \*ADSTABLE.

Thus

$$ad(C,T,I) = ADMAXT(I) * ad(C,T) / AD_{max,T1}$$

The reduced porosity for adsorbing component ic, adsorption rock type k, at grid block i is  
 $porft(k,ic) * por(p(i),T(i))$

where por is the usual porosity calculated from the block pressure and temperature.

Adsorption or mechanical entrapment can cause blockage which amounts to a reduction in the effective permeability. This is accounted for by the permeability reduction factors

$$RKW = 1 + (RRF-1) * AD(C,T)/ADMAXT$$

$$RKO = 1 + (RRF-1) * AD(C,T)/ADMAXT$$

$$RKG = 1 + (RRF-1) * AD(C,T)/ADMAXT$$

which affects the permeabilities AKW(I), AKO(I), AKG(I) as

$$AKW(I) = AK(I) * krw/RKW(I)$$

where AK(I) is standard block permeability. A similar definition holds for the oil and gas phases AKO and AKG. Thus the relative mobility of a phase containing an adsorbing component is most generally affected by viscosity (possibly non-Newtonian) and blockage.

### Multiple Rock Types with Multiple Components

If there are both multiple adsorbing components and multiple adsorption rock types, then specify the components in the “outer” loop and the rock types in the “inner” loop. Note that \*ADSCOMP for a component must appear only once, but \*ADSROCK for a rock type can appear more than once.

For example, for two adsorbing components and three rock types, use the following.

```
*ADSCOMP 'Adsorb 1'  
  *ADSLANG or *ADSTABLE ...  
  *ADSROCK 1  
    *ADMAXT ...  
  *ADSROCK 2  
    *ADMAXT ...  
  *ADSROCK 3  
    *ADMAXT ...  
*ADSCOMP 'Adsorb 2'  
  *ADSLANG or *ADSTABLE  
  *ADSROCK 1  
    *ADMAXT ...  
  *ADSROCK 2  
    *ADMAXT ...  
  *ADSROCK 3  
    *ADMAXT ...
```



# Initial Conditions

---

## Initial Conditions Identifier (Required)

\*INITIAL

### PURPOSE:

\*INITIAL indicates the beginning of initial condition values.

### FORMAT:

\*INITIAL

### DEFAULTS:

Required keyword.

### 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.

The only required keywords in this section are those which define the initial pressure distribution.

---

## Initialization Regions (Optional)

\*INITREGION, \*INTYPE

### PURPOSE:

Specify multiple initialization regions.

### FORMAT:

\*INITREGION *key*

### ARRAY:

\*INTYPE

### DEFINITIONS:

\*INITREGION *key*

Introduces an initialization region to which data is assigned. The numerical *key* is an integer from 1 to the maximum number of initialization regions.

All regions from 1 to the maximum must be specified and may appear in any order.

\*INTYPE

Assigns initialization regions to grid blocks. The value for each grid block is one of the *key* values defined by an \*INITREGION keyword. Dump this grid array to the SR2 via \*OUTSRF \*GRID \*INSETN.

### DEFAULTS:

If \*INITREGION is absent there is only one initialization region.

If \*INTYPE is absent then all blocks use region #1.

### CONDITIONS:

All keys assigned to \*INTYPE must be defined via \*INITREGION.

### EXPLANATION:

An initialization region is a collection of grid blocks to which a single set of the following keywords apply: \*REFPRES, \*REFDEPTH, \*REFBLOCK, \*TRANZONE, \*DWOC, \*DGOC and \*WOC\_SW. The following data fragment defines two hydraulically disconnected regions.

```
*VERTICAL *DEPTH_AVE
*INITREGION 1      ** === Upper member ===
  *REFDEPTH 3000  ** ft
  *REFPRES 1800   ** psi
  *DWOC 3000
*INITREGION 2      ** === Lower Member ===
  *REFDEPTH 3200  ** ft
  *REFPRES 1950   ** psi
  *DWOC 3200
  *DGOC 3100
*INTYPE *KVAR 5*1 8*2
```

---

## **Vertical Equilibrium (Optional)**

**\*REFBLOCK, \*TRANZONE**

**\*VERTICAL, \*REFPRES, \*REFDEPTH,**

**PURPOSE:**

Specify the vertical equilibrium option.

**FORMAT:**

**\*VERTICAL ( \*OFF | \*DEPTH\_AVE )**

**\*REFPRES *ref\_pres***

**\*REFDEPTH *ref\_depth* or \*REFBLOCK *uba***

**\*TRANZONE**

**DEFINITIONS:**

**\*OFF**

Do not perform gravity equilibrium calculation.

**\*DEPTH\_AVE**

Perform depth-averaged capillary-gravity vertical equilibrium calculation in conjunction with \*DWOC and \*DGOC. See EXPLANATION, below.

**\*REFPRES *ref\_pres***

Pressure at reference depth (kPa | psi). *ref\_pres* must lie within the allowed pressure range (see \*MINPRES and \*MAXPRES). This keyword may be specified for each initialization region.

**\*REFDEPTH *ref\_depth***

Reference depth for \*REFPRES (m | ft | cm). This depth must lie within the range of depths contained in the reservoir (a fatal error is issued if not). If the block has both matrix and fracture, then the fracture is indicated. This keyword may be specified for each initialization region.

**\*REFBLOCK *uba***

Address of reference block in UBA format. The depth of this block's center is used as *ref\_depth*. This keyword may be specified for each initialization region.

**\*TRANZONE**

Use gas-water capillary pressure curve(s), specified by subkeyword \*PCGW in the \*SWT table in the Rock-Fluid Data section, to generate a water/gas transition zone in a system that is initially water/gas. In such a zone  $S_g$  is above its critical value and  $S_w$  varies from  $S_{wcon}$  to 1. For a water/gas system it is assumed that  $S_{oirw} = S_{orw} = 0$ . This keyword is available only for water/gas systems, that is, where the same value is specified for the two contact depths \*DWOC and \*DGOC.

## **DEFAULTS:**

If \*VERTICAL is absent, then \*VERTICAL \*OFF is assumed.

If \*TRANZONE is absent, there will be no transition zone for a water/gas reservoir.

## **CONDITIONS:**

\*REFPRES and either \*REFDEPTH or \*REFBLOCK are required with the \*DEPTH\_AVE option. If \*REFDEPTH and \*REFBLOCK appear together for the same initialization region, \*REFBLOCK is ignored.

If \*VERTICAL \*DEPTH\_AVE and \*PRES appear together, \*PRES is ignored.

\*VERTICAL \*DEPTH\_AVE uses information from \*DWOC, \*DGOC and \*WOC\_SW, and will honor information entered via \*SW, \*SO and \*SG.

## **EXPLANATION:**

The capillary-gravity method \*DEPTH\_AVE of calculating vertical equilibrium is a conventional approach that is used also in the other CMG simulators.

There are three fluid system options:

- a) water/oil/gas is flagged when both \*DWOC and \*DGOC appear and give unequal contact depths inside the reservoir.
- b) water/oil is flagged when \*DGOC is absent, or \*DGOC is used to enter a value that is above the top of the reservoir.
- c) water/gas is flagged when both \*DWOC and \*DGOC appear and give equal contact depths inside the reservoir (not available when oil is heavier than water).

In each initialization region, this method executes in two stages:

1. Build Depth Table: Accounting for both gravity and capillarity, it builds a table of phase pressure and capillary pressure versus depth, based on over-all rock-fluid data in the initialization region, phase compositions in one representative block and the comparison of oil and water phase densities at the WOC. Note that densities are based on normalized phase compositions which may vary from reservoir values.
2. Assign Block Conditions: The following is done for each block individually by integration over the block's vertical extent. Oil pressure is assigned from the above table. In the transition zones around the contact depths saturations are obtained from the block's cap pressures curves to maintain gravity/capillary balance. If a cap pressure curve is zero then the transition zone has zero thickness. The saturations outside the transition zones depend on the block's critical saturations (see **Saturation Endpoints**, below). In obtaining these critical saturations first priority is given to the individual block's endpoint data entered via \*BSWR, etc.; otherwise, they come from the block's associated rock type whose endpoints may depend on temperature. In a gas-zone block, the composition is adjusted to satisfy vapour/liquid equilibrium. For a block in a horizontal hybrid or wellbore grid, oil pressure is not integrated but is taken at the block's node (center) depth.

One limitation of this method is that for a relative permeability rock type for which interpolation will be done between multiple sets (\*INTCOMP), only the first set will be used in the depth table generation. Therefore, the first set should correspond to the initial conditions (e.g., no surfactant present).

### Initializing With Non-Endpoint Saturations

The \*DEPTH\_AVE option assigns initial saturations based strictly on the relative permeabilities and capillary pressure curves. Sometimes it is convenient to use the VE initialization but to override the resulting saturations within a limited region with user input. An example would be an oil zone with  $S_w = S_{wc}$  that contains a communication path with  $S_w > S_{wc}$ .

This override may be accomplished with the \*SW, \*SO and \*SG grid array keywords. Overriding is done on a per-block basis, so you may refer to all blocks or to only selected blocks using the reading sub-option \*IJK. Each block referred to with these keywords will have its VE saturation replaced with the user value according to the following three cases.

1. No VE saturations are overridden.
2. One of the VE saturations is overridden. The more mobile of the other two saturations is adjusted to keep the sum of saturations equal to 1. For example, if  $S_w$  is increased above  $S_{wc}$  in the oil zone, then  $S_o$  is adjusted to keep  $S_g = 0$ . All adjustments result in saturations within the range 0 to 1.
3. Two or three VE saturations are overridden. In this case all three VE saturations are replaced with the user's values. The unspecified saturation is obtained by difference.

When any such override occurs the user is notified with a message to both the diary file as well as the main output file.

If  $S_g$  is increased from 0 in the oil or water zone, the mole fractions are adjusted to satisfy vapour-liquid equilibrium as described in "Gas Zone Composition Adjustments" in the EXPLANATION for \*MFRAC\_OIL.

The overriding of VE saturations works well if the user values are not too different from the VE values. For example, it is not possible to create a steam chamber in a cold region just by setting  $S_g > 0$ . Some composition adjustment is generally possible, but extreme variations from the VE saturations may work better if entered as a full set of temperature, saturation and composition.

### Oil Heavier Than Water

Normally the \*DEPTH\_AVE option assumes that oil is lighter than water, so that the phase sequence is, in the most general case, water on the bottom, oil in the middle and gas on the top. However, in some heavy oil reservoirs the oil phase is heavier than the water phase, so the oil zone lies below the water zone. This case is handled by the \*DEPTH\_AVE option and is denoted by "Oil on Bottom" after the fluid system type.

Before its depth table is built, the \*DEPTH\_AVE option determines which liquid phase is on the bottom by comparing the two phase densities at the water-oil contact depth. However, these densities depend upon pressure, so this step is done by integrating pressure gradient from the reference depth to the WOC depth. Since the pressure gradient depends upon which phase is mobile, this is done for two scenarios: oil on top, and water on top. If the two scenarios agree upon which phase is more dense at the WOC, then the heavier phase is assigned to the bottom position for the remainder of the \*DEPTH\_AVE calculation. These oil and water phase densities at the WOC are reported along with the WOC pressure.

If the two scenarios do not agree upon which phase is heavier, then the \*DEPTH\_AVE calculation cannot proceed and the simulation stops. This case happens very rarely, but indicates a situation in which the bottom phase cannot be determined. When it does happen, use a reference depth that is closer to, or equal to, the WOC depth. When the reference and WOC depths are the same, the two scenarios give the same WOC pressure and hence the same phase density comparison.

When there are non-zero water-oil capillary pressures, the size of the water-oil transition zone may become very large when the two phase densities are close together. In addition, the \*DEPTH\_AVE calculation will fail if the phase density difference changes sign in the water-oil transition zone (e.g., water is heavier than oil at the WOC but oil becomes heavier than water somewhere in the water-oil transition zone).

### Phase Composition Adjustments

To avoid a circular calculation dependence with the \*DEPTH\_AVE option, the depth table is generated for a representative but static set of phase compositions. These compositions correspond to the user's input values after passing through the first stage of possible adjustments (see **Stage 1 Adjustments** in section **INITIAL PHASE MOLE FRACTIONS**). If a gas zone is present and the gas mole fractions sum to less than 1, the gas mole fractions are normalized to sum to 1 with no change to the oil mole fractions.

### Entire Oil Column at Saturated Conditions

When the user specifies that the oil phase composition of a black-oil component set is the same at all depths, the resulting pressure distribution may cause grid blocks at some depths to be at the bubble point whereas other blocks are not. This is because K values typically decrease with increasing pressure and the sum of the resulting gas mole fractions (equal to 1 at bubble point) decreases away from 1 going down the oil column.

In order to get the entire oil column at saturated conditions, follow these steps when oil is lighter than water. First, use a reference depth greater than the WOC depth. Second, use a reference pressure that is greater (but not by much) than the maximum pressure expected in the oil column (some trial and error may be needed). Lastly, use \*PBC \*CON 0 to assign the composition of the oil phase at saturated conditions. This combination of data causes each grid block to start with the largest bubble point pressure, then pressure is decreased for blocks above the reference depth, after which the **Stage 2** gas zone adjustment changes oil phase composition from over-saturated to saturated conditions for the local pressure.

### Saturation Endpoints

The following table shows the endpoint saturations that are assigned to various zones. When there are oil components (numx > numw from keyword \*MODEL), the water zone contains 0.0001 oil saturation to enhance stability.

#### Water Heavier Than Oil

Zone	$S_w$	$S_o$	$S_g$
Top (gas)	$S_{wc}$	$S_{o\text{org}}$	$1-S_{wc}-S_{o\text{org}}$
Middle (oil)	$S_{wc}$	$1-S_{wc}$	0
Bottom (water)	1	0	0

## Oil Heavier Than Water

Zone	$S_w$	$S_o$	$S_g$
Top (gas)	$S_{wc}$	$S_{oirg}$	$1-S_{wc}-S_{oirg}$
Middle (water)	$1-S_{oirw}$	$S_{oirw}$	0
Bottom (oil)	$S_{wc}$	$1-S_{wc}$	0

### Rock-Fluid Hysteresis

The capillary-gravity method (\*DEPTH\_AVE) described above uses water-oil and gas-oil capillary pressures. If the hysteresis option is enabled for water-oil capillary pressure, then either the drainage curve (via \*HYS\_DRAINW, the default) or the imbibition curve (via \*HYS\_IMBIBW) can be used. If the hysteresis option is enabled for gas-oil capillary pressure, then either the drainage curve (via \*HYS\_DRAING, the default) or the imbibition curve (via \*HYS\_IMBIBG) can be used.

### \*DEPTH\_AVE for Oil Wet Reservoir (\*OILWET)

When reservoir is oil wet, the inputted drainage capillary pressure are all negative in value (see \*SWT in Rock-Fluid Data section for details). And the \*DWOC (water/oil contact for water wet cases) is then used merely to define the depth of  $p_{cow} = 0$ . To be more precise, this depth is the free water level (FWL) and should be set on the top of the water-oil transition zone.

### \*DEPTH\_AVE for Gas/Water Reservoir

Use \*TRANZONE to specify initial gravity/capillary equilibrium for a two-phase gas/water system. If there is no transition zone, specify the same depth for \*DWOC and \*DGOC. If a gas/water transition zone exists, specify (1) different contact depths and (2) gas/water capillary pressure in the Rock-Fluid Data section via \*SWT subkeyword \*PCGW.

If oil phase may appear later in such a system, account for the expected interfacial forces by specifying oil/water and gas/oil capillary as usual. Also, use keyword \*PC\_3RD\_PHASE in the Rock-Fluid Data section to specify the capillary third-phase correction, to avoid numerical difficulties when oil phase appears and disappears.

### Special Phase Cases

There are some special phase cases that may need additional data. For example, you may need to force the reservoir to be entirely in the water zone if there is no oil-based component (numx = numw from keyword \*MODEL). To do this, explicitly specify via \*DWOC a water-oil contact above the top of the reservoir, perhaps with \*DGOC also at the WOC or higher. The same idea applies to forcing the reservoir to be entirely gas zone.

### Default Contact Depths and Zones

When \*DGOC is absent, GOC is 1 m above the region top. When \*DWOC is absent, WOC is 1 m below the region bottom when water is on the bottom and just below the GOC when oil is on the bottom. Therefore, when both \*DWOC and \*DGOC are absent the region is placed entirely in the oil zone.

## **Insufficient Pressure**

Keyword \*REFPRES specifies pressure at the reference depth. If reference depth is located below the top (minimum depth) of the initialization region, a shooting algorithm estimates lower pressures going up (decreasing depths) until the top is reached. If this algorithm results in a zero or negative pressure, a message like “Negative pressure encountered ...” is issued, basic depth and pressure parameters are reported and the run stops immediately. This condition is considered fatal, requiring a change to input data.

The \*DEPTH\_AVE method assumes that mobile fluids determine the local pressure gradient. It cannot handle non-equilibrium situations, e.g., a liquid column with low pressure at the bottom. A method that always works is to specify pressure at the top of the initialization region, if only to find the resulting pressures for the column below.

## **Overburden and Zero-Porosity Cells**

Sometimes a well-to-surface is modelled by specifying grid cells in the entire overburden above the pay zone. Overburden cells containing the well are active and given zero porosity, and the remaining overburden cells are designated null. Consequently there are no fluid-bearing cells above the pay zone.

Before version 2010.13 the initialization region included zero-porosity cells, so the initialization region included the entire overburden. In this case you could specify \*REFDEPTH at the top of the overburden (surface) and a \*REFPRES of 1 atm. The fluid zone used for the overburden was the same as the top of the pay zone top, effectively extrapolating that zone to the surface. This method often gives an oil-column pressure gradient from the surface to the pay zone, which probably did not match the intended physical situation in the overburden.

Starting with version 2010.13 an initialization region does not include zero-porosity cells, so no initialization region includes the overburden as modelled above. A \*REFDEPTH specified at the top of the overburden is now out of the depth range of the initialization region, which will trigger warning messages and a suggestion to move it to within the depth range. The run continues but assumes that the overburden is a gas zone (gas phase density and no mobile liquid). This behavior matches the other CMG simulators. If \*REFPRES is close to 1 atm then the pay zone pressures will be close to 1 atm as well, which is probably not what is intended. To reproduce the initial pressures of versions before 2010.13, find the pressure value at a representative grid cell and specify that information in the data via \*REFPRES and \*REFBLOCK.

## **Initial Reservoir Pressure and Temperature**

**\*PRES, \*TEMP**

### **PURPOSE:**

Specify initial reservoir pressures and temperatures.

### **ARRAY:**

\*PRES  
\*TEMP

### **DEFINITIONS:**

#### **\*PRES**

Initial oil phase pressure in reservoir (kPa | psi). It must lie within the allowed pressure range (see \*MINPRES and \*MAXPRES).

#### **\*TEMP**

Initial temperature in reservoir (C | F). It must lie within the allowed temperature range (see \*MINTEMP and \*MAXTEMP).

### **DEFAULTS:**

If \*TEMP is absent, then \*TEMP \*CON temr is assumed (see \*TEMR).

### **CONDITIONS:**

\*PRES is required if \*VERTICAL \*DEPTH\_AVE is not used.

If \*VERTICAL \*DEPTH\_AVE and \*PRES appear together, \*PRES is ignored.

---

## Initial Saturations

**\*SW, \*SO, \*SG, \*DWOC, \*DGOC, \*WOC\_SW**

### PURPOSE:

Specify initial saturations in reservoir, or override vertical equilibrium saturations.

### ARRAY:

\*SW  
\*SO  
\*SG

### FORMAT:

\*DWOC *dwoc*  
\*DGOC *dgoct*  
\*WOC\_SW *S<sub>wbwc</sub>*

### DEFINITIONS:

#### \*SW

Initial water saturation in reservoir. Allowed range is 0 to 1.

#### \*SO

Initial oil saturation in reservoir. Allowed range is 0 to 1.

#### \*SG

Initial gas saturation in reservoir. Allowed range is 0 to 1.

#### \*DWOC *dwoc*

Water saturation is initialized in the initialization region according to the water-oil contact depth *dwoc* (m | ft | cm). Use this keyword only if a water zone actually exists within the initialization region.

The exact values assigned depend on the initialization option used. For \*VERTICAL \*DEPTH\_AVE the resulting water saturations will reflect the water/oil transition zone caused by non-zero Pcow. For \*VERTICAL \*OFF the water saturations are assigned as if Pcow = 0. See the EXPLANATION for \*VERTICAL.

For an oil wet reservoir (\*OILWET), *dwoc* (m | ft | cm) defines the depth where the condition  $p_{cow} = 0$  applies.

#### \*DGOC *dgoct*

Gas saturation is initialized in the initialization region according to the gas-oil contact depth *dgoct* (m | ft | cm). Use this keyword only if a gas cap actually exists within the initialization region.

The exact values assigned depend on the initialization option used. For \*VERTICAL \*DEPTH\_AVE the resulting gas saturations will reflect the liquid/gas transition zone caused by non-zero  $P_{cog}$ . For \*VERTICAL \*OFF the gas saturations are assigned as if  $P_{cog} = 0$ . See the EXPLANATION for \*VERTICAL.

#### \*WOC\_SW $S_{wbwc}$

Water saturation below oil-water contact for an initialization region. When using \*WOC\_SW take care to avoid specifying mobile oil in the water zone. In general,  $S_{wbwc}$  should be greater than  $1-S_{orw}$ .

#### DEFAULTS:

If \*DWOC is absent or  $dwoc$  is greater than the depth of the reservoir bottom, then the initialization region will fall in the oil zone; that is, no water zone exists in the region.

If \*DGOC is absent then the GOC, if used, will lie 1 m above the initialization region.

If \*WOC\_SW is absent for a region then  $S_{wbwc} = 0.9999$  is assumed. This value avoids most numerical stability issues associated with an absent liquid phase.

Block saturations are assigned values according to the following priority (1 is highest):

1. Explicit assignment via \*SW, \*SO and \*SG.
2. Depth-dependent assignment via \*DWOC or \*DGOC.
3. The following table.

In the following table 'x' denotes that the saturation is entered. Saturations are normalized where necessary to get  $S_w + S_o + S_g = 1$ .  $S_{wc}$  is the critical water saturation obtained from the block's rock-fluid data which may depend on temperature.

Case	SW	SO	SG	Action	Check for
1	x	x	x		$S_w + S_o + S_g = 1$
2	x	x		$S_g = 1 - S_w - S_o$	$S_g < 0$
3	x		x	$S_o = 1 - S_w - S_g$	$S_o < 0$
4		x	x	$S_w = 1 - S_o - S_g$	$S_w < 0$
5	x			$S_g = 0, S_o = 1 - S_w$	
6		x		$S_g = 0, S_w = 1 - S_o$	
7			x	$S_w = S_{wc}, S_o = 1 - S_w - S_g$	$S_o < 0$
8				$S_w = S_{wc}, S_g = 0, S_o = 1 - S_w$	

#### CONDITIONS:

\*DWOC and \*DGOC are independent of each other, that is, if you use \*DWOC to assign water saturation then you could use \*SG to assign gas saturation. However, it is simpler to be consistent by using \*SW, \*SO and \*SG together, and \*DWOC with \*DGOC.

\*DWOC and \*DGOC refer to depths in the initialization region, and assume that \*DEPTH or \*DTOP has been used to explicitly define these depths. However, depth is defined even when \*DEPTH and \*DTOP are absent, so be sure that the contact depths are consistent with the reservoir depths (a warning message is issued if not).

Note that the saturations assigned via \*DWOC do not depend on relative permeability wettability. You may include or omit any number of blocks in the \*SW, \*SO or \*SG definitions. For example, you may prefer to define \*SW for a watered-out channel only and let the remaining blocks use default  $S_w$ .

If all three saturations are specified for a block, the sum  $S_w + S_o + S_g$  for that block must be significantly greater than zero so that normalization can be done.

If there is no water zone then normally \*DWOC does not need to appear. However, if the \*DEPTH\_AVE option is used to produce a water-oil transition zone from non-zero  $P_{cow}$ , then keyword \*DWOC should be used to specify the bottom of that transition zone.

If there is no gas cap then normally \*DGOC does not need to appear. However, if the \*DEPTH\_AVE option is used to produce an oil-gas transition zone from non-zero  $P_{cog}$ , then keyword \*DGOC should be used to specify the top of that transition zone.

---

## Initial Phase Mole Fractions

\*MFRAC\_WAT, \*MFRAC\_OIL,

\*MFRAC\_GAS, \*PBC

### PURPOSE:

Specify initial mole or mass fractions of the fluid phases.

### ARRAY:

\*MFRAC\_WAT *comp\_name*  
\*MFRAC\_OIL *comp\_name*  
\*MFRAC\_GAS *comp\_name*  
\*PBC *comp\_name*

### DEFINITIONS:

\*MFRAC\_WAT

Initial mole fraction of the indicated component in the water phase; when \*MASSBASIS is in effect it is initial mass fraction. This keyword is needed only to over-ride the very commonly used default case (see DEFAULTS).

\*MFRAC\_OIL

Initial mole fractions of the indicated component in the oil phase; when \*MASSBASIS is in effect it is initial mass fraction.

\*MFRAC\_GAS

Initial mole fractions of the indicated component in the gas phase; when \*MASSBASIS is in effect it is initial mass fraction. This keyword is needed in only two cases: (1) non-condensable components and (2) condensable components based in a liquid phase that is initially absent.

\*PBC

Initial bubble point pressure of the indicated oleic component. For a black-oil type fluid system, this is a convenient alternative to \*MFRAC\_OIL for specifying initial amount of solution gas component in place.

Component i's bubble point pressure  $P_{bi}$  is converted for each block to initial oil mole fraction via

$$X_i = 1 / K_i(P_{bi}, T)$$

where  $K_i(P_{bi}, T)$  is the component's gas-liquid K value evaluated at the initial temperature T and  $P_{bi}$ . For this calculation to succeed  $K_i(P_{bi}, T)$  must be larger than 1 at the indicated conditions. See **Bubble Point Pressure** below.

For each zero value entered for  $P_{bi}$ , the initial block pressure is assigned to  $P_{bi}$ , corresponding to a fully saturated state. The block pressure used in this case may not be the final value obtained by a vertical equilibrium calculation.

#### *comp\_name*

Quoted component name. The component must be found in the phase indicated by the keyword, as determined by \*MODEL and the K value data entered.

#### **DEFAULTS:**

All initial phase mole fractions default to zero, except for the higher level defaults described below. Therefore, these keywords are needed only to assign non-zero values.

A higher level default is available for initial water mole fraction on a per-block basis, but only when there is exactly one aqueous component (see EXPLANATION for \*MODEL). If the water phase exists in a block, every zero value of that aqueous component's water mole fraction is set to 1. This default corresponds to a single water component being the only component in the water phase, the most common case.

A higher level default is available for initial oil mole fraction on a per-block basis, but only when there is exactly one oleic component (see EXPLANATION for \*MODEL) and \*PBC is absent. If the oil phase exists in a block, every zero value of that oleic component's oil mole fraction is set to 1. This default corresponds to a single oil component being the only component in the oil phase.

If \*MFRAC\_OIL is absent and \*PBC is used for each oleic component except one, that one component's initial oil mole fraction is obtained by difference.

#### **CONDITIONS:**

If the water phase exists in at least one block and there are two or more aqueous components, \*MFRAC\_WAT must appear for at least one component. If \*MFRAC\_WAT appears then the water mole fractions must sum to one for each block in which the water phase exists.

If the oil phase exists in at least one block and there are two or more oleic components, \*MFRAC\_OIL must appear for at least one component. If \*MFRAC\_OIL appears then the oil mole fractions must sum to one for each block in which the oil phase exists.

If both liquid phases are absent in at least one block, \*MFRAC\_GAS must appear for at least one component which can partition into a liquid phase. Gas mole fractions must sum to one for each block in which the gas phase is present and liquid phases are not.

\*PBC is available only for oleic components, that is, components based in the oil phase. Also, \*PBC is not available when \*MASSBASIS is in effect. In addition, the \*PBC value for any block may not exceed the total fluid pressure for that block.

Note that some restrictions apply to K values when \*PBC is used. See "Vapour Pressure" in the EXPLANATION for keyword \*KVTABLE.

Both \*PBC and \*MFRAC\_OIL should not be specified for the same component.

#### **EXPLANATION:**

These are the tasks performed to obtain a complete set of consistent initial phase mole fractions:

1. Apply low-level defaults (all set to zero)
2. Read keyword data
3. Perform fine-grid inheritance
4. Apply high-level defaults
5. Obtain derived mole fractions from basic values
6. Make phase consistency adjustments in two stages

In the following,  $w_i$ ,  $x_i$  and  $y_i$  are water, oil and gas mole fractions, respectively, of component  $i$ , and  $K_i$  and  $K_{Li}$  are the corresponding gas-liquid and liquid-liquid K values, respectively. See EXPLANATION for \*MODEL for definitions of *aqueous* and *oleic*.

### **Basic and Derived Mole Fractions**

Phase mole fractions are related by K values and so are not independent of each other. Some initial mole fractions are considered *basic* while others are *derived* from K values and the basic mole fractions. Only basic initial mole fractions need be entered via keyword. Input data that corresponds to derived initial mole fractions will be ignored. The following table shows which initial mole fractions are basic.

Aqueous components:

$S_w > 0$ :	Basic: $w_i$
	Derived: $y_i = K_i \cdot w_i$ , $x_i = K_{Li} \cdot w_i$
$S_w = 0$ :	Basic: $y_i$
	Derived: $w_i = y_i / K_i$ , $x_i = K_{Li} \cdot w_i$

Oleic components:

$S_o > 0$ :	Basic: $x_i$
	Derived: $y_i = K_i \cdot x_i$ , $w_i = K_{Li} \cdot x_i$
$S_o = 0$ :	Basic: $y_i$
	Derived: $x_i = y_i / K_i$ , $w_i = K_{Li} \cdot x_i$

Non-condensable gases:

$S_g > 0$ :	Basic: $y_i$
-------------	--------------

### **Stage 1 Adjustments**

The first adjustment stage involves the liquid phase constraints. When the water phase is present, the largest  $w_i$  of the aqueous components is adjusted to satisfy

$$\sum w_i = 1.$$

When the oil phase is present, the largest  $x_i$  of the oleic components is adjusted to satisfy

$$\sum x_i = 1.$$

When both liquid phases are present the adjustment accounts for liquid/liquid solubility.

When liquid/liquid solubility occurs (at least one  $K_{Li} > 0$ ) there will be some adjustment of liquid phase mole fractions, despite the fact that the input mole fraction data sums to 1.

Be careful with situations where an initial phase property is sensitive to the relative values of the phase mole fractions. One of those mole fractions may be adjusted, thus changing significantly the relative mole fractions of the phase and hence its property.

When the gas phase exists and at least one non-condensable gas component has an initial non-zero  $y_i$ , the largest  $y_i$  of the non-condensable components is adjusted to satisfy

$$\sum y_i = 1.$$

### Stage 2 Adjustments (Gas Zone)

Usually all of the gas mole fractions of condensable (aqueous and oleic) components are derived from other mole fractions and K values. In addition, K values depend upon initial pressures which may vary continuously with depth. As a result, it is often not practical to enter initial liquid mole fractions that satisfy liquid-vapour and phase constraints in a gas zone. The second stage of initial mole fraction adjustments takes care of this problem.

The adjustments in Stage 2 are different in type from those in the Stage 1. In Stage 2 components are divided into three groups: (1) oleic components with  $K_i > 1$ , (2) all other oleic components and (3) aqueous components. Mole fractions of all components in each group are adjusted by the same ratio to satisfy

$$\sum y_i = 1 \text{ along with } \sum x_i = 1 \text{ if oil phase exist and } \sum w_i = 1$$

if water phase exists. For example, group 1 mole fractions may be increased by the factor 1.8 while group 2 mole fractions may be decreased by the factor 0.6. This strategy preserves the character of in-place gas and oil partitions represented by several components, and works also for the basic black-oil case (dead oil and solution gas). As with Stage 1, Stage 2 adjustments account for any liquid/liquid solubility.

Generally, you may enter the composition of undersaturated oil

$$\sum y_i < 1,$$

and the  $x_i$  will be adjusted automatically from undersaturated to saturated conditions

$$(\sum y_i = 1)$$

in the gas zone. In addition, when

$$\sum y_i > 1$$

the same kind of adjustment is done so that

$$\sum y_i = 1,$$

even when  $S_g = 0$ . When there is no oil phase in the gas zone, a similar adjustment is made to  $y_i$  alone.

Stage 2 adjustments are possible only if all of the following are true:

1.  $y_i=0$  for all non-condensable components
2. Group 1  $\sum x_i$  (e.g., solution gas) is non-zero
3. Group 2  $\sum x_i$  (e.g., dead oil) is non-zero
4.  $K_i$  do not change due to composition dependence during adjustment, for example, both old and new key compositions are below  $x_{low}$  (see \*KVKEYCOMP).

### Bubble Point Pressure

The system bubble point, or saturation, pressure is the pressure at which gas phase begins to appear in a multi-component system at a given temperature and composition. Note that this definition applies to the multi-component system and not to any component in particular.

However, keyword \*PBC refers to a specified component because it caters to the black-oil type fluid system. In such a system the bubble point pressure depends only on the amount of solution gas component dissolved in the oil phase and so determines completely the composition of the oil phase. The quantity entered via keyword \*PBC is actually the bubble point pressure of an equivalent black-oil type system in which the named component is the solution gas.

The relationship between the bubble point pressure  $P_{bi}$  and oil mole fraction  $x_i$  of component i is  $x_i \cdot K_i(P_{bi}, T) = 1$  where  $K_i(P_{bi}, T)$  is the component's K value at  $P_{bi}$  and temperature T. In the more general STARS component treatment, this corresponds to the hypothetical situation where the gas phase consists entirely of that component, since component gas mole fraction  $y_i = x_i \cdot K_i = 1$ .

This relationship between  $x_i$  and  $P_{bi}$  can be extended to any component in general, since it depends only on the K value. With respect to initial composition, \*PBC may be used for any component that satisfies that keyword's conditions, but usually it is practical for only one component. Note that for the \*PBC default (one component that does not use \*PBC gets mole fraction from difference) to work, all other oleic components must appear after \*PBC, even a component whose initial mole fraction is intended to be zero.

Because \*PBC data  $P_{bi} = 0$  indicates saturated conditions, a desired bubble point pressure value of zero must be approximated by a small positive value, for example, 1e-10, or \*MFRAC\_OIL can be used instead.

With respect to output, a bubble point pressure is an alternate way to report oil phase composition.

### Obsolete Mole Fraction Defaults

Previous versions of STARS applied the higher level water and oil mole fraction defaults described in DEFAULTS to cases with more than one aqueous and oleic component, respectively. This practice is not recommended. STARS 2003 flags this condition with a warning and performs the default as before, but future versions may disallow it.

---

## Initial Solid Concentration

\*CONC\_SLD

### PURPOSE:

Specify initial solid concentration.

### ARRAY:

\*CONC\_SLD *comp\_name*

### DEFINITIONS:

\*CONC\_SLD

Initial concentration (gmol/m<sup>3</sup> | lbmol/ft<sup>3</sup> | gmol/cm<sup>3</sup>) of the indicated component in the solid phase.

*comp\_name*

Quoted component name. The component must be a solid component, as determined by \*MODEL.

### DEFAULTS:

For each solid component, if \*CONC\_SLD is absent then that component's initial concentration is zero.

### EXPLANATION:

Keyword \*CONC\_SLD is a single-component variation of the standard grid array input data type.

#### Example:

If fuel deposition is 2 lb per cubic foot, porosity is 30%, and coke molecular mass is 13, then

$$\begin{aligned} \text{Ccoke} &= (2 \text{ lb fuel} / \text{one cubic foot reservoir volume}) \\ &\quad * (1 \text{ lb mole fuel} / 13 \text{ lb fuel}) \\ &\quad * (1 \text{ reservoir volume} / 0.3 \text{ pore volume}) \\ &= 0.513 \text{ lb mole fuel} / \text{pore volume} \end{aligned}$$

Keyword data for this case would be

```
*COMPNAME . . . 'Coke'  
*CONC_SLD 'Coke' *CON 0.513
```

---

## Datum Depth Specification (Optional)

\*DATUMDEPTH

### PURPOSE:

Specify datum depth for output of pressure corrected to a datum.

### FORMAT:

```
*DATUMDEPTH Zdatum (*INITIAL  
| *REFDENSITY ρdatum  
| *REFDENSITY *GRIDBLOCK )
```

### DEFINITIONS:

Z<sub>datum</sub>

A real number denoting the depth to the datum. (m | ft | cm)

#### \*INITIAL

The initial equilibrium pressure distribution in the reservoir will be used to calculate the corrected datum pressures.

#### \*REFDENSITY ρ<sub>datum</sub>

Specified mass density ρ<sub>datum</sub> (kg/m<sup>3</sup> | lb/ft<sup>3</sup> | kg/cm<sup>3</sup>) will be used to correct the pressures for the gravity head.

#### \*REFDENSITY \*GRIDBLOCK

The grid block fluid mass density will be used to correct the pressures for the gravity head.

### DEFAULTS:

If \*DATUMDEPTH is absent, datum pressures are not available.

If \*DATUMDEPTH is present without \*INITIAL or \*REFDENSITY, \*INITIAL is assumed.

If \*REFDENSITY is specified without ρ<sub>datum</sub> or \*GRIDBLOCK, \*GRIDBLOCK is assumed.

### CONDITIONS:

Datum pressures \*DATUMPRES are available for output only if \*DATUMDEPTH is present.

### EXPLANATION:

\*DATUMDEPTH modifies the individual grid block pressures to a common depth by removing the gravity head. Removal of the gravity head can be accomplished with one of three methods:

1. If \*INITIAL (the default) is used, pressures are corrected based on the initial equilibrium pressure distribution of the reservoir. For each block (i)

$$p_{\text{datum}}(i) = p(i) - p_0(i) + p_{0,\text{datum}}$$

where

- $p_{\text{datum}}(i)$  is the corrected grid block datum pressure,  
 $p(i)$  is the actual grid block pressure,  
 $p_0(i)$  is the grid block pressure at initial time (time = 0), and  
 $p_{0,\text{datum}}$  is the datum pressure at initial time.

Note: The above calculation is a good approximation during the simulation only when:

- a) The phase densities do not change by a large amount, and
  - b) The water-oil and gas-oil contacts do not move by a large amount.
2. If \*REFDENSITY  $\rho_{\text{datum}}$  is used, pressures are corrected based on the fluid head of a specified density. For each block (i)

$$p_{\text{datum}}(i) = p(i) + \rho_{\text{datum}} \cdot g \cdot (Z_{\text{datum}} - Z_i)$$

where

- $\rho_{\text{datum}}$  is specified after \*REFDENSITY,  
 $g$  is the acceleration due to gravity,  
 $Z_{\text{datum}}$  is specified after \*DATUMDEPTH, and  
 $Z_i$  is the depth of the grid block.

3. If \*REFDENSITY \*GRIDBLOCK is used, pressures are corrected based on the fluid head of a local density. For each block (i)

$$p_{\text{datum}}(i) = p(i) + \rho(i) \cdot g \cdot (Z_{\text{datum}} - Z_i)$$

where

- $\rho(i)$  is grid block fluid mass density (oil phase if present; otherwise, gas phase if present; otherwise, water phase).

Examples:

```
*DATUMDEPTH 1000 *INITIAL
*DATUMDEPTH 1000 ** Same as previous line

*DATUMDEPTH 850 *REFDENSITY *GRIDBLOCK
*DATUMDEPTH 850 *REFDENSITY ** Same as previous line

*DATUMDEPTH 2300 *REFDENSITY 56 ** density in lb/ft3
```

### Output of Datum Pressures

Datum pressures are available for per-grid block output via subkeyword \*DATUMPRES in the **PRN\_GRID** list describe with \*OUTPRN \*GRID. This subkeyword may be used by \*OUTSRF \*GRID as well as \*OUTSRF \*SPECIAL \*BLOCKVAR. If \*DATUMDEPTH is absent, subkeyword \*DATUMPRES is not available.

Datum pressure is also reported for each well. If \*DATUMDEPTH is absent, the well datum pressure reported is the raw pressure.

If \*OUTPRN \*WELL is present, well reports in the “.out” file include values of datum pressure for each well. If \*DATUMDEPTH is absent, these reported values represent the average of unmodified grid block pressure (no removal of gravity head) in the well completion layers, weighted by mobilities of the fluids in those layers.

---

## Initial Conditions from Restart

**\*INIT\_FROM\_RESTART**

### PURPOSE:

Specify that initial conditions are to come from a restart record.

### ARRAY:

**\*INIT\_FROM\_RESTART**

### DEFINITIONS:

**\*INIT\_FROM\_RESTART**

Specify that initial conditions of a non-restart run are to come from a restart record.

### DEFAULTS:

If **\*INIT\_FROM\_RESTART** is absent, all initial conditions for a non-restart run are obtained from the **\*INITIAL** data section.

### CONDITIONS:

Keyword **\*INIT\_FROM\_RESTART** is available only for non-restart runs.

This feature is not available together with these features:

1. **\*AQUIFER**
2. Dynamic or recurrent grid changes.

The two runs must have the same thermal/isothermal status.

The two runs must have the same number of cells or the same grid.

### EXPLANATION:

With **\*INIT\_FROM\_RESTART** you are able to copy a non-restart run's initial conditions from the restart record of another run with the same grid (or the same number of cells). This feature is useful when a restart cannot be done directly, such as when the second run is a continuation of the first run but with an expanded component set.

Well and aquifer accumulations are not copied and so will be initialized to zero.

The number of cells in the second run must match exactly the number of cells in the first run, which normally implies that the grid definition is the same.

Per-cell material-in-place of pre-existing components is obtained from the restart record. This assumes that the PVT properties of pre-existing components are unchanged. Any change to a property that determines material-in-place (e.g., density) results in a material balance error that cannot be reconciled and may cause the run to abort. This material-balance restriction makes it necessary to use the same grid specification for the two runs. It also requires the same component PVT properties, especially K values and density.

It is possible to obtain some initial conditions from an IMEX SR2 file set. See keyword **\*INIT\_FROM\_IMEX**.

## **Changing Component List**

Component-based quantities are obtained on a component-name basis, that is, the information for a component will be transferred only if the same component name is used in both runs. The components in the second run can be ordered differently via keyword \*COMPNAME.

New components can be inserted in any position consistent with the component types described for keyword \*MODEL and \*COMPNAME. For each new component, phase mole fractions, solid concentration, adsorption and material in place are initialized to zero.

## **Simulation Time**

If you plan to plot histories of the first and second runs together on the same time axis, use \*DATE instead of \*TIME immediately after \*RUN in both data sets. This allows Results Graph to use a common date axis even though the two runs start at different dates. You may use either \*TIME or \*DATE at other reference times, but use of \*DATE throughout is more easily managed in practical cases.

## **Recurrent Data**

For recurrent features (e.g., wells, transmissibility multipliers) that are used during both runs, data can be transferred almost as-is from the first run. When transferring such data, ensure that the specification at the beginning of the second run represents the specification at the restart time of the first run, which may require including modifications at intermediate times.

When transferring well data from the first run you may find that well numbers are inconsistent between the two data sets. It is recommended that you use well names exclusively by removing well numbers from the \*WELL cards.

Using a well of the same name in both runs is convenient and allowed, but in the second run that well's accumulation starts at zero since well accumulations are not transferred.

Note that \*INCOMP depends upon the component list, so \*INCOMP may need to be changed if it is copied from the first run.

## **Why Initialize from Restart?**

An enhanced oil recovery method often is applied after another recovery method, such as primary recovery, water flood or steam flood. The enhanced method also may involve additional chemical components, such as solvent, surfactant or air. A common simulation scenario is this:

- |         |  |
|---------|--|
| Run #1: | Simulate first recovery method with minimal component set.   |
| Run #2: | Rerun first recovery method with expanded component set (but new components not present or injected), hoping the result is the same as (or at least close to) result #1. |
| Run #3: | Simulate the enhanced recovery method by restarting from run #2.   |

The usual problem with this scenario is that the result of run #2 cannot be made close to that of run #1. Run #2 may be just run #1 with “absent” components, but a number of numerical issues can cause the results of run #2 to deviate from those of run #1. In many cases the results can be made close only when the convergence tolerances of both runs are tightened.

The consequence may well be that the result of tight-tolerance run #1 may be significantly different from the original run #1 result. This is particularly frustrating when the original run #1 results have been adopted as the reference from which decisions were made. Another issue with run #2 is that it will consume more resources (memory and CPU) than run #1 due at the very least to increased equations per cell. In addition, run #2 needs to be redone for each different type of enhanced recovery method (Run #3).

The value of the ability to initialize with another run's restart and an expanded component list is to eliminate run #2, that is, initialize run #3 directly from the restart of run #1.

### Verifying Expanded Component List

Once the second data set is constructed, you should test the correctness of the component PVT and recurrent data. The best test is to compare the results of a few timesteps for the following two runs:

1. the first data set, restarted from the original restart record (with additional \*DATE/\*TIME card if restart time is the end of the original run).
2. the second data set (with \*INIT\_FROM\_RESTART), with no data that introduces new components (e.g., injection, reactions).

These two test runs should give the same result except that the second run has “unused” components. This comparison can be carried out in detail by enabling \*OUTPRN \*GRID \*ALL and \*OUTPRN \*WELL \*ALL (or \*WELLCOMP) and comparing the corresponding “.out” files. The results of these two runs should be identical for the pre-existing components, except for well accumulations (which \*INIT\_FROM\_RESTART does not carry forward from the restart record) and output ordering.

The comments in “WHAT CAN BE CHANGED AT A RESTART” in the Tutorial chapter apply to \*INIT\_FROM\_RESTART with respect to pre-existing components.

### Example

Templates “stflu024.dat” and “stflu025.dat” provide an example of continuing a simulation with an expanded component set. Case “stflu024.dat” is a steam pre-heat phase, followed by combustion in “stflu025.dat”. In fact, “stflu025.dat” contains both the steam and combustion phases and so duplicates the steam-phase simulation with the “unused” combustion components in the component list.

*COMPNAME	stflu024	stflu025
'water'	x	x
'bitumn'	x	x
'ch4'	x	x
'co2'		x
'co/n2'		x
'oxygen'		x
'coke'	x	x

In “stflu025” the component set is expanded to include combustion components. Unusually, the steam phase includes a coke component to model pre-combustion fuel laydown.

In theory the steam-phase results should be identical for the two templates, but practical numerical behavior prevents this. These two templates can give very nearly the same steam-phase results only if the convergence criteria are tightened significantly (and run times increased by 3-4 times). This technique proves the concept but is not practical for normal use where individual runs can take several days each and more complex runs could require even more significant tightening of convergence criteria. Therefore, the more practical method is to start the combustion run (stflu025) with initial conditions from the end of the steam phase (stflu024).

---

## Initial Conditions from IMEX Run

\*INIT\_FROM\_IMEX

### PURPOSE:

Specify that initial conditions are to come from the SR2 files generated by an IMEX run.

### FORMAT:

\*INIT\_FROM\_IMEX ( \*SGAS ( *sgas\_name* ) )

### DEFINITIONS:

#### \*INIT\_FROM\_IMEX

Specify that some initial conditions are to come from the SR2 files generated by an IMEX run. The IMEX SR2 files are specified via \*FILENAME \*INDEX-IN (or command argument “-r”). The particular time record in that file is specified via \*RESTART (or command argument “-restart”) or \*RESTIME (or command argument “-restime”).

#### \*SGAS ( *sgas\_name* )

Optional subkeyword \*SGAS causes the IMEX bubble-point pressure to be assigned to one of the STARS components. If \*SGAS is absent, no attempt is made to read bubble-point pressure from the IMEX SR2 files, in which case keyword \*MFRAC\_OIL or \*PBC may assign solution gas content.

Quoted character string *sgas\_name* is optional after \*SGAS and indicates which STARS component is initialized with the IMEX bubble-point pressure. *sgas\_name* must appear in the \*COMPNAME list. This component must be oleic (based in oil phase) and appear also in the gas phase (have non-zero K value).

### DEFAULTS:

If \*INIT\_FROM\_IMEX is absent, no initial conditions are obtained from an IMEX SR2 file.

If \*INIT\_FROM\_IMEX is present but \*SGAS is absent, no attempt is made to read bubble-point pressure from the IMEX SR2 files.

If \*INIT\_FROM\_IMEX \*SGAS is present but *sgas\_name* is absent, the IMEX bubble-point pressure is assigned to the first oleic component in the \*COMPNAME list that exists also in the gas phase. If no such component is found, execution proceeds as if \*SGAS were absent.

### CONDITIONS:

Keyword \*INIT\_FROM\_IMEX is available only for non-restart runs and may appear at most once in a data set.

The IMEX and STARS runs must have the same number of active cells or the same grid.

The \*SGAS option of \*INIT\_FROM\_IMEX may not be used together with \*MASSBASIS.

When \*INIT\_FROM\_IMEX is used, the solution gas component must be initialized at most one way: with \*SGAS, with \*MFRAC\_OIL or with \*PBC.

## **EXPLANATION:**

With \*INIT\_FROM\_IMEX you are able to obtain some initial conditions for a STARS run from an SR2 file set generated by an IMEX run. The number of active cells in the two runs must match exactly, which normally implies that the grid definition is the same.

The following quantities can be obtained from an IMEX run:

Quantity	IMEX *OUTSRF *GRID	STARS Initial Condition Data
Fluid Pressure	*PRES	*PRES
Oil Saturation	*SO	*SO
Water Saturation	*SW	*SW
Bubble-point Pressure	*BPP	*PBC <i>sgas_name</i>

The single SR2 time record of interest is specified by keyword \*RESTART, \*RESTIME or the corresponding command-line arguments, and only that time record is searched for these quantities. All these quantities are available at an IMEX restart time. If that time has no restart record, a quantity is available if a grid dump had been requested for that time via \*WRST \*GRID and the corresponding \*OUTSRF \*GRID subkeyword.

If a quantity is not found, normal STARS defaulting rules apply. Fluid pressure has no default initial condition, so it must be specified via STARS Initial Condition keyword \*PRES if it cannot be obtained from the IMEX SR2 file.

No matter where a quantity comes from, the usual normalizations and adjustments are applied (see EXPLANATION sections of keywords \*SW and \*PBC). Do not use Initial Condition keyword \*SG; this allows gas saturation to come from the usual defaulting rules.

You must specify initial conditions for other quantities such as \*TEMP for temperature and \*MFRAC\_OIL for oleic components other than *sgas\_name*.

## **Simulation Time**

The \*RESTART or \*RESTIME time is used only to specify the time record of interest in the IMEX SR2 files. The STARS run starts at 0 days.

## **Verifying Conversion from IMEX**

You should test the correctness of the conversion of the IMEX conditions to STARS by comparing the STARS initial material-in-place with the IMEX material-in-place at the reading time. Take care to account for differences in formation porosity models as well as fluid compressibility and vapourization models. Typically the STARS property data would be constructed to match the IMEX properties, but differences in model types may well result in properties at initial conditions that are only close.

## **Notification**

The .out file displays what quantities were obtained from the IMEX SR2 files, in the Restart Messages area:

```

MESSAGES FOR RESTART FILE READING:
Reading initial conditions from IMEX restart:
- pressure
- water saturation
- oil saturation
- bubble-point pressure

```

If a quantity is absent from this display list, it was not obtained from the IMEX SR2.

### **Example**

Template “stsмо058.dat” obtains initial conditions from the IMEX run “mxdrm001.dat”. This STARS data has the following component set and initialization data.

```

*MODEL 4 4 4
*COMPNAME 'Water' 'Hevy Oil' 'Lite Oil' 'Soln Gas'
**           C20          C10          C1
...
*INITIAL
*TEMP *CON 93 ** deg F
*MFRAC_OIL 'Hevy Oil' *CON 0.25
*MFRAC_OIL 'Lite Oil' *CON 0.75
*INIT_FROM_IMEX *SGAS 'Soln Gas'

```

Both components 'Lite Oil' and 'Soln Gas' appear in both the gas and oil phases. If *sgas\_name* did not appear, the bubble-point pressure would be assigned to 'Lite Oil'. The presence of *sgas\_name* forces the assignment to component 'Soln Gas'.

The liquid oil (corresponding to “dead oil” in IMEX) is represented by two separate components, 'Lite Oil' and 'Hevy Oil', and their relative composition is specified by \*MFRAC\_OIL. Composition normalization accommodates the 'Soln Gas' bubble-point pressure from IMEX while the relative composition of the STARS liquid-oil is unaltered.

For example, the \*MFRAC\_OIL data above indicates that the mole fraction of 'Lite Oil' is three times that of 'Hevy Oil', constituting the composition of the “dead” oil phase. If the IMEX bubble-point pressure is converted into a 'Soln Gas' oil mole fraction of 0.4, then the 'Lite Oil' and 'Hevy Oil' oil mole fractions are adjusted to 0.45 and 0.15, respectively. This preserves the 3-to-1 mole ratio of 'Lite Oil' to 'Hevy Oil' while satisfying the condition that oil mole fractions sum to 1. This adjustment is done in each cell that contains gas phase.



# Numerical Methods Control

---

## Summary of Numerical Methods Control

Define parameters that control the simulator's numerical activities such as timestepping, iterative solution of non-linear flow equations, and the solution of resulting system of linear equations.

### List of Options

The equation/variable formulation has the following options:

- thermal or isothermal
- saturation/mole fraction/temperature, global mole/temperature, or global mole/enthalpy formulations

The matrix solver has the following options:

- four choices for block ordering (e.g., natural or red-black)
- direct (Gauss) or iterative incomplete LU (ILU) methods
- choice of degree (fill) for ILU method

Flow equation building has the following options:

- stability-based or threshold adaptive-implicit method
- option to allow or disallow backflow in well layers
- control over level (N or K) for the upstream of some flow quantities
- control over timestep sizes
- control over iteration convergence tolerances

### Required Data

There are no required or mandatory keywords in this section. Each keyword has a default value which can be used. If no keywords from this section are used, then the \*NUMERICAL keyword may be omitted.

### Critical Keyword Ordering

There is no critical keyword ordering.

## Use of Defaults

The defaults used in the numerical solution techniques provide a robust and efficient solution to most simulation problems. You should override the defaults only if you have a good understanding of the solution methods involved. Inappropriate overriding of the defaults may result in the use of much more CPU time than would otherwise be required for a problem.

For detailed explanations of the matrix solution controlling keywords (\*NORTH, \*SORDER, \*PIVOT, \*SDEGREE, \*PRECC, \*ITERMAX), please refer to section “Improving Numerical Performance” in the Tutorial chapter.

## Usage in Other Sections

The following keywords in this section may be used also in the Well and Recurrent Data section:

*MAXSTEPS	*DTMAX	*DTMIN
*NUMSET	*NORM	*CONVERGE
*MATBALTOL	*NEWTONCYC	*UNRELAX
*UPSTREAM	*PRECC	*NORTH
*PIVOT	*ITERMAX	*AIM
*PVTOSCMAX	*NCUTS	

## Changing Numerical Controls at Restart

Except for \*TFORM and \*ISOTHERMAL, any of the numerical control data can be changed at a restart. This includes changing the linear equation solver choice (AIMSOL versus PARASOL) as well as individual solver operation parameters. In general, numerical control data (e.g., \*SDEGREE) that can be changed at a restart but not in recurrent data has an effect on storage allocation done at the beginning of the run.

When you change or add keywords to the Numerical Control section at a restart, be careful that the new data is not overwritten by a pre-existing keyword in the Recurrent Data section. For example, suppose a data set has \*DTMAX 10 in Numerical Control and \*DTMAX 3 in Recurrent Data at 40 days. If you wish to restart from 40 days but with \*DTMAX changed to 5, then it would be necessary to change the \*DTMAX value specified at 40 days from 3 to 5; changing the \*DTMAX value in Numerical Control would not work.

## Equations and Their Solution

Appendix F contains detailed discussions of construction of equations, as well as methods for their solution.

---

## **Numerical Methods Control Identifier (Optional)**

**\*NUMERICAL**

### **PURPOSE:**

\*NUMERICAL identifies the beginning of all numerical methods control keywords.

### **FORMAT:**

**\*NUMERICAL**

### **DEFAULTS:**

All the keywords in this section are optional, and therefore the entire section is optional.

### **CONDITIONS:**

The NUMERICAL METHODS CONTROL keyword group follows the INITIAL CONDITIONS keyword group in the data file.

---

## Maximum Timestep Number (Optional)

\*MAXSTEPS

### PURPOSE:

Specify the maximum allowed timestep number.

### FORMAT:

\*MAXSTEPS *nstop*

### DEFINITIONS:

*nstop*

Timestep number at which the simulation is stopped, unless it stops before that from other causes such as \*STOP. Integer *nstop* must be non-negative. The command line argument '-maxsteps *nstop*' is the same as \*MAXSTEPS *nstop*. The command line argument '-onestep' is the same as \*MAXSTEPS 1.

### DEFAULTS:

\*MAXSTEPS 9999

### CONDITIONS:

This keyword may be located also in the Well and Recurrent Data section.

### EXPLANATION:

See **How To Do a Restart** in the Tutorial section.

\*MAXSTEPS is useful when you are very unsure how much time a new data file will take to run. It is also useful for running your data for 1 timestep for debugging purposes.

### Example:

```
** Stop the run at timestep #100
*MAXSTEPS 100
```

---

## Maximum, Minimum Timestep Size (Optional)

\*DTMAX, \*DTMIN

### PURPOSE:

Specify the maximum and minimum allowed time-step sizes.

### FORMAT:

\*DTMAX *max\_time\_size*  
\*DTMIN *min\_time\_size*

### DEFINITIONS:

*max\_time\_size*

The maximum time-step size allowed (day | day | min). The allowed range is *min\_time\_size* to  $10^{20}$  days.

*min\_time\_size*

The minimum time-step size allowed (day | day | min). The allowed range is  $10^{-20}$  days to *max\_time\_size*.

### DEFAULTS:

\*DTMAX  $10^{35}$  (days)  
\*DTMIN  $10^{-8}$  (days)

### CONDITIONS:

These keywords may be located also in the Well and Recurrent Data section.

The condition *min\_time\_size* < *max\_time\_size* is always enforced.

### 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.

Examples:

```
** Limit the maximum time-step size to half a day
*DTMAX 0.5
```

---

## Model Formulation (Optional)

\*TFORM, \*ISOTHERMAL

### PURPOSE:

Indicate formulation options of numerical model.

### FORMAT:

\*TFORM ( \*SXY | \*ZH | \*ZT )  
\*ISOTHERMAL

### DEFINITIONS:

#### \*SXY

Standard use of phase saturations, phase mole fractions, and temperature as primary variables. Phase appearance or disappearance causes variable switching.

#### \*ZH

Isenthalpic flash algorithm. Energy flow equation is aligned with system fluid enthalpy  $H_{sys}$  as primary variable and temperature is viewed as a function of pressure, global composition and  $H_{sys}$ .

#### \*ZT

Temperature flash algorithm. Energy flow equation is aligned directly with temperature. This approach is appropriate to thermal flow problems with wide-boiling systems (e.g., hot/cold water problems with no steam appearance, or combustion problems with large concentrations of non-condensable gases). This approach is equivalent to the flash associated with compositional simulation.

#### \*ISOTHERMAL

Energy conservation is not imposed and grid cell temperatures will not change from their specified initial values. This option is available only with \*TFORM \*ZT.

Input temperatures are used to evaluate various quantities such as fluid properties (densities, viscosities), gas-liquid phase equilibrium and reaction rates. The principle input temperatures are:

- \*TEMP Initial grid cell temperatures (possibly non-uniform)
- \*TEMR Reference temperature
- \*TINJW Temperature of injected fluids

### DEFAULTS:

\*TFORM \*SXY

### CONDITIONS:

The \*TFORM option and \*ISOTHERMAL setting may not be changed at a restart. This implies that a thermal run may not be restarted from an isothermal run.

---

## Numerical Set

\*NUMSET, \*NUMTYPE

### PURPOSE:

Define and assign multiple numerical sets.

### FORMAT:

\*NUMSET *key*

### ARRAY:

\*NUMTYPE

### DEFINITIONS:

*key*

Numerical set number, starting from 1 and increasing by 1 for each different numerical set. \*CONVERGE and \*NORM data will be assigned to the numerical set number that is in force as keywords are read in sequence.

\*NUMTYPE

Enter a numerical set *key* for each grid block. Only 1 and *key* values that have been defined are allowed.

### DEFAULTS:

The default numerical set *key* is 1. \*NUMSET is needed only to define multiple numerical sets.

The default *key* assigned to each block is 1. \*NUMTYPE is needed only to assign multiple numerical sets to the grid.

Unless you have multiple numerical sets, you do not need \*NUMSET or \*NUMTYPE.

When \*CONVERGE and/or \*NORM is used in the Well and Recurrent Data Section without \*NUMSET, those values are applied to the numerical set specified by the last appearance of keyword \*NUMSET.

### CONDITIONS:

Keyword \*NUMSET may be located also in the Well and Recurrent Data section.

### EXPLANATION:

Different Numerical Sets may be needed when different parts of a reservoir need different \*CONVERGE and/or \*NORM values. This may be useful in heterogeneous reservoirs, fractured reservoirs or when Discretized Well Model is used. For example fracture \*NORM and/or \*CONVERGE may be different from matrix values. Note that \*CONVERGE \*TOTRES applies to the grid as a whole and so cannot be assigned different values in different regions.

**Example:**

```
*NUMSET 1
*CONVERGE *PRESS 2.0  *TEMP  3.0  *SATUR  0.05
*NORM  *PRESS 400.0   *SATUR  0.3

*NUMSET 2
*CONVERGE *PRESS 1.0  *TEMP  3.0  *SATUR  0.05
*NORM  *PRESS 200.0   *SATUR  0.35

*NUMTYPE *FRACTURE *CON 2
*NUMTYPE *MATRIX   *CON 1
```

---

## **Normal Variation in Variables per Timestep (Optional)      \*NORM**

### **PURPOSE:**

\*NORM identifies the typical changes in the basic variables over a timestep.

### **FORMAT:**

```
*NORM { *PRESS x | *SATUR x | *TEMP x | *Y x  
| *X x | *W x | *FLUIDH x  
| *ZO x | *ZNCG x | *ZAQ x }
```

### **DEFINITIONS:**

#### **\*PRESS *x***

Normal pressure change (kPa | psi). The suggested range is 100 kPa to 2000 kPa.

#### **\*SATUR *x***

Normal saturation change. The suggested range is 0.05 to 0.4.

#### **\*TEMP *x***

Normal temperature change (C deg | F deg). The suggested range is 5 C deg to 150 C deg. Note that this quantity is a temperature difference, and so conversion between C (or K) and F (or R) involves only the factor 1.8.

#### **\*Y *x***

Normal gas mole fraction change. The suggested range is 0.05 to 4.

#### **\*X *x***

Normal oil mole fraction change. The suggested range is 0.05 to 4.

#### **\*W *x***

Normal water mole fraction change. The suggested range is 0.05 to 4.

#### **\*FLUIDH *x***

Normal fluid enthalpy change (J/gmol | Btu/lbmole). The suggested range is 0 to 1e9.

#### **\*ZO *x***

Normal change in oleic component global mole fraction. The suggested range is 0.05 to 4.

#### **\*ZNCG *x***

Normal change in non-condensable gas global mole fraction. The suggested range is 0.05 to 4.

#### **\*ZAQ *x***

Normal change in aqueous component global mole fraction. The suggested range is 0.05 to 4.

**DEFUALTS:**

*NORM	*PRESS	500	**kPa
	*SATUR	0.2	
	*TEMP	30	** C deg
	*Y	0.2	
	*X	0.2	
	*W	0.2	
	*ZO	0.2	
	*ZNCG	0.2	
	*ZAQ	0.2	
	*FLUIDH	2500	** J/gmol

**CONDITIONS:**

- \*TEMP is not used with \*ISOTHERMAL.
- \*ZO, \*ZNCG, \*ZAQ and \*FLUIDH are not used with \*TFORM \*SXY which is the default.
- \*Y, \*X and \*W are not used with the \*ZT or \*ZH options of \*TFORM.
- \*FLUIDH is used only with \*TFORM \*ZH.

**EXPLANATION:**

This keyword specifies the typical changes in the basic variables during a timestep. These are used for automatic time-step selection.

Examples:

```
*NORM *PRESS 145.0 *SATUR 0.08
```

The timestep size is calculated from the formula

$$T^{N+1} = T^N * \frac{1.75 \text{ DNORM}(I)}{D(I)_{\text{MAX}}^N + 0.75 * \text{DNORM}(I)} \Big|_{\text{MIN OVER I}}$$

where D(I)<sub>N-max</sub> is the maximum change in each basic variable during the previous timestep.

**Numerical Sets**

Different values of \*NORM can be assigned to different regions of the reservoir. See \*NUMSET and \*NUMTYPE.

## Convergence Tolerances (Optional)

\*CONVERGE, \*MATBALTOL

### PURPOSE:

Overwrite default converge tolerances.

### FORMAT:

```
*CONVERGE *TOTRES (*NORMAL | *LOOSE | *LOOSER  
| *TIGHT | *TIGHTER | x)  
*CONVERGE { *MAXRES (( eqn_id ) x ) }  
  eqn_id = comp_name | *ENERGY | *PHASEQ  
*CONVERGE { *VARONLY | *PRESS x | *SATUR x | *TEMP x  
| *Y x | *X x | *W x | *BHP x | *ZO x | *ZNCG x | *ZAQ x | FLUIDH x }  
*CONVERGE *WELLRES x  
*MATBALTOL x
```

### DEFINITIONS:

#### \*TOTRES

Convergence is checked by comparing scaled equation residuals for each equation with the specified tolerances on a **grid-average basis**, plus scaled equation residuals on a per-block basis. The suggested range of *x* is 1e-7 to 2. Smaller values of *x* will result in smaller material balance errors, but at the cost of more iterations. Large values of *x* will defeat residual checking as a convergence criterion.

Form	Value of <i>x</i> used
*TOTRES <i>x</i>	Specified value
*TOTRES *LOOSER	1.0e-1
*TOTRES *LOOSE	1.0e-2
*TOTRES *NORMAL	1.0e-3
*TOTRES *TIGHT	1.0e-4
*TOTRES *TIGHTER	1.0e-5

#### \*MAXRES

Convergence is checked by comparing scaled equation residuals with the specified tolerances, on a **per-block basis**. The suggested range of *x* is 1e-7 to 2. Smaller values of *x* will result in smaller material balance errors, but at the cost of more iterations. Large values of *x* will defeat residual checking as a convergence criterion. Use this keyword to disable convergence checking based on averaging over the entire grid.

Form	applies <i>x</i> to
*MAXRES <i>x</i>	all equations
*MAXRES ' <i>comp_name</i> ' <i>x</i>	fluid component's flow equation (see *COMPNAME)
*MAXRES *ENERGY <i>x</i>	energy flow equation
*MAXRES *PHASEQ <i>x</i>	phase equilibrium constraint equation (see *TFORM)

**\*VARONLY**

Only the changes in variables are used as the convergence criterion. Use this keyword to over-ride the default convergence method \*TOTRES.

**\*PRESS *x***

Pressure tolerance (kPa | psi). The suggested range is 1 kPa to 100 kPa.

**\*SATUR *x***

Saturation tolerance. The suggested range is 0.0001 to 0.1.

**\*TEMP *x***

Temperature tolerance (C deg | F deg). The suggested range is 0.01 C deg to 10 C deg. Note that this quantity is a temperature difference, and so conversion between C (or K) and F (or R) involves only the factor 1.8.

**\*Y *x***

Gas mole fraction tolerance. The suggested range is 0.0001 to 1.

**\*X *x***

Oil mole fraction tolerance. The suggested range is 1e-7 to 1.

**\*W *x***

Water mole fraction tolerance. The suggested range is 1e-7 to 1.

**\*BHP *x***

Well bottomhole pressure tolerance (kPa | psi). The suggested range is 1 kPa to 100 kPa. Use of \*WELLRES is recommended over this subkeyword to control deviations from specified well rates.

**\*ZO *x***

Oleic component global mole fraction tolerance. Also applies to water component. The suggested range is 0.0001 to 0.1.

**\*ZNCG *x***

Non-condensable gas global mole fraction tolerance. The suggested range is 1e-7 to 1.

**\*ZAQ *x***

Global mole fraction tolerance of aqueous component other than water. The suggested range is 1e-7 to 1.

**\*FLUIDH *x***

Enthalpy tolerance, used only for \*TFORM \*ZH.

**\*WELLRES *x***

Relative tolerance of residual of each well rate constraint equation. The suggested range of *x* is 1e-5 to 1. This method is recommended over \*BHP for controlling deviations from the target rate.

If the primary iterating variables have converged, and the residual of a well rate constraint equation exceeds *x* times the target rate, another Newton iteration is done. Smaller values result in smaller deviations from the specified well rates, but at the cost of more iterations. Large values of *x* will defeat residual checking as a convergence criterion.

For example, *x* = 0.001 means that the calculated rate after convergence will not differ from the specified target rate by more than 0.001 of the target rate value.

**\*MATBALTOL *x***

Maximum allowed increase in the relative material balance error allowed over a timestep. The allowed range is 1e-7 to 1.

This is the maximum amount by which the relative material balance error is allowed to increase in each timestep. If the \*CONVERGE criteria are satisfied, and the material balance error over the timestep exceeds *x*, another Newton iteration is done.

Smaller values result in a smaller material balance error at the end of the run, at the cost of more iterations. Smaller values also may cause excessive iterations and timestep cuts if oscillating phase switch flags cause the material balance error to exceed *x*, in which case a larger *x* is recommended.

**DEFAULTS:**

Of the three methods available for convergence control of the fluid flow equations (\*TOTRES, \*MAXRES, and \*VARONLY), \*TOTRES \*LOOSE is the default.

The default variable tolerances are (see EXPLANATION, below):

*CONVERGE	*PRESS	50	**kPa
	*SATUR	0.05	
	*TEMP	5	** C deg (0.01 for *ZH)
	*Y	0.05	
	*X	0.05	
	*W	0.05	
	*BHP	(same value as *PRESS)	
	*ZO	0.05	
	*ZNCG	0.05	
	*ZAQ	0.05	
	*FLUIDH	500.0	**J/gmol
	*WELLRES	0.001	
*MATBALTOL		0.0001	

If \*MAXRES is present but  $x$  is absent, these internal default values are used. If \*MAXRES  $eqn\_id\ x$  is used, then each equation not referenced uses its internal default.

0.5	- component #1 (usually water)
5e-3	- other aqueous components (numw > 1)
0.05	- oleic and gaseous components
5e-5	- Non-condensable components when numw > 1
1.5	- energy
0.05	- phase equilibrium

If \*TOTRES is in effect and \*MAXRES is absent, the criteria values for maximum scaled residual for each block are 100 times the \*TOTRES criteria values.

### CONDITIONS:

\*TEMP is not used with \*ISOTHERMAL.

\*ZO, \*ZNCG and \*ZAQ are not used with \*TFORM \*SXY (which is the default \*TFORM option).

\*SATUR, \*Y, \*X and \*W are not used with the \*ZT or \*ZH options of \*TFORM.

\*TOTRES, \*MAXRES and \*VARONLY are mutually exclusive. If more than one of these keywords appears, then the one that appears last in the data file is used.

These keywords may be located also in the Well and Recurrent Data section.

### EXPLANATION:

There are two basic types of convergence checking for grid-block equations: variable changes and equation residuals.

#### Variable Changes

Triggered with subkeyword \*VARONLY, this option specifies that convergence is achieved when the changes of all iterating and secondary reservoir variables (e.g., pressure) for all grid blocks over a Newton iteration are smaller than the corresponding tolerances. The default values of these tolerances are quite large, so use subkeywords like \*PRESS to over-ride the defaults. \*VARONLY corresponds to the default convergence control method used by STARS before version 2000. This method is recommended only for certain specific purposes like testing and debugging convergence difficulties.

#### Equation Residuals

When subkeyword \*VARONLY is absent, convergence is achieved when the scaled grid-block equation residuals are smaller than corresponding tolerances. There are two kinds of residual checking, per-block and grid-average, each with a separate set of tolerances. For per-block checking, the comparison of scaled equation residual to per-block tolerance is done for each grid block separately. For grid-average, both the unscaled residual and scale factor for each equation (component, energy, etc.) are summed over the entire grid, and the resulting scaled average value is compared to the grid-average tolerance.

Subkeyword \*MAXRES causes residual convergence checking to be done for each equation on a per-block basis only. No grid averaging is done. The \*MAXRES keyword causes per-block tolerances to be defaulted as shown above and then over-ridden with input values. This is called the “maximum” method because the entire equation set is not deemed converged until the maximum residual over the entire grid satisfies its tolerance.

Subkeyword \*TOTRES (the default) combines per-block and grid-average checking by triggering several actions. First, it defaults the grid-average tolerances and then over-rides them with any input data following the \*TOTRES subkeyword. Second, it defaults any unassigned per-block tolerances with values that are 100 times the grid-average tolerances. Third, it causes residual convergence checking to be done for each equation on a grid-average basis as well as on a per-block basis. The intent here is that grid-average criteria are tight to control material balance error, while per-block criteria are looser (by 100 times) to reduce excessive iterations by individual blocks while preventing those blocks slipping into a completely unphysical solution.

Since \*TOTRES uses grid-average, of the three options it is closest to controlling the final material balance error.

To use \*TOTRES with your own per-block criteria, before \*CONVERGE \*TOTRES put \*CONVERGE \*MAXRES followed by per-block tolerance data.

Both \*TOTRES and \*MAXRES also use variable changes after iteration #6, by flagging successful convergence if either the residual *OR* the variable changes are less than tolerance values. Therefore, modification of variable change tolerances like \*PRESS affects not only the \*VARONLY option but also the \*TOTRES and \*MAXRES options to some degree. The large variable-change tolerance default values effectively prevent variable-change checking after iteration #6 from slowing convergence when variable changes are modest.

### Examples:

To specify a normal level of average residual convergence for flow equations, use

```
*CONVERGE *TOTRES *NORMAL
```

The following are examples of specifying maximum residual control:

```
*CONVERGE *MAXRES          ** Use internal defaults  
*CONVERGE *MAXRES x        ** Use uniform values  
*CONVERGE *MAXRES eqn_id x ** Use value for equation
```

To use \*TOTRES but non-default per-block tolerances for water and CO<sub>2</sub>, use

```
*CONVERGE *MAXRES 'Water' x2  
      *MAXRES 'CO2' x3  
*CONVERGE *TOTRES *LOOSE
```

To use uniform value *x1* except for the flow equations of water and CO<sub>2</sub>, use

```
*CONVERGE *MAXRES x1  
      *MAXRES 'Water' x2  
      *MAXRES 'CO2' x3
```

To specify variable change criteria instead of residual, use

```
*CONVERGE *VARONLY *PRESS 10 *SATUR 0.01
```

## Pseudo-Infinite Block

One way to model a constant-pressure boundary (e.g., aquifer) is via blocks with very large volumes. This method was used before proper aquifer models became available, and may still be used to bypass a restriction or undesirable behavior of an aquifer model. However, this technique may have subtle negative side effects and should be used only with great caution.

The \*TOTRES option compares the specified tolerance to a scaled residual that is averaged over the entire reservoir volume. When large blocks are added to the reservoir this average can be dominated by the new reservoir volume, effectively disabling \*TOTRES and allowing convergence to stop before normal blocks are converged. Therefore, when such large blocks are used it may be better to use \*MAXRES or \*VARONLY.

## Numerical Sets

Different convergence tolerances can be assigned to different regions of the reservoir, with a few exceptions. \*TOTRES applies to the reservoir as a whole and so cannot be assigned different values to various regions. Also, keyword \*VARONLY will switch the entire grid to checking convergence using variable changes. All variable tolerances like \*PRESS can be assigned different values to various regions. See \*NUMSET and \*NUMTYPE.

---

## **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. The allowed range is 1 to 30.

### **DEFAULTS:**

\*NEWTONCYC 15

### **CONDITIONS:**

This keyword may be located also in the Well and Recurrent Data section.

### **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

---

## **Under-Relaxation Option (Optional)**

**\*UNRELAX**

### **PURPOSE:**

\*UNRELAX controls the under relaxation option.

### **FORMAT:**

\*UNRELAX urpm

### **DEFINITIONS:**

urpm

A real number to specify the under relaxation value. The allowed range is -1 to 1.

- |     |  |
|-----|--|
| = 1 | No under-relaxation is done.   |
| < 0 | An under-relaxation parameter is calculated automatically. This option usually performs better than URPM > 0. The maximum value used after the third Newton iteration is -URPM. If this option is chosen, then start with a value of -1. -URPM < 0.5 is not recommended.   |
| > 0 | Under-relaxation parameter employed by the model after the third Newtonian iteration to aid convergence. This option should be used sparingly, that is, use values of URPM as close as possible to 1. This option will improve overall convergence only for certain types of difficult problems. Values of URPM < 0.5 are not recommended. |

### **DEFAULTS:**

\*UNRELAX -1

### **CONDITIONS:**

This keyword may be located also in the Well and Recurrent Data section.

### **EXPLANATION:**

The under relaxation parameter is employed by the model after the third Newtonian iteration to aid convergence. It is particularly useful for difficult problems. Values of 0.5 are recommended when the simulator consistently fails to converge. Use of this option may slow down the simulation if it is not required.

Example:

\*UNRELAX 0.5

---

## **Upstream Calculation Option (Optional)**

**\*UPSTREAM**

### **PURPOSE:**

\*UPSTREAM controls the upstream calculation option.

### **FORMAT:**

\*UPSTREAM      (\*NLEVEL)  
                  (\*KLEVEL)

### **DEFINITIONS:**

#### **\*NLEVEL**

Densities and capillary pressures are the N-level values, from the beginning of the timestep. Use of this option usually results in less oscillations in upstream flags.

#### **\*KLEVEL**

Densities and capillary pressures are the K-level values, which are the most up-to-date. Use of this option may result in more oscillations than \*NLEVEL, but may be necessary in preventing certain types of unphysical results. This value is recommended for dual porosity options.

### **DEFAULTS:**

\*UPSTREAM \*NLEVEL

### **CONDITIONS:**

This keyword may be located also in the Well and Recurrent Data section.

### **EXPLANATION:**

The upstream flow direction is evaluated at (1) the beginning of the timestep, (2) whenever a phase appears or disappears, or (3) whenever a well is shut in or opened.

The upstream direction depends upon the potentials of the fluid phases which are functions of pressure, temperature, capillary pressure (and hence saturations) and density (and hence composition). The pressure and temperature used are always the most up-to-date values (K level). This keyword determines what value of densities and capillary pressures to use.

\*NLEVEL is the default because it is the safer option. Use \*KLEVEL only if \*NLEVEL gives unphysical results that cannot be eliminated by better numerical control (e.g., tighter convergence tolerances, remove \*AIM). For example, \*NLEVEL may cause temperature decreases in an isolated block for no physical reason, especially during flow direction reversals. Usually this is caused by an incorrect upstream flag and \*KLEVEL may eliminate this behavior.

\*KLEVEL is not automatically better than \*NLEVEL generally and should be used with significant caution. In fact, \*KLEVEL will make a somewhat unstable physical situation even more difficult to simulate numerically. For example, steam injection into a perfectly uniform multi-layer horizontal well is somewhat unstable physically since small temperature non-uniformities will grow due to positive feedback (hotter oil has higher mobility resulting in higher injectivity). \*KLEVEL will magnify the feedback loop.

---

## **Discretized Well - Reservoir Upstream Calculation Option (Optional)**

**\*DW-RES-UPSTREAM**

### **PURPOSE:**

\*DW-RES-UPSTREAM controls the flow calculation between a discretized wellbore (DW) block and a reservoir block when fluid flows from a DW block to a reservoir block.

### **FORMAT:**

\* DW-RES-UPSTREAM (\*ON/\*OFF)

### **DEFINITIONS:**

\*ON

Use wellbore block properties to calculate wellbore-reservoir flow when the wellbore block is an upstream block.

\*OFF

Use downstream total mobility (reservoir gridlock mobility) when the wellbore block is an upstream block.

### **DEFAULTS:**

\* DW-RES-UPSTREAM \*OFF when the keyword is missing and DW-RES-UPSTREAM \*ON when only DW-RES-UPSTREAM is present.

### **EXPLANATION:**

Flow between a Discretized wellbore (DW) block and a reservoir block when the

\*WELLBORE keyword is used may be calculated two different ways. This calculation applies **only** when the DW block is an upstream direction, i.e. fluid is injected into a reservoir.

1. with \*ON keyword – DW block properties are used to evaluate the flow

rate = transmissibility\*delta P \* DW block mobility

With this treatment the steam quality injected into the reservoir will be the flowing fraction

Quality = gas density\*gas mobility/(gas density\*gas mobility + water density\*water mobility)

2. with \*OFF keyword – the total mobility of the reservoir block is used to evaluate the flow

rate = transmissibility\*delta P \* total reservoir block mobility

With this treatment the steam quality injected into the reservoir will be the in-situ fraction

Quality = gas density\*gas saturation/(gas density\*gas saturation + water density\*water saturation)

Flow between a Sink/Source injection well and a reservoir is calculated also in this fashion (see Appendix A).

---

## **Small Rates Option (Optional)**

**\*SMALL-RATES**

### **PURPOSE:**

Over-ride the default cut-off for small well rates.

### **FORMAT:**

**\*SMALL-RATES ( \*OFF | \*ON )**

### **DEFINITIONS:**

**\*OFF**

Cut-off rate is  $10^{-15} \text{ m}^3$  ( $6.3 \cdot 10^{-15} \text{ bbl}$ ,  $3.5 \cdot 10^{-14} \text{ ft}^3$ ,  $10^{-9} \text{ cm}^3$ ).

**\*ON**

Cut-off rate is  $10^{-30} \text{ m}^3$  ( $6.3 \cdot 10^{-30} \text{ bbl}$ ,  $3.5 \cdot 10^{-29} \text{ ft}^3$ ,  $10^{-24} \text{ cm}^3$ ).

### **DEFAULTS:**

If the total reservoir pore volume is greater than  $10^4 \text{ m}^3$  ( $3.5 \cdot 10^5 \text{ ft}^3$ ,  $10^{10} \text{ cm}^3$ ) then the default is \*OFF, otherwise the default is \*ON.

### **EXPLANATION:**

A computed well rate that is effectively zero may be very small but non-zero because of numerical round-off and iterative convergence effects. In several tasks in STARS a zero well rate is significant, such as for detection of completions and wells that should be shut in. For these tasks the well rate is set to exactly zero if the computed rate is smaller in magnitude than the cut-off rate. In other words, the cut-off rate determines what rates are close enough to zero to be considered as zero.

Normal field-scale simulations can use the larger cut-off rate without problems. However, simulations at smaller scales need to use a lower cut-off rate so that significant rates are not artificially truncated. This is especially true of simulations working at pore scales (micrometer) or over geological time periods (millions of years).

Zero well rates are used extensively to determine if well performance reporting is needed. In the diary (.log file) output a zero rate is reported with a blank space but a very small non-zero rate may appear as a single "0".

---

## **Convergence Precision 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 relative amount of residual reduction required before convergence is achieved. The allowed range is 1e-9 to 1e+9.

### **DEFAULTS:**

\*PRECC 1e-6

### **CONDITIONS:**

This keyword may be located also in the Well and Recurrent Data section.

### **EXPLANATION:**

Convergence of the linear equation solver occurs when the RMS value of the residuals of the equations has been reduced to precc times its original value.

If the set of linear equations is

$$Av = b$$

the RMS residual is

$$r = \| b - Av \|$$

Convergence occurs when, for the  $i$ th iteration,

$$r(i) / r(0) < \text{precc}$$

---

## 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:**

\*NORTH 30

**CONDITIONS:**

This keyword may be located also in the Well and Recurrent Data section. This quantity affects the amount of storage used. If more than one \*NORTH keyword appears then the largest num is the value used for storage allocation purposes.

**EXPLANATION:**

A "matrix failure" occurs when the number of matrix solver inner iterations is cut off (see \*ITERMAX) before the desired residual reduction is reached. In general, simulations with grids and reservoir properties that are not too far from uniform should have almost no matrix failures. However, very non-uniform grids and/or properties may result in a significant amount of matrix failures. Increasing \*NORTH above the default can reduce the frequency of these matrix failures.

Increasing \*NORTH can increase the CPU cost of each solver iteration, but a reduced number of iterations should result in a net savings. There is a modest cost in storage with increased \*NORTH.

There is no advantage in specifying a value for \*NORTH larger than the value of \*ITERMAX.

**Examples:**

A 1000-block simulation with uniform block sizes and permeabilities probably can use the default.

A 40000-block simulation with widely varying block thicknesses, and permeabilities in adjacent blocks different up to a factor of 1000, may use \*NORTH 50.

---

## 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 labelled "black", saving storage and computer time. Use of the nine-point option will result in more "black" blocks and less savings.

#### \*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:

\*SORDER \*REDBLACK for 2D and 3D grids over 20 blocks, otherwise \*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. It must be positive.

\*GAUSS

Keyword specifying that Gaussian elimination be used.

**DEFAULTS:**

\*SDEGREE 1 for 2D and 3D grids over 20 blocks, otherwise \*SDEGREE \*GAUSS.

**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, or difficult thermodynamic problems such as in-situ combustion). Larger values of max\_deg result in more calculations and a longer simulation run time.

Examples:

```
** Use Gaussian elimination
*SDEGREE *GAUSS
```

---

## Pivot Stabilization (Optional)

\*PIVOT

### PURPOSE:

\*PIVOT controls the diagonal sub-matrix inversion pivot stabilization.

### FORMAT:

\*PIVOT (\*ON | \*OFF)

### DEFINITIONS:

\*ON

Pivot stabilization is performed.

\*OFF

No pivot stabilization is performed.

### DEFAULTS:

\*PIVOT \*OFF

\*PIVOT is the same as \*PIVOT \*ON.

### CONDITIONS:

This keyword may be located also in the Well and Recurrent Data section.

### EXPLANATION:

This keyword selects the pivot stabilization of diagonal sub-matrix 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. The allowed range is 1 to 300.

### **DEFAULTS:**

\*ITERMAX 30

### **CONDITIONS:**

This keyword may be located also in the Well and Recurrent Data section.

---

## Adaptive Implicit Flag (Optional)

\*AIM

### PURPOSE:

Indicate adaptive implicit option.

### FORMAT:

\*AIM ( \*OFF | \*STAB (\*BACK freq) | THRESH frnorm )

### DEFINITIONS:

#### \*OFF

All blocks will be treated in a fully implicit manner.

#### \*STAB

Switching blocks between adaptive and fully implicit is done using a stability criterion. This is the recommended AIM method.

#### \*BACK freq

The frequency of checking for backward switching (implicit to explicit) for stability criteria option \*STAB. Parameter freq is a positive integer. The backward checking will be done for each timestep whose number is evenly divisible by freq.

#### \*THRESH frnorm

Parameter required for the threshold-type adaptive -implicit switching criterion. The allowed range is to 0 to 1. The recommended AIM method is \*STAB.

For steam problems FRNORM = 0.5 is typical. For combustion problems with vigorous reactions, experience has shown that simulator performance is very sensitive to the threshold values frnrm\*dnorm. Larger values of FRNORM will result in more blocks using the IMPES method, with decreased CPU per iteration but also decreased stability (more iterations). Use \*AIMSET to reset at well changes.

### DEFAULTS:

\*AIM \*OFF

\*BACK 20

Frnorm 0.5 for \*AIM \*STAB – see Explanation

### CONDITIONS:

The \*AIM option should not be used with multi-component water phase.

This keyword may be located also in the Well and Recurrent Data section.

## **EXPLANATION:**

When \*AIM \*STAB keyword is used, STARS uses a stability criterion check ONLY at an implicit front (boundary between explicit and implicit grid blocks). If the stability criterion is violated then an explicit block switches to an implicit block. Sometime, it happens that some explicit blocks (e.g. blocks close to faults, etc.) would violate the criterion at other locations outside of the implicit front. Stability check is CPU intensive and therefore the whole reservoir is checked also with the threshold criterion. If it is necessary to change the default value of frnorm, specify first the \*THRESH frnorm with the required value of frnorm and then the \*AIM \*STAB.

Examples:

```
*AIM *STAB    ** Backward switching frequency is 20  
*AIM *STAB *BACK 5
```

## **Limitations of Adaptive Implicit Option**

The adaptive implicit option should not be used with grid blocks whose sizes vary significantly, for example, local grid refinement. If it is necessary to use LGR then set those blocks fully implicit using \*AIMSET. Also, the adaptive implicit option should not be used for blocks where fluid porosity is near zero. If porosity goes near zero then set the blocks in question to fully implicit via \*AIMSET. Blocks with exactly zero porosity are set to fully implicit internally.

More generally, the adaptive implicit option should not be used where the through-put per pore volume changes significantly from one block to another due to a contrasting factor. The main factors are grid block size (normal to any front), porosity and permeability, but other factors (relative permeability) may also have an influence. Better stability is observed when \*AIMSET is used to force full implicitness for the offending blocks: LGR regions, high-permeability streaks, etc.

---

## Pressure and Temperature Limits (Optional)

\*MINPRES,

\*MAXPRES, \*MINTEMP, \*MAXTEMP

### PURPOSE:

Over-ride default upper and lower limits for pressures and temperatures.

### FORMAT:

\*MINPRES *minpres*  
\*MAXPRES *maxpres*  
\*MINTEMP *mintemp*  
\*MAXTEMP *maxtemp*

### DEFINITIONS:

*minpres*

Lower limit for pressure during the simulation (kPa | psi). The allowed range is from 0.01 kPa (0.0014 psi) to 1000 kPa (145 psi).

*maxpres*

Upper limit for pressure during the simulation (kPa | psi). The allowed range is from 100 kPa (14.5 psi) to  $10^9$  kPa ( $1.45 \cdot 10^8$  psi).

*mintemp*

Lower limit for temperature during the simulation (C | F). The maximum allowed value of *mintemp* is 1000 C (1340 F). Normally, the minimum allowed value of *mintemp* is 0.85 C (33.5 F). However, the minimum allowed value of *mintemp* is -100 C (-148 F) when at least one of these conditions is met: (1) no aqueous component is present (numw = 0 from keyword \*MODEL); and (2) keyword \*ICE appears in the Component Properties data section.

*maxtemp*

Upper limit for temperature during the simulation (C | F). The allowed range is from 2 C (35.6 F) to 12000 C (21632 F).

### DEFAULTS:

*MINPRES 50	** kPa (half an atmosphere)
*MAXPRES 1e6	** kPa ( $1.45 \cdot 10^5$ psi)
*MINTEMP 1	** C (33.5 F)
*MAXTEMP 2000	** C (3140 F)

### CONDITIONS:

1. *maxpres* must always be greater than *minpres*, and
2. *maxtemp* must always be greater than *mintemp*.

## **EXPLANATION:**

Pressure and temperature are compared to their upper and lower limits in the following places:

1. Initial pressure (\*PRES and \*REFPRES) and temperature (\*TEMP).
2. Well operating condition from \*OPERATE \*BHP, \*TINJW and \*PINJW.
3. Well BHP altered with \*ALTER.
4. Well and block pressures at the end of each non-linear iteration.

---

## Maximum Number of Phase Switches per Timestep (Optional)

\*PVTOSCMAX

### PURPOSE:

\*PVTOSCMAX is used to force the maximum number of phase switches (e.g. water component in gas and water phase) per timestep when a phase is appearing or disappearing in a grid block.

### FORMAT:

\*PVTOSCMAX *maxn*

### DEFINITION:

*maxn*

An integer indicating the maximum value of allowed phase switches (gas-oil, gas-water etc.) per timestep in a grid block. The allowed values are 1 to 60.

### DEFAULTS:

If \*PVTOSCMAX *maxn* is absent, then it is assumed that *maxn* = 60.

### CONDITIONS:

This keyword may be located also in the Well and Recurrent Data section.

### EXPLANATION:

When the reservoir conditions are at saturated values and reservoir fluid contains very volatile components or total heat capacity of a grid block is small (for example in fractures because of low or non-existing rock volume), it may happen that a very small change in pressure or temperature will cause phase appearance or disappearance in a grid block in almost every iteration during a timestep. This behavior may create numerical problems.

When keyword \*PVTOSCMAX is used the number of phase appearances and disappearances is limited to the specified value *maxn*. In many cases the numerical performance will improve but sometimes mass balance error may occur.

---

## Well Pre-Elimination Control (Optional)

\*MAXLAYPRE

### PURPOSE:

\*MAXLAYPRE controls the pre-elimination of wells before iterative matrix solution.

### FORMAT:

\*MAXLAYPRE nlypre

### DEFINITIONS:

nlypre

A positive integer specifying the maximum number of completion layers allowed for a well before pre-elimination of that well is defeated.

### DEFAULTS:

If keyword \*MAXLAYPRE is absent, then nlypre is 3.

### EXPLANATION:

Pre-elimination of well completion layers amounts to applying Gaussian elimination to the well equations. For wells with layers consisting of up to three contiguous blocks, this elimination induces little or no fill in addition to fill associated with the block solution. The default nlypre = 3 reflects this fact, since completion layers usually are contiguous.

A well with more than three layers will induce extra fill of around  $(nly-1)*(nly-2)$  sub-matrices each of neq\*neq, where nly is number of completion layers and neq is number of equations per block. For large nly this extra fill can be very significant, up to or exceeding the fill associated with the blocks, so well equation pre-elimination should not be done.

When pre-elimination is not done, a well equation and its primary iterating variable pass through to the iterative matrix solution. In this case a well equation enjoys the benefits of less fill and lower CPU associated with the iterative methods. The only disadvantage over pre-elimination is a possible reduction in robustness, but this is rare.

Note that for versions before 96, pre-elimination was done for all cases. For version 96 and after pre-elimination can be forced by using a large value for nlypre.

---

## Maximum Cuts Allowed (Optional)

\*NCUTS

### PURPOSE:

Control the number of timestep size cuts allowed in a timestep before the run aborts.

### FORMAT:

\*NCUTS *value*

### DEFINITIONS:

*value*

An integer specifying the maximum number of cuts allowed.

### DEFAULTS:

If \*NCUTS is absent then \*NCUTS 7 is assumed.

### CONDITIONS:

This keyword may be located also in the Well and Recurrent Data section.

### EXPLANATION:

When convergence at a certain timestep size fails, the timestep size is reduced and convergence is attempted again. It is normal to experience some convergence failures, depending on what process is being simulated. Sometimes a major adjustment of conditions is needed at some point in time, such as steam breakthrough. Several cuts may be needed and then the simulation proceeds with few cuts.

However, in some circumstances convergence cannot proceed without some intervention by the user. This is characterized by many convergence failures in a single timestep. This case is separate from the "normal" case by the number of convergence failures in one timestep.

Keyword \*NCUTS lets the user adjust this criterion to suit the expected level of difficulty of the run.

However, for most types of convergence failure, reduction of timestep size below a certain minimum value (1.0e-8 days) is not helpful. Therefore, increasing \*NCUTS above the default *value* rarely helps since the run will stop if the minimum timestep size is violated.

---

## **Number of Parallel Processing Threads (Optional) \*PNTHRDS**

**PURPOSE:**

Specify the number of parallel processing threads to use in the simulation.

**FORMAT:**

\*PNTHRDS *ithrds*

**DEFINITIONS:**

*ithrds*

Number of parallel processor threads to use.

**DEFAULTS:**

If keyword \*PNTHRDS is absent then the following default value is used, in the order of decreasing priority in the condition. Here  $n_{cpu}$  is the number of logical CPUs available.

Condition	Default
Command-line argument “-parasol <i>n</i> ” used:	$\min(n, n_{cpu})$
Any parallel processing option is specified:	$\min(2, n_{cpu})$
Otherwise:	1

See keyword \*SOLVER for a description of “-parasol *n*”.

**CONDITIONS:**

In order to use parallel processing (*ithrds* > 1), the parallel computing licensing feature must be enabled and sufficient parallel license tokens must be available.

**EXPLANATION:**

If *ithrds* is set to a number greater than the number of processors, performance will degrade. If *ithrds* > 2 then the solver \*PPATTERN should be changed in order to balance the load properly, otherwise poor performance is likely to occur.

**Example:**

Use either of two methods to enable parallel processing for both matrix solution and Jacobian building using all 8 of a machine’s logical CPUs.

1. Command-line argument: -parasol 8 -doms
2. Keywords:

```
*PNTHRDS 8
*SOLVER *PARASOL
*DPLANES
```

---

## **AIMSOL/PARASOL Switch (Optional)**

**\*SOLVER**

### **PURPOSE:**

Choose which solver to use, AIMSOL or PARASOL.

### **FORMAT:**

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

### **DEFINITIONS:**

**\*AIMSOL**

Use CMG's non-Parallel Iterative Solver.

**\*PARASOL**

Use CMG's Parallel Iterative Solver.

### **DEFAULTS:**

If both keyword **\*SOLVER** and command-line argument “-parasol” are absent, then **\*AIMSOL** is assumed.

### **CONDITIONS:**

**\*SOLVER \*PARASOL** is required in order to solve the linear system of equations in parallel, in which case the conditions of keyword **\*PNTHRDS** apply.

Command-line arguments “-parasol” and “-aimsol” over-ride keyword **\*SOLVER**.

### **EXPLANATION:**

The CMG Launcher controls parallel STARS with these command line arguments:

<b>Command-line argument</b>	<b>Equivalent to keywords</b>
-parasol <i>n</i>	<b>*SOLVER *PARASOL</b> <b>*PPATTERN *AUTOSLAB <i>n</i></b> <b>*PNPROSL <i>n</i></b> <b>*PNTHRDS <i>m</i></b> <b>*SOLVER *AIMSOL</b> (overrides any PARASOL keywords)
-aimsol	

where *n* is an integer greater than 0, representing the desired target number of threads and *m* = min(*n*,*n<sub>cpu</sub>*) where *n<sub>cpu</sub>* is the number of logical CPU's on the current machine.

---

## Number of PARASOL Classes for GMRES (Optional)

\*PNPROSL

### PURPOSE:

Choose the number and scaling of GMRES vector operation classes (a class is defined as a disjoint set of blocks) used in the scaling and GMRES iteration.

### FORMAT:

\*PNPROSL *nprosl*

### DEFINITIONS:

*nprosl*

The number of parallel processes to target for the scaling and also for the vector operations in the GMRES iteration.

### DEFAULTS:

Optional keyword. If \*PNPROSL is absent then *nprosl* is equal to the target number of level one classes.

### CONDITIONS:

This keyword is used only with \*SOLVER \*PARASOL.

### EXPLANATION:

This keyword controls the number of solver parallel classes used for scaling and for vector operations in the GMRES iteration.

The classes defined by PNPROSL are used in the scaling and the low level parallelization of GMRES vector operations and are different from the classes defined by \*PPATTERN. The default of \*PNPROSL is recommended.

## **Red-Black Ordering Check for Parasol (Optional)**

**\*CHECKRB**

### **PURPOSE:**

Choose when to abandon using Red-Black Ordering for a PARASOL class (a class is defined as a disjoint set of blocks).

### **FORMAT:**

**\*CHECKRB** (\*ON (redmin) | \*OFF)

### **DEFINITIONS:**

**\*CHECKRB \*OFF**

Always use Red-Black ordering for a class.

**\*CHECKRB \*ON**

Red-Black ordering is abandoned for a class in which the fraction of red blocks is too small (when the number of red blocks is less than the fraction redmin times the number of black blocks).

**redmin**

Fraction which determines when red-black ordering is abandoned (see the explanation under **\*CHECKRB \*ON** above). Any positive value not exceeding 1 is valid. If no value for redmin is entered after **\*ON**, redmin defaults to 0.6.

### **DEFAULTS:**

Optional keyword. If **\*CHECKRB** is absent then **\*OFF** is assumed. redmin defaults to 0.6.

### **CONDITIONS:**

This keyword is used only with **\*SOLVER \*PARASOL**.

### **EXPLANATION:**

It may not always be efficient to perform Red-black ordering in situations where the number of red blocks is small. This keyword allows the user to abandon Red-Black ordering in this situation. This keyword has no effect if a red-black ordering has not been specified or defaulted. When red-black ordering is abandoned because of application of this option, the factorization degree within the class is increased by one. For example, if the original direction was to treat a PARASOL class with red-black system reduction and degree 1 factorization of the reduced system, and **\*CHECKRB** found the number of red blocks to be fewer than redmin times the number of black blocks, then the class would be treated with no system reduction and degree two factorization.

---

## **Factorization Degree within PARASOL Classes (Optional)**

**\*PDEGAA**

### **PURPOSE:**

Choose the factorization degree within PARASOL classes (a class is defined as a disjoint set of blocks).

### **FORMAT:**

**\*PDEGAA *idegaa***

### **DEFINITIONS:**

*idegaa*

Factorization degree within each class

### **DEFAULTS:**

Optional keyword. If \*PDEGAA is absent then *idegaa* is equal to the value of \*SDEGREE (see \*SDEGREE in the Numerical Methods Section).

### **CONDITIONS:**

This keyword is used only with \*SOLVER \*PARASOL.

### **EXPLANATION:**

This keyword allows the user to control the degree of factorization used within a class for \*SOLVER \*PARASOL.

---

## Factorization Degree between PARASOL Classes (Optional)

\*PDEGAB

### PURPOSE:

Choose the factorization degree between PARASOL classes (a class is defined as a disjoint set of blocks).

### FORMAT:

\*PDEGAB *idegab*

### DEFINITIONS:

*idegab*

Factorization degree between classes.

### DEFAULTS:

Optional keyword. If \*PDEGAB is absent then *idegab* is *idegaa*+1 (see \*PDEGAA).

### CONDITIONS:

This keyword is used only with \*SOLVER \*PARASOL.

### EXPLANATION:

This keyword allows the user to control the degree of factorization used between classes for \*SOLVER \*PARASOL. PARASOL allows red-black elimination only within a class; thus when 1<sup>st</sup> 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 1<sup>st</sup> degree natural ordering.

---

## PARASOL Class Partitioning Pattern (Optional)

\*PPATTERN

### PURPOSE:

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

### FORMAT:

```
*PPATTERN      *AUTOPSLAB inum
or
*PPATTERN      ipatrn
or
*PPATTERN      *PARTITION
                { partition_table }
or
*PPATTERN      *PPARTITION
                { p_partition_table }
or
*PPATTERN      *GPARTITION
                { p_partition_table }
or
*PPATTERN      *APARTITION
                { p_partition_table }
```

### DEFINITIONS:

\*AUTOPSLAB *inum*

*inum* is the target number of level 1 classes. There are *inum*-1 separating plane classes plus 0 or 1 class containing demotions. The direction taken is such that the planes do not cut the dominant transmissibility direction. Thus *inum* corresponds to the target number of processors desired.

*ipatrn*

*ipatrn* can have the values 0 to 9. Figure 10.1 below provides a geometrical representation of different classes under *ipatrn* 1 through 7. Table 10.1 below summarizes the class distribution in the *ipatrn*s. Column 5 of the table shows the number of level-1 classes under each *ipatrn*, corresponding to the target number of threads desired.

Note: Unlike \*AUTOPSLAB, 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\_table*

One to 64 table rows, each starting on a new line and each with the following structure. The first four columns are quoted character strings.

column 1:	' <i>class partitioned</i> '
column 2:	' <i>1<sup>st</sup> major new class</i> '

column 3:           '2<sup>nd</sup> major new class'  
column 4:           'separator class'  
column 5:           ( \*I / \*J / \*K )  
column 6:           *ind*

Each row directs the partitioning of the first class into two major and one separator classes, with the original class no longer existing after the partition. The partitioning is planar, with the separator plane normal to the I, J, or K axis as specified, with index value *ind*. Initially there is the one class 'FIELD'; each row creates three new classes and destroys one, for a net gain of two classes per row. The names serve only to identify the classes while the pattern is being created; they are not referred to thereafter.

#### *p\_partition\_table*

One to 64 table rows, each starting on a new line and each with the following structure. All four columns are quoted character strings.

column 1:           'class partitioned'  
column 2:           '1<sup>st</sup> major new class'  
column 3:           '2<sup>nd</sup> major new class'  
column 4:           'separator class'

Like *partition\_table*, except that the simulator uses an algorithm to decide what direction plane to use and where to put it.

#### \*PPARTITION

This partitioning method equalizes class sizes as much as possible and minimizes the size of the separator class.

#### \*GPARTITION

Use Alan George's rooted-level-structure partition method, which is like \*PPARTITION but doesn't use planes.

#### \*APARTITION

Use the "agglomeration partition" method, which is like \*GPARTITION but provides classes somewhat more nearly equal in size but somewhat less regular in shape.

#### DEFUALTS:

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 linear equation solver requires the partitioning of the reservoir into disjoint sets of blocks (classes). These classes are further organized into levels. There must be no flow between blocks which are in different classes at the same level.

### **Example:**

Consider a 101 x 50 x 10 reservoir partitioned it into 3 classes as follows:

Class 1:	I = 1:50	J = 1:50	K = 1:10
Class 2:	I = 52:101	J = 1:50	K = 1:10
Class 3:	I = 51	J = 1:50	K = 1:10

The large classes, 1 and 2, have no direct flow interactions because all flow between them must go through blocks in class3. Classes 1 and 2 are at Level 1; class 3 is at level 2. Each of the three following data fragments can be used to specify this class partitioning.

```
** Method 1
*PPATTERN *AUTOSLAB 2
** Method 2
*PPATTERN 2
** Method 3
*PPATTERN *PARTITION
'FIELD' 'Class 1' 'Class 2' 'Class 3' *I 51
```

### **Reporting Class List and Distribution**

The list of Parasol classes and their distribution among cells is reported in the main output (.out) file when \*OUTPRN \*ITER level \*TSS or higher is used. Search for string “class” or “Level 1”. Since this report can be large and may be issued several times during the run, it should be used only for data debugging purposes and disabled during production runs. This report is issued for a run using \*CHECKONLY (command-line “-checkonly”) to facilitate preliminary evaluation of different Parasol class scenarios.

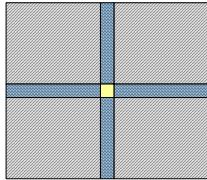
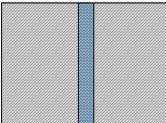
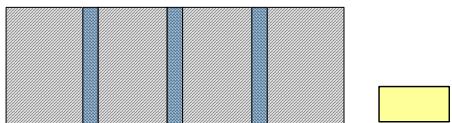
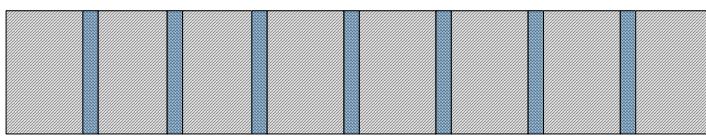
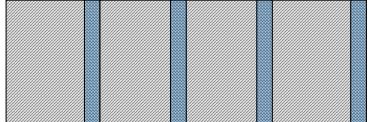
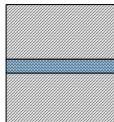
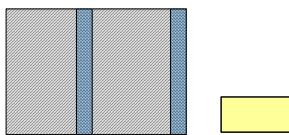
### **Discretized Wellbore**

A discretized wellbore (DW) specified by keyword \*WELLBORE in the Reservoir Description data section consists of a number of DW cells linked together. The open-pipe transmissibility between two adjacent DW cells is very large, so all the cells of any one DW should be in the same Parasol class (see \*AUTOSLAB description above). A DW often spans a large fraction of at least one reservoir direction, so normal class partitioning methods are not sufficient. For example, a DW that runs the entire length of reservoir *ipatr* = 4 in Figure 1 would have cells in 15 different classes.

A post-processing step of class partitioning ensures that all the member cells of a DW are in the same class. Specifically, if initial class partitioning puts the member cells of a DW in more than one class, all the member cells of that DW are demoted to the last class (e.g., the level-3 class in Figure 1).

For purposes of class partitioning, the member cells of a DW include each annulus cell (and tubing cell if circulating) as well as its surrounding parent cell. A command-line argument can modify the post-processing step as follows:

- dwndpar 0            DW members are annulus/tubing cells only.
- dwndpar 1            “0” plus surrounding parent cells. Default.
- dwndpar 2            “1” plus neighbours of parent cells.
- dwndpar off         Disable DW post-processing step (action before v2010).

<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 *ipatrn*s.**

ipatrn	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>ipatrn</i> 2, but separating plane is horizontal	2	3	2	1	0	
7	Like <i>ipatrn</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

---

## Parallel Jacobian-Building Planes per Domain (Optional)

**\*DPLANES**

### PURPOSE:

Specify target number of planes per domain for parallel Jacobian building.

### FORMAT:

**\*DPLANES ( *ipldom* )**

### DEFINITIONS:

**\*DPLANES ( *ipldom* )**

Enable parallel processing for Jacobian building, and optionally specify the target number of planes per Jacobian domain. Planes are chosen in the grid direction with the largest number of non-trivial planes. *ipldom* is the number of corresponding non-trivial planes in this direction per domain.

### DEFAULTS:

If both \*DPLANES, command-line argument “-doms” and \*DTYPE are all absent, then Jacobian building is performed using a single processor.

If \*DPLANES appears without *ipldom*, then *ipldom* = 4 is assumed.

### CONDITIONS:

To use \*DPLANES, multiple processors must be available. See \*PNTHRDS.

\*DPLANES keyword data is over-ridden by command-line argument “-doms”.

If both keywords \*DPLANES and \*DTYPE appear, then the \*DTYPE data is ignored.

### EXPLANATION:

There are three ways to enable parallel processing for Jacobian building. If more than one of these methods is used together, the priority is shown from highest to lowest.

1. Command-line argument “-doms”,
2. Keyword \*DPLANES, and
3. Keyword \*DTYPE.

If parallel processing is enabled for Jacobian building, the number of threads used for that task is determined by \*PNTHRDS.

A common and convenient method of enabling parallel processing for Jacobian building is via command-line argument “-doms”. This method is used by the CMG Launcher.

---

## Parallel Jacobian-Building Domain Numbers (Optional)

**\*DTYPE**

### PURPOSE:

\*DTYPE specifies domain numbers of individual blocks for parallel Jacobian building.

### ARRAY:

\*DTYPE

### DEFINITIONS:

\*DTYPE

Enable parallel processing for Jacobian building, and specify a positive integer for each block's domain number.

### DEFAULTS:

If \*DTYPE, \*DPLANES and command-line argument “-doms” are all absent, then Jacobian building is performed using a single processor.

### CONDITIONS:

To use \*DTYPE, multiple processors must be available. See \*PNTHRDS.

\*DTYPE keyword data is over-ridden by command-line argument “-doms”.

If both keywords \*DTYPE and \*DPLANES appear, then the \*DTYPE data is ignored.

### EXPLANATION:

This keyword explicitly sets the Jacobian domains of individual blocks. See EXPLANATION for keyword \*DPLANES.



# Geomechanics

---

## Summary of Geomechanical Model

There are two separate model options:

1. Plastic and Nonlinear Elastic Deformation Model
2. Single-Well Boundary Unloading Model

Each of these options is described in detail below.

### Required Keywords

This entire section is optional. If either of the two options is chosen, the keywords required for that option are described below.

#### Plastic and Nonlinear Elastic Deformation Model

The plastic deformation model performs a finite-element elasto-plastic stress analysis of the reservoir formation using a specific set of displacement and traction boundary conditions. The theory of plasticity provides the theoretical description of the relationship between stresses and strains for a material which exhibits an elasto- plastic response. Detail discussion on the theory of plasticity may be found in many textbooks on the subject, for example, Hoffman and Sachs (1953) or Prager (1959).

When a material behaves elastically, its stress-strain properties can be described by two material constants. For example, Young's modulus and Poisson's ratio is a set of such constants. However, the material may exhibit plastic behavior at an increased stress state. In this case, a yield criterion to prescribe the stress state at which plastic flow commences must be included. This is further complicated by the fact that different class of material exhibit different elasto- plastic characteristics. The post yield stress-strain behavior where deformation consists of both elastic and plastic components requires additional relationships to describe plastic flow.

Plastic strain is considered to be irreversible which occurred after the material reaches a yield state at a certain stress level. The yield criteria, Mohr-Coulomb and Drucker-Prager, which are suitable for the description of geologic material, are currently available to prescribe the yielding condition. There is also an isotropic strain-hardening option. This option allows the material to gain strength as it accumulates plastic strains. More information about constitutive laws and yield criteria for geologic material may be found in Desai and Christian (1977).

The behavior of cyclic loading and unloading as a result of cyclical injection and production processes can be modelled. During injection, the stress state at a location may reach a yield condition and begins to accumulate plastic strains. Shear dilatancy is a component of the resulting volumetric dilatation. Upon production, the material may be unloaded, resulting in the stress state dropping-off from the yield surface. During this period, the material may lose some of the reversible elastic strains.

For an elasto-plastic material, a cap model may be added to its yield criteria to avoid the unlimited hydrostatic compression which may occur in the material during production.

For nonlinear elasticity, the rock behavior may be allowed to obey either a hypoelastic constitutive model or a hyperelastic constitutive model. In the hypoelastic constitutive model, the Poisson's ratio is kept constant whereas bulk modulus and shear modulus vary with the mean effective stress. The model also has loading, unloading and neutral loading cases which are distinguished by the work done criteria  $dW$ . If  $dW$  is positive, the rock is under a loading condition. If  $dW$  is negative, the rock is under an unloading condition. When  $dW$  equals zero, neutral loading will occur. In the hyperelastic constitutive model, both values of Poisson's ratio and Young's modulus vary with minimum principle stresses. The model also has loading and unloading case which is distinguished by the shear stress criteria. If the current shear stress is greater than the reference shear stress, the loading condition is applied. If the current shear stress is less than the reference shear stress, the unloading condition occurs. The main advantage of these two constitutive models versus the elasto-plastic model is the computing efficiency. However, both nonlinear elastic constitutive models are only good for pre-failure behavior. Unlike the elasto-plastic model, once shear failure occurs, the nonlinear elastic models can not predict the post failure phenomena.

The above constitutive models and cap models can be used in 2D or 3D problems in either the Cartesian grid, corner-point grid or axisymmetric radial grid. A 3D radial (cylindrical) grid is converted in the geomechanics module into a corner-point grid type with straight lines replacing arcs between block corners. Each block's angle must not exceed 160 degrees, thereby preventing block volumes approaching zero. Also available is a plane-strain (pseudo 3D) approach, whereby 2D computations are performed successively on each y-z (or x-z) plane, based on the assumption that strain on the direction perpendicular to the plane is negligible.

The geomechanics module solves for the force equilibrium of the formation and calculates the volumetric dilatation/compression as a result of both elastic and plastic straining. The pore volume changes may be caused by a combination of compression/tension or shear stresses. These changes in pore volume and the associated changes in transmissibilities are used in the reservoir model for calculating mass and energy balances in the reservoir.

Sign conventions:

Compressive stress is positive and tensile stress is negative.

Shear stress is positive when its direction follows the coordinate direction.

The option is enabled by the presence of any one of these keywords:

*BCDOMAIN	*GCAPMAT	*MCCMODEL	*RCONT
*BCOEF	*GCFATOR	*MCETHA	*RIGIDNULL
*BIOTSCOEF	*GCINCRMT	*MCOCR	*RIGIDTOP
*CALIB_POR	*GCOUPLING	*MCOMINDX	*RPLTABD
*COHESHARD	*GDI	*MCPREHD	*RPWTABD
*COHESION	*GDJ	*MCRCSL	*SHEARMOD
*DFRICANGLE	*GDK	*MCSWINDX	*SOLVERG
*DISPLACTOL	*GEODEPTH	*MOHRCOUL	*SPECGRAV
*DLOADBC	*GEODOMAIN	*MOHRCOUL	*STIFFCOM1
*DLOADBC3D	*GEOGRID	*NB	*STIFFCOM2
*DLOADIJK	*GEOMECH	*NE	*STIFFINIT
*DRUCKER	*GEOM3D	*NINCS	*STIFFTANG
*ECOEF	*GEORBLOCK	*NTERGEO	*STRESS
*ELASTMOD	*GEOROCK	*NLINEAR	*STRESS3D
*EXPNI	*GEOTYPE	*NODE4	*STRESSGRAD
*EXPN2	*GEXPONENTN	*NODE8	*STRESSGRAD3D
*FORCETOL	*GLOADBC	*NODE9	*TDMAX
*FPVOLM	*GNULL	*NTB	*TDMIN
*FRANGSOFT	*GPATM	*NTE	*TRESCA
*FRATIO	*GPOLY	*PLOADBC	*URBCOEF
*FRICANGLE	*GPTOLMUL	*PLOADBC3D	*URECOEF
*FRICANGMN	*GROTATEI	*PLSTRAINY	*UREXPNI
*FRICANGMX	*GROTATEJ	*POISSRATIO	*UREXPN2
*GAMMA	*GROTATEK	*PRESCBC	*URNB
*GAUSSPNT	*GTRANSLI	*PRESCBC3D	*URNE
*GCAPMODEL	*GTRANSLJ	*PRINTGEO	*URNTB
*GCAPLOC	*GTRANSLK	*RCONBK	*URNTE
*GCAPD	*GULBULKMOD	*RCONBT	*VONMISES
*GCAPW	*HARDEN	*RCONF	*WRADIUS
*GCAPR	*INTPOWER	*RCONLF	*YLDSTRESS
*GCAPTEN	*MCOEF	*RCONRT	

Of these keywords some are mandatory, representing the minimum data required in order to use the option (all other data have defaults):

*GEOMECH	Main keyword for coupling
*ELASTMOD	Young's elastic modulus
*POISSRATIO	Poisson's ratio
*STRESS or *STRESS3D	Initial stresses for 2D and 3D, respectively
*GEOM3D	For 3D finite elements
*GEOGRID, *GDI, *GDJ, *GDK	For independent geomechanics grid

**Note:** For a hyperelastic constitutive model, keywords \*ELASTMOD and \*POISSRATIO are not mandatory.

## Multiple Rock Types

There is a multiple rock-type option which is enabled with the \*GEOROCK and \*GEOTYPE keywords. If these keywords are absent, it is assumed that there is only one rock type. The following keywords are applied to the current rock type number:

\*ELASTMOD, \*POISSRATIO, \*YLDSTRESS, \*COHESION, \*HARDEN,  
\*FRICANGLE, \*BIOTSCOEF, \*GRTEMTAB, \*GRPORTAB

## Single-Well Boundary Unloading Model

The boundary unloading model is restricted to an axisymmetric radial grid analysis where the wellbore is located at the axis. The model performs an elasto-plasticity analysis as in the plastic deformation model described above. In addition, it allows the user to specify the external boundary stress to be unloaded as a result of sand movement into the wellbore. This reduction in well boundary compressive stress may lead to tension failure adjacent to the perforations.

This model is enabled by the presence of the keyword \*UNLOADSTR. The keyword \*WRADIUS is used to define the well radius at which boundary unloading is to take place. In addition to \*UNLOADSTR, all required keywords for the plastic deformation model must be present and the other optional keywords may also be used to define parameters for the model.

It should be noted that during the finite-element analysis, the failed finite elements remain in the model at all time. They are not detached from the model even though the failed elements may contain tension fractures and have little or no strengths. Additionally, the effects of the removal of sand grains from the matrix are not taken into account. These may cause the inadequate description of boundary conditions for the finite elements adjacent to failed elements. These aspects are currently being worked on by CMG.

## Host Grid versus Independent Grid

The geomechanics model uses a deformable finite-element grid to perform its calculations. There are two options for specifying the origin of the geomechanics grid description.

1. Use **Host Grid** when keyword \*GEOGRID is absent. The geomechanics grid is constructed such that it exactly overlays the grid of the host fluid-flow simulator at initial conditions. More precisely, the two grids are collocated which means that corresponding cell sides and vertices occupy the same location in 3D space. As the simulation progresses, the Geomechanics grid is allowed to move but the assumption of one-to-one cell correspondence remains. Not all host grid features are supported by the geomechanics grid; see “Restricted Host Grid Options”, below.
2. Use **Independent Grid** specified by \*GEOGRID and its associated keywords. Communication between the geomechanics model and its host simulator is accomplished using a mapping algorithm between the geomechanics and host grids. This technique places few restrictions on the host grid options that can be accommodated.

## **Restricted Host Grid Options**

When geomechanics uses the host (fluid-flow) grid, the following host grid options may not be used:

1. Local refined grids (\*REFINE) and dynamic gridding (\*DYNAGRID).
2. Faulted grids.
3. Grids in which the corners of adjacent blocks do not coincide. This condition can occur for corner-point grids, especially those generated by mapping software. Also, this condition can be found in Cartesian-based grids that are known generally as variable-thickness and variable-depth and require the \*VARI sub-option of keyword \*GRID in the Reservoir Description data section. If this grid condition does exist in data without \*GEOM3D, a fatal error message is issued when geomechanics is initialized. For \*GEOM3D data this condition is not detected in STARS but is detected in Builder, so such data should be passed through Builder to do this check. A conversion tool is available for \*VARI data.

## **Converting \*VARI to Corner-Point**

Geomechanics will reject host \*GRID \*VARI data that has non-coincident adjacent block corners, as described above. Use keyword \*CONVERT-TO-CORNER-POINT in the "Reservoir Description" data section to convert such data internally to a corner-point grid that does satisfy this restriction. Each new single corner location is simply the average of the different \*VARI corner locations. Volumes and transmissibilities of individual grid blocks will differ from the \*VARI grid, but fractional changes should be reasonable for a well-formed \*VARI grid and global quantities like total pore volume should be little different. More extreme variable depth and thickness situations may not convert satisfactorily, in which case some manual adjustment of the original \*VARI data is recommended. In addition, deliberate modelling of faults cannot be converted with this keyword. In all cases, you can view both grid types in Results using data sets with and without the keyword.

## **Modelling Geomechanics Outside Host Fluid Reservoir**

Use \*GEOGRID to model geomechanics beyond the bounds of a fluid reservoir. Keyword \*FLUIDHEAT is no longer available.

## **References**

1. Hoffman, O. and Sachs, G., "Introduction to the Theory of Plasticity for Engineers," McGraw-Hill, 1953.
2. Prager, W., "An Introduction to Plasticity," Addison-Wesley, Amsterdam and London, 1959.
3. Desai, C.S. and Christian, J.T., "Numerical Method in Geotechnical Engineering," Chapter 2 & 3, McGraw-Hill, 1977.

## **Output of Geomechanics Responses**

A large number of possible geomechanics response quantities are available for viewing in CMG's graphical post-processor "Results". These quantities are available in the SRF\_GRID list associated with \*OUTSRF \*GRID, under the heading "The following are available only with \*GEOMECH". Typical quantities are stresses, strains, displacements and material model parameters like Young's modulus. Each quantity can be used also with the special history types \*BLOCKVAR, \*MAXVAR, MINVAR and AVGVAR. Special history type \*BLOCKVAR allows you to plot a single quantity in a single block versus time, not limited to output times defined by \*WSRF \*GRID.

In addition, special history type \*STRESSDIFF allows you to plot stress differential (maximum principal stress minus minimum principal stress) of a grid cell versus time, whereas a true triaxial stress-strain plot consists of \*STRESSDIFF versus a strain.

No \*GEOMECH output is available to the text ".out" file via \*OUTPRN.

The geomechanics status/output file with name suffix ".geo" contains the echo of entered data, grid construction and equation solution. See keyword \*PRINTGEO.

## **Geomechanics Porosities**

Available for viewing in Results are several kinds of porosity generated by geomechanical calculations. Generally, porosity is pore volume divided by bulk or gross volume, but there are several different kinds of pore volume and bulk volume.

*True Porosity* is current pore volume divided by current bulk volume, each generated by geomechanics calculations. This ratio is dumped to SR2 via SRF\_GRID quantity VPOROSTGEO and appears as Results menu item "True Porosity - Geomechanics".

*Geo-corrected Porosity* is current pore volume divided by initial bulk volume. This ratio is dumped to SR2 via SRF\_GRID quantity VPOROSGEO and appears as Results menu item "Porosity – Geo-corrected". Current pore volume is generated by geomechanics calculations, but initial bulk volume is static. The initialization process ensures that each geomechanics cell has the same initial bulk volume and pore volume as the corresponding fluid-flow cell.

Fluid-flow equations cannot use geomechanical porosity directly but instead get porosity from a correlation whose coefficients are provided by the Geomechanics module. Since the fluid-flow porosity is based upon initial bulk volume, it is comparable to the *Geo-corrected Porosity*. The magnitude of difference between the *Geo-corrected Porosity* and the fluid-flow porosity is an indication of the quality of the coupling between fluid-flow and geomechanics simulation. This difference is dumped to SR2 via SRF\_GRID quantity PORDIFF and appears as Results menu item "Void Porosity Difference".

Note that in the geomechanics calculations all pore volumes correspond to "void" pore volume as opposed to "fluid" pore volume (see Appendix F.2).

## **Visualizing Geomechanics Grid Deformation**

Keyword \*WSRF \*GRIDDEFORM allows the user to view grids in Results that deform with time as calculated by the geomechanics module, but there are some restrictions and other issues. See **Visualizing Geomechanics Grid Deformation** in the EXPLANATION for keyword \*WSRF.

## Storage Usage

Storage usage by the geomechanics option can be significant, especially for \*GEOM3D. As an example, the memory statistics for template STGEO023 are as follows. Enabling 2D geomechanics (without \*GEOM3D) doubled the storage required, whereas \*GEOM3D requires over 8 times the storage. The first three cases will fit on a 32-bit machine whereas the \*GEOM3D case requires a 64-bit computer. Stress/strain equations are solved at block corners and for the specified directions, so the number of equations per block may be much larger than those used to solve the fluid flow.

Case	*SOLVERG	Used (Gb)	Allocated (Gb)	Usage Ratio
No geomech		0.45	1.11	1.00
2D Geomech	*AIMSOL	0.83	2.11	1.85
2D Geomech	*PGSOLV	0.82	2.11	1.84
*GEOM3D	*AIMSOL	3.76	6.27	8.4

## Casing Treatments with Radial Grids

A radial grid normally is used to model a single wellbore. The boundary between wellbore and formation is the inner-most radial face of the Geomechanics grid. This grid face can have two types of boundary condition in the radial direction, according to the wellbore's casing treatment.

1. **Cased Hole:** A rigid casing corresponds to the boundary being **constrained**, that is, grid nodes on that boundary are not allowed to move in the radial direction.
2. **Open Hole:** No casing corresponds to the boundary being **unconstrained**, that is, grid nodes on that boundary are allowed to move radially. In addition, fluid pressure in the wellbore provides stress support in the radial direction. For example, sudden drawdown lowers the supporting stress from the wellbore which in turn causes the hole to contract.

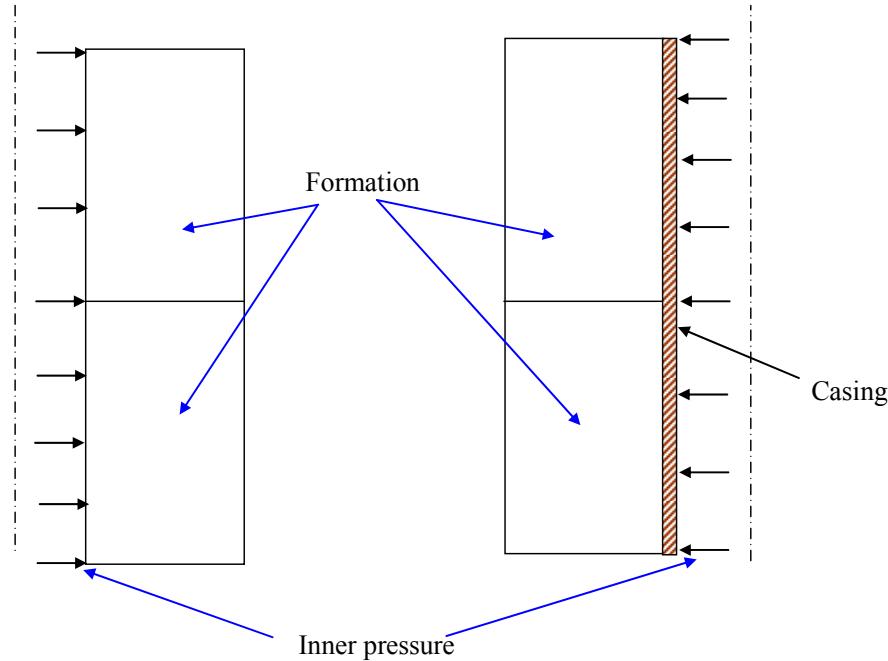
A node is constrained by applying a prescribed displacement of zero. For 2D versus 3D finite-element grids (keyword \*GEOM3D absent or present) this is done with different keywords which have different defaults. The following is a summary for radial grid types.

### 2D Radial Grids

When Reservoir Description data specifies \*GRID \*RADIAL with  $nj = 1$ , the host grid is **2D radial** since there is no resolution in the angular direction. The Geomechanics 2D finite-element grid conforms exactly to the host grid. The wellbore-formation boundary is the lower face of the inner-most radial ring which is accessed via I-direction index  $i = 1$ . For open-hole treatment, fluid pressure comes from the wellbore. See below Figure.

Most well configurations are allowed (e.g. cycle between injector and producer, inject into some layers while producing from others). Wellbore pressure is well-defined for an open well but is not for a closed well. If it is important to model an equilibrated wellbore pressure during a closed well state, keep the well open with a small rate.

Use keyword **\*PRESCBC** to constrain nodes. By default **\*PRESCBC** constrains the wellbore-formation boundary (cased-hole treatment), so you must over-ride the default to obtain an open-hole treatment.



*Wellbore Hole Treatment for 2D Radial Grid, (a) Without Casing, and (b) With Casing.*

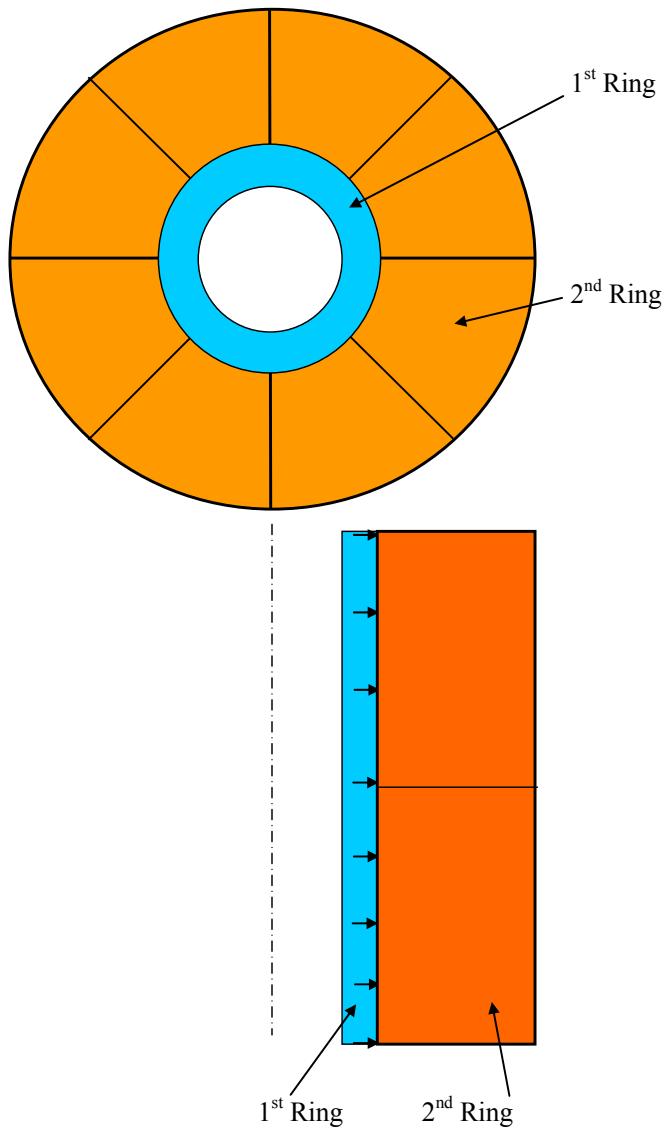
### 3D Cylindrical Grids

When Reservoir Description data specifies **\*GRID \*RADIAL** with  $nj > 1$ , the host grid is **3D cylindrical** since there is resolution in the angular direction. However, this grid has no angular resolution in the inner-most radial ring ( $i = 1$ ). The Geomechanics 3D finite-element grid (**\*GEOM3D** must be used) cannot conform to the host grid and so makes the following approximations:

1. The inner-most ring of the geomechanics grid is  $i = 2$ .
2. A corner-point grid is used, with straight lines replacing arcs between block corners. Each block's angle must not exceed 160 degrees, thereby preventing block volumes approaching zero.

Input and output uses the same I-J-K index system as the host grid, but data entered for  $i = 1$  is ignored. Therefore, use I-direction index  $i = 2$  to address the wellbore-formation boundary. For open-hole treatment, fluid pressure comes from the host grid  $i = 1$ . See below Figure.

Use keyword **\*PRESCBC3D** to constrain nodes. By default **\*PRESCBC3D** does not constrain the wellbore-formation boundary (open-hole treatment), so you must over-ride the default to obtain a cased-hole treatment (opposite of the 2D case).



Wellbore Hole Treatment for 3D Radial Grid with  $n_j = 8$ .

---

## **Geomechanical Model Identifier (Optional)**

**\*GEOMECH**

### **PURPOSE:**

Enable the geomechanical model.

### **FORMAT:**

**\*GEOMECH**

### **DEFINITIONS:**

**\*GEOMECH**

This indicates that keywords from the optional geomechanical model section will follow.

### **CONDITIONS:**

Use of the geomechanical model is optional. However, if one of the sub-models is enabled, the mandatory data for that sub-model must be provided.

### **EXPLANATION:**

The geomechanical model consists of a collection of two sub-models:

1. Plastic and nonlinear elastic deformation, and
2. Single-well boundary unloading analysis.

Each of these sub-models is enabled by the presence of at least one of its associated keywords. Model 1 and model 2 cannot be run together.

See the summary at the beginning of this section.

## 3D Finite Element

\*GEOM3D

### PURPOSE:

Use 3D Finite Elements in computation.

### FORMAT:

\*GEOM3D

### DEFAULT:

If keyword \*GEOM3D and \*PLSTRAINY are absent, then 2D plane strain finite elements normal to the X (I) direction are used.

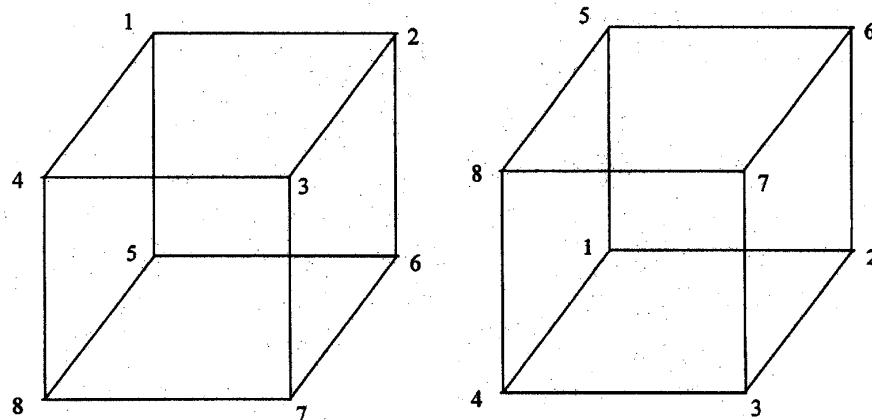
### CONDITIONS:

\*GEOM3D must appear immediately after keyword \*GEOMECH when 3D finite elements are used.

This keyword is not allowed together with \*PLSTRAINY.

### EXPLANATION:

Currently, the 3D finite element has 8 nodes which are locally ordered as follows:



\*KDIR \*DOWN

\*KDIR \*UP

### Example

\*GEOMECH

\*GEOM3D    \*\* using 3D finite elements

---

**Independent Geomechanics Grid** \*GEOGRID, \*GDI, \*GDJ, \*GDK,  
\*GEODEPTH, \*NULL, \*GTRANSLI, \*GTRANSLJ, \*GTRANSLK, \*GROTAIEI,  
\*GROTAIEJ, \*GROTAKEK, \*RCONBT, \*RCONTNP, \*RCONLF, \*RCONRT, \*RCONBK,  
\*RCONFT, \*INTERPOWER, \*GPOLY

**PURPOSE:**

Use and specify geomechanics grid that is independent of the host simulator's grid.

**FORMAT:**

```
*GEOGRID ( *GCART | *GRADIAL | *GCORNER ) ni nj nk
*GDI ( *GCON | *GIVAR ) I-size
*GDJ ( *GCON | *GJVAR ) J-size
*GDK ( *GCON | *GKVAR ) K-size
*GEODEPTH ( *GCENTER | *GTOP ) i j k depth
*NULL *IJK
*GTRANSLI I-trans
*GTRANSLJ J-trans
*GTRANSLK K-trans
*GROTAIEI I-angle
*GROTAIEJ J-angle
*GROTAKEK K-angle
*RCONBT ( *ALL | *IJK )
*RCONTNP ( *ALL | *IJK )
*RCONLF ( *ALL | *IJK )
*RCONRT ( *ALL | *IJK )
*RCONBK ( *ALL | *IJK )
*RCONFT ( *ALL | *IJK )
*INTPOWER n
*GPOLY
```

**DEFINITIONS:**

\*GEOGRID ( \*GCART | \*GRADIAL | \*GCORNER ) ni nj nk

Use and specify a geomechanics grid that is independent of the host grid.

The number of cells in each grid direction is *ni* (I-direction), *nj* (J-direction) and *nk* (K-direction). The K axis orientation (k increasing up or down) is the same as the host simulator grid which was specified by \*KDIR.

For \*GCART the geomechanics grid is Cartesian and \*GEOM3D is used.  
The I, J and K grid directions are associated with X, Y and Z spatial directions, respectively.

For \*GRADIAL the geomechanics grid is 2D radial (*nj* = 1) or 3D radial (*nj* > 1). The I, J and K grid directions are associated with radial, angular and Z spatial directions, respectively.

For \*GCORNER the geomechanics grid is specified using keywords available to \*GRID \*CORNER (\*ZCORN, \*XCORN, \*YCORN, \*COORD and \*CORNERS). When the manual entries for those keywords refer to \*DI, \*DJ or \*DK, use \*GDI, \*GDJ and \*GDK, respectively, in the \*GCORNER context. For example, you may combine \*ZCORN with \*GDI and \*GDJ. For \*GCORNER the finite element method \*GEOM3D is used.

**\*GDI ( \*GCON | \*GIVAR ) *I-size***

Specify cell size (m | ft | cm) in the I direction.

For \*GCON, *I-size* is a single number that is applied to all *ni* cells.

For \*GIVAR, *I-size* is *ni* separate numbers for cells with i=1 to i=*ni* in that order.

**\*GDJ ( \*GCON | \*GJVAR ) *J-size***

Specify cell size in the J direction. For \*GCART *J-size* is linear length (m | ft | cm). For \*GRADIAL *J-size* is angle in degrees and the total angle described may not exceed 360 degrees.

For \*GCON, *J-size* is a single number that is applied to all *nj* cells.

For \*GJVAR, *J-size* is *nj* separate numbers for cells with j=1 to j=*nj* in that order.

**\*GDK ( \*GCON | \*GKVAR ) *K-size***

Specify cell size (m | ft | cm) in the K direction. The K axis orientation (k increasing up or down) is the same as the host simulator grid which was specified by \*KDIR.

For \*GCON, *K-size* is a single number that is applied to all *nk* cells.

For \*GKVAR, *K-size* is *nk* separate numbers for cells with k=1 to k=*nk* in that order.

**\*GEODEPTH ( \*GCENTER | \*GTOP ) *ij k depth***

Specify depth (m | ft | cm) of grid via *depth* for reference cell (*i,j,k*). The reference for depth is the same as for the host simulator's grid.

For \*GCENTER, or if both \*GCENTER and \*GTOP are absent, *depth* is applied to the center of cell (*i,j,k*).

For \*GTOP, *depth* is applied to the top of cell (*i,j,k*).

**\*GNULL \*IJK**

Specify which blocks in the geomechanics grid (specified by \*GEOGRID) will be null, that is, there will be no geomechanics computation.

**\*GTRANSI *I-trans***

Specify I-direction translation (m | ft | cm) of the geomechanics grid relative to the host simulator grid. For \*GCART and \*GCORNER, X = 0 in the geomechanics grid corresponds to X = *I-trans* in the host simulator grid.

**\*GTRANSJ *J-trans***

Specify J-direction translation (m | ft | cm) of the geomechanics grid relative to the host simulator grid. For \*GCART and \*GCORNER, Y = 0 in the geomechanics grid corresponds to Y = *J-trans* in the host simulator grid.

**\*GTRANSLK *K-trans***

Specify K-direction translation (m | ft | cm) of the geomechanics grid relative to the host simulator grid. For \*GCART and \*GCORNER, Z = 0 in the geomechanics grid corresponds to Z = *K-trans* in the host simulator grid.

**\*GROTATEI *I-angle***

Specify rotation angle about I axis (degrees) of the geomechanics grid.

**\*GROTATEJ *J-angle***

Specify rotation angle about J axis (degrees) of the geomechanics grid.

**\*GROTATEK *K-angle***

Specify rotation angle about K axis (degrees) of the geomechanics grid.

**\*RCONBT ( \*ALL | \*IJK )**

Specify zero K-direction (vertical) deformation on bottom of host (fluid-flow) grid.

**\*RCONTP ( \*ALL | \*IJK )**

Specify zero K-direction (vertical) deformation on top of host (fluid-flow) grid.

**\*RCONLF ( \*ALL | \*IJK )**

Specify zero I-direction deformation on left side of host (fluid-flow) grid.

**\*RCONRT ( \*ALL | \*IJK )**

Specify zero I-direction deformation on right side of host (fluid-flow) grid.

**\*RCONBK ( \*ALL | \*IJK )**

Specify zero J-direction deformation on back side of host (fluid-flow) grid.

**\*RCONFT ( \*ALL | \*IJK )**

Specify zero J-direction deformation on front side of host (fluid-flow) grid.

**\*INTPOWER *n***

Specify power *n* in the inverse distance interpolation (IDI) used to interpolate variables between host and geomechanics grids.

**\*GPOLY**

Using polynomial approach to interpolate deformation and stress between geomechanics and host grids.

**\*ALL**

This keyword refers to all blocks in the grid.

**\*IJK**

This keyword refers to selected blocks via one or more lines of the form

*i1(:i2) j1(:j2) k1(:k2)*

Each line specifies a single index or a range of indices in the I, J and K directions. The allowed range for each index is from 1 to the maximum number of blocks in that direction, and each lower index must not exceed the corresponding upper index.

**DEFUALTS:**

If \*GEOGRID is absent then the geomechanics grid is constructed to exactly overlay the host simulator's grid, in which case some restrictions are imposed on acceptable host grid options.

If \*GEODEPTH is absent then *depth* = 0 is assumed.

If \*GTRANSLI is absent then *I-trans* = 0 is assumed.

If \*GTRANSLJ is absent then *J-trans* = 0 is assumed.

If \*GTRANSLK is absent then *K-trans* = 0 is assumed.

If \*GROTATEI is absent then *I-angle* = 0 is assumed.

If \*GROTATEJ is absent then *J-angle* = 0 is assumed.

If \*GROTATEK is absent then *K-angle* = 0 is assumed.

If \*RCONBT is absent then the bottom of the host grid is not constrained.

If \*RCONTB is absent then the top of the host grid is not constrained.

If \*RCONLF is absent then the left side of the host grid is not constrained.

If \*RCONRT is absent then the right side of the host grid is not constrained.

If \*RCONBK is absent then the back side of the host grid is not constrained.

If \*RCONF is absent then the front side of the host grid is not constrained.

If \*INTPOWER is absent then *n* = 2 is assumed.

## **CONDITIONS:**

The intersection of the 3D spaces covered by the independent geomechanics and host simulator grids must not be null, that is, the two grids must have some 3D space in common.

If \*GEOGRID is used, the minimum required keywords are \*GDI, \*GDJ and \*GDK.

The translational and rotational options are available only with \*GEOGRID.

## **EXPLANATION:**

### **Grid Orientation**

Host and geomechanics Cartesian grids are assumed to have the same grid orientation, that is, the geomechanics grid X, Y and Z directions are the same as the host grid X, Y and Z directions, respectively. Specifically, the Z axis orientation (k index increasing up or down) is the same as the host simulator grid which is specified by \*KDIR.

### **Grid Translation**

The origin (X,Y,Z) = (0,0,0) of the geomechanics grid is the corner of cell (1,1,1) that is furthest away from the rest of the grid. The origin of the host grid is defined similarly. By default the origins of the two grids are collocated.

Often the geomechanics grid extends beyond the host simulator grid, sometimes in both the negative and positive direction of a spatial axis. The \*GTRANSLI/J/K keywords allow you to translate the origin of the geomechanics grid relative to the host grid origin. This lets you control the position the geomechanics grid over the host grid.

For example, a host grid covers a pay zone only but the geomechanics grid covers overburden, pay zone and underburden. Therefore, we want the geomechanics grid to extend beyond the host grid in both the -Z and +Z directions. If the host grid uses \*KDIR \*UP (origin on bottom) and the underburden thickness modelled by the geomechanics grid is 50 ft, then the vertical translation of the geomechanics grid would be specified by

\*GTRANSLK -50

This would put the geomechanics grid origin 50 ft below the bottom of the host grid (pay zone).

### **Grid Rotation**

The geomechanics grid can be rotated about the X, Y or Z axis to match with reservoir orientation. This option can also be combined with the option of grid translation. When rotation and translation are combined, the geomechanics grid is rotated first then translated.

For instance, the geomechanics grid is rotated about the X axis, Y axis and Z axis with angles of 30 degrees, 70 degrees and 120 degrees, respectively.

*GROTATEI 30	** In degrees
*GROTATEJ 70	** In degrees
*GROTATEK 120	** In degrees

## Plotting Results for Geomechanics Grid

Use keyword \*GOUTSRF \*GGRID to create a separate SR2 file set for writing results for the geomechanics grid. Geomechanics quantities (e.g., displacements) reported for individual cells may be more accurate with this SR2 file set which uses the “native” geomechanics grid. The same quantities reported in the host SR2 file set have gone through an interpolation step and so may have lost some accuracy, especially if the two grids have cells sizes that are significantly different.

## Constraining Host Grid Deformation

Deformations are calculated for all host grid cell corners via the 3D mapping. On a 2D plane boundary that the host and geomechanics grid have in common, the calculated host grid deformations normal to the plane may not be zero even though the geomechanics grid is constrained there to no deformation in that direction. Keywords \*RCONBT, etc., force the host grid to have no deformation on the indicated boundary in the normal direction, in order to match the typical (default) host grid behavior on that boundary.

For the purpose of keywords \*RCONBT, etc., “boundary” means the first non-null cell encountered when searching from the indicated direction. For example, for \*KDIR \*UP the bottom grid layer is  $k = 1$  so that \*RCONBT causes the searching in an I-J column to start at  $k = 1$  and increases  $k$  until a non-null cell is encountered. For \*KDIR \*DOWN the bottom grid layer is  $k = nk$  so that \*RCONBT searching in a column starts at  $k = nk$  and decreases  $k$  until a non-null cell is found.

The constraints of host grids can be applied on their outer boundaries or on any host grid. If one of the constrained keyword is used with \*ALL, the outer boundary corresponding to the keyword is fully constrained. By convention:

Left : from  $I = 1$  to  $I = NI$  ; Right : from  $I = NI$  to  $I = 1$   
Back : from  $J = 1$  to  $J = NJ$  ; Front : from  $J = NJ$  to  $J = 1$   
Top : from  $K = 1$  to  $K = NK$  ; Bottom : from  $K = NK$  to  $K = 1$  For KDIR DOWN  
Top : from  $K = NK$  to  $K = 1$  ; Bottom : from  $K = 1$  to  $K = NK$  For KDIR U

It should be noted that this kind of constraints is only applied to the host grids but not geomechanics grids. The constraint for geomechanics grids is controlled by the keyword \*PRESCBC or \*PRESCBC3D.

## Example

```
*GEOGRID *GCART 50 40 10      ** Cartesian grid
*GDI   *GCON   10              ** Size of grid
*GDJ   *GCON   5               ** Size of grid
*GDK   *GKVAR  2*1 5*3 3*10   ** Size of grid
*NULL  *IJK    20:30 15:30 3:5  ** Null blocks
*GEODEPTH *GTOP    1000        ** Top of first block
```

```

*GTRANSLI -100          ** X translation
*GTRANSLJ -200          ** Y translation
*GTRANSLK 500           ** Z translation
*GROTATEI 30            ** Rotate about X
*GROTATEJ 70            ** Rotate about Y
*GROTATEK 120           ** Rotate about Z
*INTERPOWER 3             ** Power used for IDI
** constrain the host grid in full (*ALL) or part (*IJK)
*RCONBT *ALL             ** On the bottom
*RCONLF *ALL             ** On the left
*RCONRT *ALL             ** On the right
*RCONBK *ALL             ** On the back
*RCONFT *ALL             ** On the front
*RCONTP *IJK             ** On the top of layer 5
 1:10  1:10  5

```

## References

Tran, D., Nghiem, L. and Buchanan L.: "Aspects of Coupling between Petroleum Reservoir Flow and Geomechanics," paper ARMA 09-08 presented at the 43rd US Rock Mechanics Symposium and 4th US-Canada Rock Mechanics Symposium, Asheville, NC, USA, June 28-July 1, 2009.

Tran, D., Buchanan, L. and Nghiem, L.: "Improved Gridding Technique for Coupling Geomechanics to Reservoir Flow," SPE Journal, Vol. 15, pp. 64-75, 2010.

---

## Independent Geomechanics Graphics

\*GOUTSRF

### PURPOSE:

Enable independent graphics file (SR2) for geomechanics.

### FORMAT:

\*GOUTSRF \*GGRID ( \*ALL | \*NONE | (\*REMOVE) *item\_list* )

### DEFINITIONS:

\*GGRID ( \*ALL | \*NONE | (\*REMOVE) *item\_list* )

Specifies grid quantities (one value for each grid block) to be written to the geomechanics SR2 file at times determined by the host simulator via \*WSRF \*GRID. Generally, each item on the GSRF\_GRID list is flagged for writing as either enabled or disabled. The simulation starts with all items disabled. Use *item\_list* (selected keywords in the GSRF\_GRID list) to enable individual items. Use \*ALL to enable all items. Use \*REMOVE with *item\_list* to disable individual items, or use \*NONE to disable all items.

Enabling GSRF\_GRID items for writing can increase the size of the geomechanics SR2 files. The availability of some items depends on the use of other geomechanics keywords or options.

### GSRF\_GRID List

The GSRF\_GRID list consists of all properties and quantities in the following table.

PRES	Fluid pressure
TEMP	Temperature
STRESI:	Effective I-direction stress ( X or R )
STRESJ:	Effective J-direction stress ( Y or theta )
STRESK:	Effective K-direction stress ( Z )
STRESSH:	Shear stress ( Y-Z or R-Z ) for plane strain only
STRESMXP:	Maximum principal stress (+ for compressive, - for tensile)
STRESMNP:	Minimum principal stress (+ for compressive, - for tensile)
STRESINT:	Intermediate principle stress (+ for compressive, - for tensile)
VMSTRESS:	Von Mises stress
STRNEPL:	Effective Plastic strain
STRESEFF:	Mean effective stress (+ for compressive, - for tensile)
STRESSM:	Mean total stress (+ for compressive, - for tensile)
STRESNORM:	Effective stress normal to fracture
PRMXDIR:	Vector of maximum principle effective stress (*GRID only)
PRMNDIR:	Vector of minimum principle effective stress (*GRID only)
STRAINI:	I-direction normal strain ( X or R )
STRAINJ:	J-direction normal strain ( Y or theta )
STRAINK:	K-direction normal strain ( Z )
STRAINSH:	Shear strain
STRNMXP:	Maximum principle strain
STRNMNP:	Minimum principle strain
STRAINVOL:	Volumetric strain

VERDSPLGEO:	Vertical displacement “up” based on geomechanics, at centre of cell
TVERDPLGEO:	Vertical displacement “up” based on geomechanics, at top of cell
SUBSIDGEO:	Vertical displacement “down” (subsidence) based on geomechanics, at centre of cell (negative of VERDSPLGEO)
TSUBSIDGEO:	Vertical displacement “down” (subsidence) based on geomechanics, at top of cell (negative of TVERDPLGEO)
VDISPL:	Vector of grid displacement
YLDSTATE:	Stress state = 0 In Elastic state = 1 On shear failure envelope = 2 On the compressive cap = 3 At the corner (intercept between cap and shear failure envelope) = 4 On the tensile cutoff surface
BIOT:	Biot’s constant
GCOHESION:	Cohesion value
HARDENING:	Hardening parameter
POISSON:	Poisson’s ratio
YIELD:	Yielding stress
YOUNG:	Young’s elastic modulus
FRICANGLE :	Friction angle
THEXPCOEF:	Linear thermal expansion coefficient of rock
THCOMPR:	Thermal compressibility of rock
BULKVOL:	Bulk volume
GEORTYPE:	Rock type number for geomechanical material

### DEFUALTS:

If \*GOUTSRF is absent then a separate geomechanics SR2 file set will not be created.

### CONDITIONS:

The host simulator (via keyword \*WSRF \*GRID) determines the times at which the specified quantities are written to the geomechanics SR2 file.

\*GOUTSRF \*GGRID may appear in the Well and Recurrent Data section as well as the Geomechanics data section.

### EXPLANATION:

Since an SR2 file set can contain only one grid, the \*GEOGRID option requires a separate geomechanics SR2 file set for the independent geomechanics grid. Keyword \*GOUTSRF triggers the creation of a geomechanics SR2 file set that is separate from the host simulator’s SR2 file set. The geomechanics SR2 files have the same base name as the host output files, with extensions “.gmch.irf” and “.gmch.mrf”. The geomechanics SR2 file set uses the same units as the host simulator’s SR2 file set.

If \*GOUTSRF \*GGRID is used when \*GEOGRID is absent, the geomechanics SR2 file set is created and the information written to it is the same as what can be written to the host SR2 file set via the geomechanics-specific subkeywords of \*OUTSRF \*GRID.

## **Example**

```
*GEOMECH
    ** Independent geomechanics grid definition
*GEORGRID *GCART 50 40 10
...
    *GOUTSRF *GGRID *PRES *STRESEFF * STRAINVOL
*TIME 0
...
    *GOUTSRF *GGRID *TEMP *NONE  ** Disable all
*TIME 100
...
    *GOUTSRF *GGRID *STRNEPL
*TIME 200
*STOP
```

---

## Plane Strain Option

\*PLSTRAINY

### PURPOSE:

Specify that strains will be calculated by a series of planes normal to the Y (J) direction.

### FORMAT:

\*PLSTRAINY

### DEFINITIONS:

\*PLSTRAINY

Indicates that the 3D strains will be calculated as a series of 2D planes normal to the Y (J) direction.

### DEFAULTS:

If \*PLSTRAINY and \*GEOM3D are absent, then the strains will be calculated by a series of planes normal to the X (I) direction.

### CONDITIONS:

This keyword works only with Cartesian and corner point grids. Radial grids are not allowed.

This keyword is not allowed together with \*GEOM3D.

### EXPLANATION:

Three-dimensional strain calculations are performed by a plane strain method, where strains are calculated in a series of 2D planes normal to one grid direction. Consequently, strains are assumed to be zero in that direction. The purpose of using this method is to improve the speed of stress-strain calculations while maintaining appropriate mechanical responses due to disturbances in the reservoir. In order to minimize estimating errors, it is recommended that the user select the direction in which the strains can be ignored. Normally, strains are smallest in the longest horizontal extent of a reservoir field.

### Example

*GEOMECH	** keyword for coupling with the geomechanics module
*PLSTRAINY	** keyword for plane strains along the Y direction

---

## Deformation Rock Type

\*GEOROCK, \*GEOTYPE

### PURPOSE:

Assign multiple deformation rock types.

### FORMAT:

\*GEOROCK typ\_num ( \*COPY old\_typ\_num )

### ARRAY:

\*GEOTYPE

### DEFINITIONS:

typ\_num

Current rock type number to be applied to data which depends on rock type. See \*GEOMECH. Data is read starting with type #1 as default. 'typ\_num' can be any number from 1 to the maximum dimensioned value.

old\_typ\_num

Previously defined rock type which is to be copied into current rock type. \*COPY is optional. This option is useful when rock types differ in only a few properties.

\*GEOTYPE

Assigns rock type numbers to the grid blocks, using any of the array reading options. This array is initialized with rock type #1 for every block.

### DEFAULTS:

\*GEOROCK 1

\*GEOTYPE \*CON 1

### CONDITIONS:

All the rock type numbers referenced by \*GEOTYPE must be defined by \*GEOROCK.

### EXPLANATION:

For a list of the keywords which can vary with rock type, see section **Multiple Rock Types** in the summary at the beginning of this chapter.

---

## **Plastic Model Formation Properties**

\*ELASTMOD, \*POISSRATIO,  
\*YLDSTRESS, \*COHESION, \*HARDEN, \*FRICANGLE, \*BIOTSCOEF, \*DILANGLE

### **PURPOSE:**

Define strength properties of the formation.

### **FORMAT:**

*ELASTMOD	<i>elastmod</i>
*POISSRATIO	<i>poissratio</i>
*YLDSTRESS	<i>yldstress</i>
*COHESION	<i>cohesion</i>
*HARDEN	<i>harden</i>
*FRICANGLE	<i>fricangle</i>
*BIOTSCOEF	<i>biotscoef</i>
*DILANGLE	<i>dil_angle</i>

### **DEFINITIONS:**

*elastmod*

Young's elastic modulus (kPa | psi | kPa).

*poissratio*

Poisson's ratio (dimensionless).

*yldstress*

Yield stress for Tresca and Von Mises materials (kPa | psi | kPa).

*cohesion*

Cohesion for Mohr-Coulomb and Drucker-Prager materials (kPa | psi | kPa).

*harden*

Hardening parameter for the linear strain hardening option (kPa | psi | kPa).

*fricangle*

Angle of internal friction for Mohr-Coulomb and Drucker-Prager materials (degrees).

*biotscoef*

Biot's coefficient.

*dil\_angle*

Dilation angle (degrees) for Mohr-Coulomb and Drucker-Prager materials. The recommended range is  $0 \leq \text{dil\_angle} \leq \text{fricangle}$ . Specifying  $\text{dil\_angle} > \text{fricangle}$  may lead to numerical instability. When  $\text{dil\_angle} = \text{fricangle}$  the results of non-associated flow and associated flow must be the same.

For some materials such as soil or sand, the use of friction angle alone (associated flow rule) may lead to a large discrepancy in computing the volumetric strain. For such materials, dilation angle *dil\_angle* is used instead to obtain more accurate results which are comparable to lab data. Generally, the non-associated flow rule is based on the plastic potential function whereas the associated flow rule is based on the yield function.

Since the stiffness matrix generated by the non-associated flow rule is unsymmetrical, an appropriated solver must be used (see \*SOLVERG).

## DEFAULTS:

If the following keywords are absent, the indicated action is taken.

<u>Absent</u>	<u>Action</u>
*YLDSTRESS	<i>yldstress</i> = 0
*COHESION	<i>cohesion</i> = 0
*HARDEN	<i>harden</i> = 0
*FRICANGLE	<i>fricangle</i> = 30
*BIOTSCOEF	<i>biotscoef</i> = 1
*DILANGLE	<i>dil_angle</i> = <i>fricangle</i>

## CONDITIONS:

Both \*ELASTMOD and \*POISSRATIO are required if the plastic deformation option is used. See the \*GEOMECH keyword. The other keywords are optional.

These keywords are rock-type dependent, and are applied to the rock type number in effect at the time they are read. See the \*GEOROCK and \*GEOTYPE keywords.

## EXPLANATION:

When the Mohr-Coulomb or the Drucker-Prager yield criteria are used, the relevant keywords are:

\*ELASTMOD \*POISSRATIO \*COHESION \*FRICANGLE \*HARDEN  
\*BIOTSCOEF

Mohr-Coulomb and Drucker-Prager are popular yield criteria for geological materials. By default, the material is assumed to be cohesionless, with a friction angle of 30 degree, a good number to use for sand, and a perfectly plastic behavior upon yielding with no strain hardening. Biot's coefficient is one, representing full interaction between pore pressure and stress.

When the Von Mises or the Tresca yield criteria are used, the relevant keywords are:

\*ELASTMOD \*POISSRATIO \*YLDSTRESS \*HARDEN \*BIOTSCOEF

Von Mises and Tresca are popular yield criteria for metal plasticity. Due to their simplicity, they are included here mainly for testing purposes.

If the elasticity is applied to a rock type, only three main keywords are needed such as:

\*ELASTMOD, \*POISSRATIO, \*YLDSTRESS. In order that the rock behaves elasticity through the course of loading, the yield stress (\*YLDSTRESS) must be a very large value.

## Temperature-dependent Properties

\*GRTEMTAB

### PURPOSE:

Specify temperature dependence of geomechanical properties.

### FORMAT:

```
*GRTEMTAB key1 ... keyn
  { T value1 ... valuen }
```

### DEFINITIONS:

\*GRTEMTAB key<sub>1</sub> ... key<sub>n</sub>

Specify which geomechanical properties will vary with temperature. Table columns are defined by one or more of the following subkeywords, in any order. Each subkeyword may be used at most once.

*ELASTMOD	Young's modulus (kPa   psi   kPa)
*POISSRATIO	Poisson's ratio (dimensionless)
*BIOTSCOEFF	Biot's coefficient (dimensionless)
*COHESION	Cohesion (kPa   psi   kPa)
*FRICANGLE	Friction angle (degrees)
*THEXPCOEF	Linear thermal expansion coefficient (1/C   1/F   1/C)

```
{ T value1 ... valuen }
```

Specify table consisting of a column of temperature values *T* plus a column of values for each quantity defined by the ordered subkeyword list key<sub>1</sub> ... key<sub>n</sub>. Temperatures *T* must be monotonically increasing down the table. The allowed number of rows is 2 to 20, inclusive.

### DEFAULTS:

If \*GRTEMTAB is absent for a rock type, formation geomechanical properties for that rock type do not depend on temperature.

If \*GRTEMTAB appears but a subkeyword is absent, the corresponding value is obtained from the primary keyword of the same name (or its default). For example, if \*GRTEMTAB appears without \*ELASTMOD, the constant Young's modulus is obtained from primary keyword \*ELASTMOD.

### CONDITIONS:

Keyword \*GRTEMTAB is available on a rock-type basis (see \*GEOROCK).

\*GRTEMTAB is not available together with the Pseudo-Dilation Model.

## **EXPLANATION:**

Significant temperature variations can be found in a number of oil recovery methods, e.g., steam injection, air injection (combustion) and electrical heating. In these cases geomaterials can experience changes in property values, which will affect geomechanical responses as well as fluid flow. The properties found in the lab to have significant temperature dependence are Young's modulus, Poisson's ratio, cohesion, Biot's coefficient, friction angle and thermal expansion coefficients. Use of \*GRTEMTAB can help improve a history match when temperature plays an important role in the recovery process.

### **Example**

The following is an example using SI units:

	*GRTEMTAB	*ELASTMOD	*COHESION	*POISSRATIO
** temp (C)	(kPa)	(kPa)		
50	1.0e8	200	0.295	
250	0.5e8	190	0.3	
450	0.1e8	185	0.303	

### **Data Sources**

Temperature dependent data of these properties can be obtained from handbooks or papers that deal with rock properties. However, a special or unique rock type may require a tailored lab test in order to determine physical properties related to a specific field study. The following references may help you understand the trends of geomaterial properties with changing temperature.

Koca, M.Y et al., "Changes in the Engineering Properties of Marble in Fire-exposed Columns", Int. J. Rock Mechanics & Mining Sciences, Vol. 43, pp. 520-530 (2006).

Heuze, F.E., "High-temperature Mechanical, Physical and Thermal Properties of Granitic Rocks – A Review", Int. J. Rock Mech. Min. Sci. & Geomech. Abstract, Vol. 20, pp. 3-20 (1983)

Madland, M.V., Korsenes, R.I. and Risnes, R., "Temperature Effects in Brazilian, Uniaxial and Triaxial Compressive Tests with High Porosity Chalk", SPE 77761 (2002).

## Porosity-Dependent Properties

\*GRPORTAB

### PURPOSE:

Specify formation geomechanical properties that depend on fluid porosity.

### FORMAT:

```
*GRPORTAB key1 ... keyn
{ φf value1 ... valuen }
```

### DEFINITIONS:

\*GRPORTAB key<sub>1</sub> ... key<sub>n</sub>

Specify which geomechanical properties will vary with fluid porosity. Table columns are defined by one or more of the following subkeywords, in any order. Each subkeyword may be used at most once.

*ELASTMOD	Young's modulus (kPa   psi   kPa)
*POISSRATIO	Poisson's ratio (dimensionless)
*BIOTSCOEFF	Biot's coefficient (dimensionless)
*COHESION	Cohesion (kPa   psi   kPa)
*FRICANGLE	Friction angle (degrees)
*THEXPCOEF	Linear thermal expansion coefficient (1/C   1/F   1/C)

{ φ<sub>f</sub> value<sub>1</sub> ... value<sub>n</sub> }

Specify table consisting of a column of fluid porosity values φ<sub>f</sub> plus a column of values for each quantity defined by the ordered subkeyword list key<sub>1</sub> ... key<sub>n</sub>. Porosities φ<sub>f</sub> must be monotonically increasing down the table. The allowed number of rows is 2 to 20, inclusive.

### DEFAULTS:

If \*GRPORTAB is absent for a rock type, formation geomechanical properties for that rock type do not depend on porosity.

If \*GRPORTAB appears but a subkeyword is absent, the corresponding value is obtained from the primary keyword of the same name (or its default). For example, if \*GRPORTAB appears without \*ELASTMOD, the constant Young's modulus is obtained from primary keyword \*ELASTMOD.

### CONDITIONS:

Keyword \*GRPORTAB is available on a rock-type basis (see \*GEOROCK).

\*GRPORTAB is not available together with the Pseudo-Dilation Model.

### EXPLANATION:

Keyword \*GRPORTAB allows you to define geomechanical properties that change with fluid porosity. The properties specified by primary keywords \*ELASTMOD, etc., normally are measured in the laboratory at constant porosity. However, fluid porosity changing due to injection or production in the field can affect geomechanical properties.

## **Example**

The following is an example using SI units:

```
*GRPORTAB  *ELASTMOD  *COHESION  *POISSRATIO  *FRICANGLE
** porosity
  0.01      1e8        200        0.29        25
  0.1       0.5e8       190        0.3          30
  0.2       0.1e8       185        0.32        35
```

## **References**

Li, L. and Aubertin, M., "A general relationship between porosity and uniaxial strength of engineering materials," *Can. J. Civ. Eng.*, Vol. 30, pp. 644-658, 2003

Phani, K.K. and Sanyal, D., "Critical reevaluation of the prediction of effective Poisson's ratio for porous materials," *J. Materials Science*, Vol. 40, pp. 5685-5690, 2005.

---

## Solid Component Properties

\*GEOSOLID, \*SDELASTMOD,

\*SDPOISSON, \*SDCOHES, \*SDFRICANG, \*SDBIOTSCF, \*SDTHEXPCF, \*SDILANG

### PURPOSE:

Assign geomechanical properties for a solid component.

### FORMAT:

*GEOSOLID	'namec'
*SDELASTMOD	<i>elastmod</i>
*SDPOISSON	<i>poissratio</i>
*SDCOHES	<i>cohesion</i>
*SDFRICANG	<i>fricangle</i>
*SDBIOTSCF	<i>biotscoef</i>
*SDTHEXPCF	<i>thexpcoef</i>
*SDILANG	<i>dil_angle</i>

### DEFINITIONS:

#### \*GEOSOLID 'namec'

Name of solid component, in quotes. This component name must be one of the solid components specified via keyword \*COMPNAME. This keyword must be the first of this keyword group and must appear at most once for each *namec*.

*elastmod*

Young's modulus of the solid component (kPa | psi | kPa).

*poissratio*

Poisson's ratio of the solid component (dimensionless).

*cohesion*

Cohesion of the solid component (kPa | psi | kPa). Use only with the Mohr-Coulomb or the Drucker-Prager yield criteria.

*fricangle*

Friction angle of the solid component (degrees).

*biotscoef*

Biot's coefficient of the solid component (dimensionless).

*thexpcoef*

Linear thermal expansion coefficient of the solid component (1/C | 1/F | 1/C).

*dil\_angle*

Dilation angle of the solid component (degrees). See the description for keyword \*DILANGLE.

## DEFUALTS:

If \*GEOSOLID appears for a component but the following keywords are absent, the indicated action is taken.

<u>Absent</u>	<u>Action</u>
*SDCOHES	<i>cohesion</i> = 0
*SDFRICANG	<i>fricangle</i> = 30
*SDBIOTSCF	<i>biotscoef</i> = 1
*SDTHEXPCF	<i>thexpcoef</i> = 0
*SDILANG	<i>dil_angle</i> = <i>fricangle</i>

## CONDITIONS:

If \*GEOSOLID appears for a component, then \*SDELASTMOD and \*SDPOISSON are required for that component.

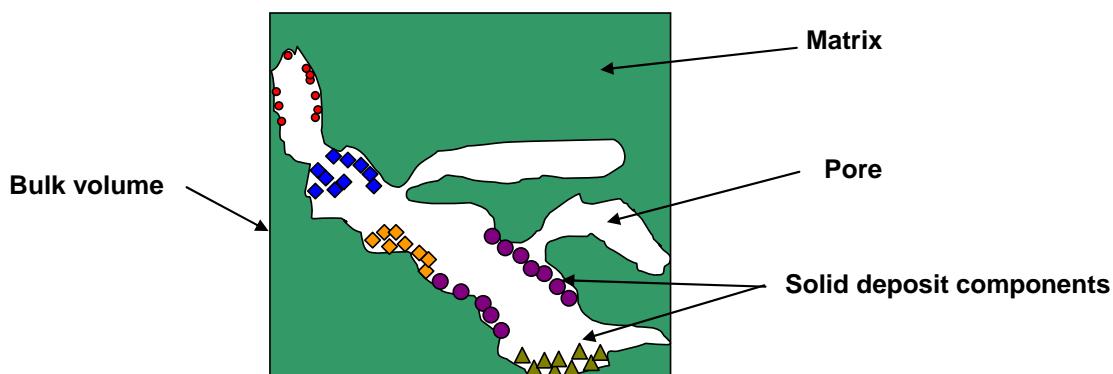
These keywords are independent of rock type defined by \*GEOROCK or its default.

This keyword group must appear once for each solid component for which geomechanical properties are assigned.

## EXPLANATION:

Solid component properties are defined in both the fluid-flow and geomechanical data sections. In the fluid-flow section (Component Properties), those properties relate to volume (density, blockage), energy (heat capacity) and chemical reactions. In the geomechanical data section, solid component properties involve mechanical deformation (Young's modulus, Poisson's ratio, cohesion, etc). In this case the geomechanical properties of a grid block depend not only on the formation material properties but also on the amount of each solid component present.

Volume fractions of the solid components are obtained directly from the flow simulation, potentially different for each grid block. Depending on its volume fraction and its geomechanical properties, a solid component can have a strong or weak influence on the deformation response.



Bulk volume of a grid block can be considered as a sum of volumes of matrix (rock), fluid and solid components.

$$V_b^0 = V_m + V_f + \sum_{i=1}^n V_s^i \quad (1)$$

Dividing both sides of (1) by  $V_b^0$  gives:

$$1 = \phi_m + \phi_f + \sum_{i=1}^n \phi_s^i \quad (2)$$

$$\text{Let: } \phi_b = \phi_m + \phi_f \quad (3)$$

Substituting (3) into (2) leads to:

$$\phi_b = 1 - \sum_{i=1}^n \phi_s^i \quad (4)$$

Property  $\chi$  of a grid block can be estimated as:

$$\chi = \left\{ \phi_b (\chi_b)^m + \left( \sum_{i=1}^n \phi_s^i (\chi_i)^m \right) \right\}^{1/m} \quad (5)$$

where:

$\chi$	Geomechanical property of a grid block
$\chi_i$	Geomechanical property of a solid deposit component i
$\chi_b$	Geomechanical property of matrix and fluid
$V_m$	Volume of matrix
$V_f$	Volume of fluid
$V_s^i$	Volume of solid deposit component i

$$\phi_m = \frac{V_m}{V_b^0} \quad \text{Volume fraction of matrix}$$

$$\phi_f = \frac{V_f}{V_b^0} \quad \text{Volume fraction of fluid}$$

$$\phi_s^i = \frac{V_s^i}{V_b^0} \quad \text{Volume fraction of a solid deposit component i}$$

$$\phi_b \quad \text{Volume fraction of matrix and fluid}$$

$$V_b^0 \quad \text{Initial bulk volume of a grid block}$$

$$n \quad \text{Number of solid deposit components}$$

$$m \quad \text{Scaling power}$$

In the above equation (5), the property  $\chi_b$  can be constant or a function of volume fraction of fluid and it will be determined by porosity-dependent properties (\*GRPORTAB).

### Example

Consider a sand-flow simulation where sand is mobilized, flows entrained in the fluid stream and then deposits again. Entrained sand “FREE\_SAN” does not contribute to formation strength and is modelled with a fluid flow equation. The original sand is called “SAND1” and the settled sand is called “SAND2”. These three sands have the same volumetric and thermal properties. Reactions change one kind of sand into another kind in response to local conditions. The Component Properties would be:

```
** Fluid component density data
*COMPNAME 'WATER' 'OIL' 'SOLN_GAS' 'FREE_SAN' 'SAND1' 'SAND2'
**
*CMM      0      0.508   0.016      1.150   1.150   1.150
*MASSDEN    0      985     320       2650
*CP          0      82E-8   1.45E-6    1.0E-7
*CT1         0      7E-4    810E-6    205E-6

** Solid component density data
*SOLID_DEN  'SAND1'  2650.0E-0  1.0E-7   205E-6
*SOLID_DEN  'SAND2'  2650.0E-0  1.0E-7   205E-6

** Reactions linking 'FREE_SAN', 'SAND1' and 'SAND2'
...
```

In the Geomechanics data section is found:

```
** For matrix (rock)
*ELASTMOD  4.0E5    ** Young's modulus
*POISSRATIO 0.3      ** Poisson's ratio
*COHESION   15        ** Cohesion
*FRICANGLE  40        ** Friction angle

** For original sand:
*GEOSOLID 'SAND1'
 *SDELASTMOD 4.0E6    ** Young's modulus
 *SDPOISSON  0.1       ** Poisson's ratio

** For settled-out sand:
*GEOSOLID 'SAND2'
 *SDELASTMOD 1.0E7    ** Young's modulus
 *SDPOISSON  0.2       ** Poisson's ratio
```

---

## **Yield Criterion**

**\*TRESCA, \*VONMISES, \*MOHRCOUL, \*DRUCKER**

### **PURPOSE:**

Assign yield criterion.

### **FORMAT:**

**\*MOHRCOUL | \*DRUCKER | \*TRESCA | \*VONMISES**

### **DEFINITIONS:**

#### **\*MOHRCOUL**

Mohr-Coulomb yield criterion is used.

#### **\*DRUCKER**

Drucker-Prager yield criterion is used.

#### **\*TRESCA**

Tresca yield criterion is used.

#### **\*VONMISES**

Von Mises yield criterion is used.

### **DEFAULTS:**

#### **\*MOHRCOUL**

### **CONDITIONS:**

Only one of the yield criteria may be in use, i.e. they are mutually exclusive.

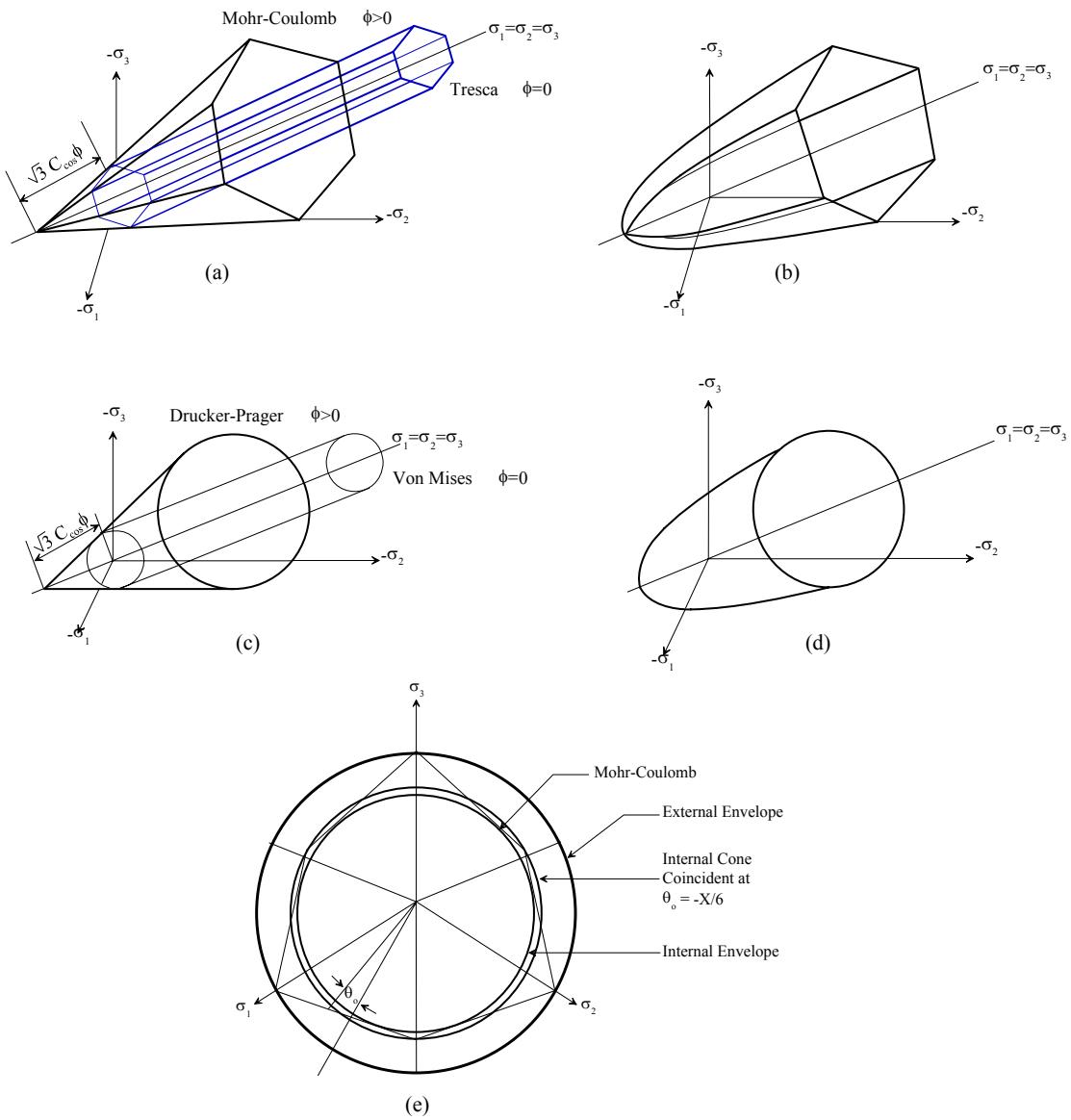
### **EXPLANATION:**

A pictorial view of these yield criteria in the three dimensional stress space are shown in Figure below.

These yield surfaces prescribe the stress states below which the material will behave elastically. As the material strain hardens, the yield surface expands as a function of the accumulated plastic strains. Presently, the model assumes isotropic hardening where the expansion of the yield surface is symmetric about the axis of the yield cone.

The Mohr-Coulomb and the Drucker-Prager yield criteria are popular criteria for frictional porous material. They are typically used for analysis of soils and rocks.

The Tresca and the Von Mises criteria are popular for metal plasticity and are available here mainly for testing.



*General and Linearized Yield Surfaces in Principal-Stress Space (from Desai & Christian, 1977)*

---

## Cap Model

\*GCAPMODEL

### PURPOSE:

Using cap model for an elasto-plastic constitutive model.

### FORMAT:

\*GCAPMODEL *nmodel*

### DEFINITION:

\*GCAPMODEL

Specify a cap model for a material.

*nmodel*

Indicate a specific cap model used in computation.

### CONDITIONS:

When the keyword \*GCAPMODEL appears, others keywords related to a cap model must be accompanied with it.

### EXPLANATION:

For soils, the Mohr-Coulomb criterion and Drucker-Prager criterion both suffer a deficiency which is the material can support an unlimited hydrostatic compression. This deficiency can be removed by adding a cap model which acts as a yield surface in the criteria. The cap model would allow consideration of compressive plastic volumetric strains and limit the amount of plastic dilation that occurred when loading on the Mohr-Coulomb or Drucker-Prager failure surface. The cap model is allowed to expand and contract along the hydrostatic axis. Details of a cap model will be given in the next section.

---

## **Cap Model 1**

\*GCAPLOC, \*GCAPD, \*GCAPW, \*GCAPR, \*GCAPTEN,  
\*GCAPMAT, \*ELASTMOD, \*POISSRATIO, \*COHESION, \*FRICANGLE, \*GCINCRMT

### **PURPOSE:**

Define coefficients for a cap model #1 as described in the keyword \*GCAPMODEL.

### **FORMAT:**

*GCAPLOC	$\kappa$
*GCAPD	$D$
*GCAPW	$W$
*GCAPR	$R$
*GCAPTEN	<i>tension</i>
*GCAPMAT	<i>type</i>
*ELASTMOD	<i>elastmod</i>
*POISSRATIO	<i>poissratio</i>
*COHESION	<i>cohesion</i>
*FRICANGLE	<i>fricangle</i>
*GCINCRMT	<i>segment</i>

### **DEFINITION:**

$\kappa$

Initial hardening parameter (kPa | psi | kPa)

$D$

Cap material constant (1/kPa|1/psi |1/kPa|)

$W$

Cap material constant (unitless)

$R$

Aspect ratio ( $\geq 0$ ) of the ellipse (ratio of horizontal to vertical axes)

*tension*

Tension cut-off limit strength (kPa | psi | kPa)

*type*

= 1 for soil whose hardening surface is allowed to move back toward the origin  
= 2 for rock whose hardening surface is not allowed to move back

*elastmod*

Young's elastic modulus (kPa | psi | kPa)

*poissratio*

Poisson's ratio

*cohesion*

Cohesion for Drucker-Prager model (kPa | psi | kPa)

*fricangle*

Friction angle for Drucker-Prager model (degrees)

*segment*

Number of divisions for strain

## DEFUALTS

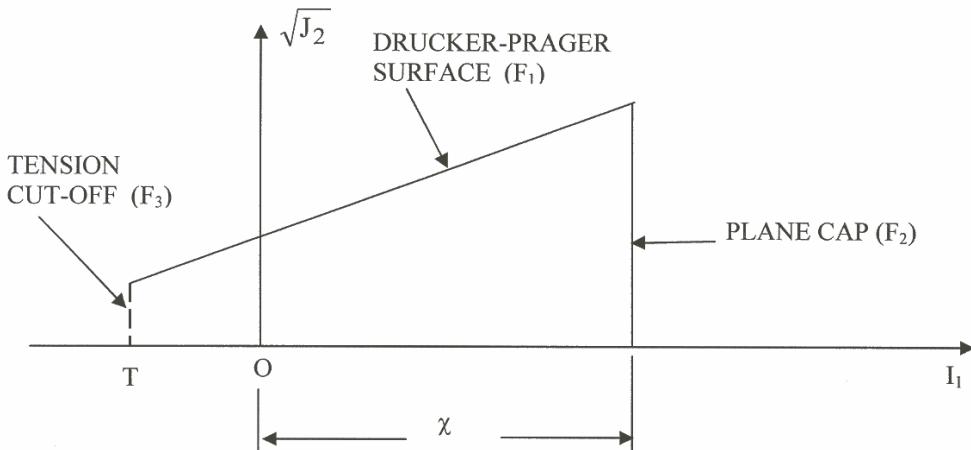
*COHESION	0
*FRICANGLE	30
*GCAPR	0
*GCAPTEN	0
*GCAPLOC	0
*GCAPMAT	2
*GCINCRMT	1

## CONDITION:

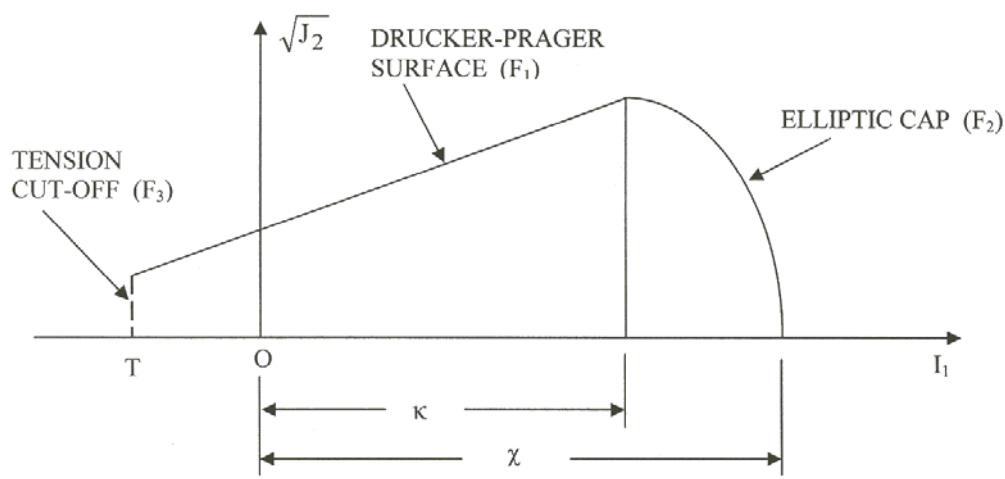
Accept the above default properties; other properties of the cap model must be given.

## EXPLANATION:

This cap model # 1 is only used for a perfectly plastic material and is coupled with the Drucker-Prager model. The cap surface can be a plane cap (Bath et al. and Sandler et al.) or an elliptic cap (Chen and Mizuno) depending on the value of shape ratio R. If R equals zero, the plane cap is used otherwise the elliptic cap is used.



Compressive Plane Cap Models in  $I_1$  and  $\sqrt{J_2}$  Plane



*Compressive Elliptic Cap Models in  $I_1$  and  $\sqrt{J_2}$  Plane*

Where:

$I_1$ : First invariance of effective stresses

$J_2$ : Second invariance of effective deviatoric stresses

As seen in the above figures, the failure surfaces are consisted of three different surfaces such as Drucker-Prager failure surface expressed by function  $F_1$ , cap failure surface expressed by function  $F_2$  and tension cut-off plane expressed by function  $F_3$ . It should be mentioned that this cap model is only applied to perfectly plastic materials and coupled to Drucker-Prager failure criterion. Also, compressive stress is positive and tensile stress is negative.

In the above figure, there are three failure surfaces as follows:

Drucker-Prager failure surface  $F_1$  is defined by:

$$F_1 = \alpha I_1 + \sqrt{J_2} - k = 0$$

Where  $\alpha$  and  $k$  are material parameters related to the cohesion and the friction angle.

Cap failure surface  $F_2$  is introduced by:

Compression plane cap surface:

$$F_2 = I_1 - \chi = 0$$

Compression elliptic cap surface:

$$F_2 = [I_1 - L_{(\kappa)}]^2 + R^2 J_2 - [\chi - L_{(\kappa)}]^2 = 0$$

Where:

$$L_{(\kappa)} = \kappa \text{ for } \kappa > 0 \text{ and } L_{(\kappa)} = 0 \text{ for } \kappa \leq 0$$

$\chi$  is the cap location dependent on the plastic volumetric strain  $\varepsilon_{kk}^p$  and is assumed to be:

$$\chi = \frac{1}{D} \ln \left( 1 + \frac{\varepsilon_{kk}^p}{W} \right)$$

D and W are cap material constants which are defined by keywords \*GCAPD and \*GCAPW, respectively.

$\kappa$  and R are the hardening parameter of the cap and aspect ratio of the ellipse as defined by the keywords \*GCAPLOC and \*GCAPR, respectively.

Tension cut-off limit plane  $F_3$  is given by:

$$F_3 = I_1 - T = 0$$

Details of this cap model # 1 and related equation can be reviewed in the references.

In addition, the keyword \*GCINCRMT is used to improve the accuracy and convergence. When strain is large, it can be divided into many segments (strain/segment) so that the stress return algorithm converges easily.

For example: In the geomechanical section of a data set, when a cap model is used.

*GCAPMODEL	1	** Cap model type 1
*ELASTMOD	40.3E+3	** Young modulus (kPa)
*POISSRATIO	0.2736	** Poisson ratio
*FRICANGLE	49.093	** Friction angle (degrees)
*COHESION	10.0	** Cohesion (kPa)
*GCAPR	0.0	** plane cap
*GCAPD	1.42E-6	** D value (1/kPa)
*GCAPW	0.0075	** W value
*GCAPTEN	-5.0	** Tension cut-off (kPa)
*GCAPLOC	100	** kPa, N/A when plane cap is used
*GCINCRMT	5	** Number of strain divisions

## References

Bath, K.J., Snyder, M.D., Cimento, A.P. and Rolph, W.D.: "On Some Current Procedures and Difficulties in Finite Element Analysis of Elastic-Plastic Response", Comput. Struct., Vol. 12, pp. 607-624, 1980.

Sandler, I.S., DiMaggio, F.L. and Baladi, G.Y.: "Generalized Cap Model for Geological Materials", J. Geotech. Eng. ASCE, Vol. 102 (GT7), pp. 683-699, 1976.

Chen, W.F. and Mizuno, E.: Nonlinear Analysis in Soil Mechanics: Theory and Implementation, Elsevier, 1990.

---

## Nonlinear Constitutive Model

\*NLINEAR

### PURPOSE:

Assign a nonlinear constitutive model for a type of rock.

### FORMAT:

\*NLINEAR *nmodel*

### DEFINITION:

\*NLINEAR

Specify a nonlinear constitutive model for a material.

*nmodel*

Indicate a specific constitutive model used in computation.

### CONDITIONS:

When the keyword \*NLINEAR appears, others keywords related to a constitutive model must be accompanied with it.

### EXPLANATION:

The material may have different aspects of behavior such as linear elasticity, elasto-plasticity, nonlinear elasticity or nonlinear plasticity. There are so many constitutive models which are used to express such nonlinear behavior of a material. However, only some of the most used models are selected to implement in the geomechanics module. Depending on the value of '*nmodel*' after the keyword \*NLINEAR, a type of constitutive model is defined. For instance,

\*NLINEAR 1 → Nonlinear elastic constitutive model 1.

\*NLINEAR 2 → Nonlinear elastic constitutive model 2.

Details of each nonlinear constitutive model will be explained in an appropriate section containing coefficients related to the model.

---

## **Nonlinear Elastic Constitutive Model 1**

\*ELASTMOD, \*POISSRATIO,  
\*GAMMA, \*GEXPONENTN, \*GULBULKMOD

### **PURPOSE:**

Define coefficients for a nonlinear elastic constitutive model # 1 as described in the keyword \*NLINER.

### **FORMAT:**

*ELASTMOD	<i>elastmod</i>
*POISSRATIO	<i>poissratio</i>
*GAMMA	<i>gamma</i>
*GEXPONENTN	<i>n</i>
*GULBULKMOD	<i>bulkmod</i>

### **DEFINITION:**

*elastmod*

Young's elastic modulus (kPa | psi)

*poissratio*

Poisson's ratio.

*gamma*

Coefficient multiplier (kPa | psi) used to determine the bulk modulus at a given mean effective stress.

*n*

Exponential power 'n' used to define nonlinearity of the bulk modulus.

*bulkmod*

Bulk modulus (kPa | psi)<sup>(1-n)</sup> used for unloading behavior.

### **DEFAULT**

If the keyword \*GULBULKMOD disappears, its value is zero.

### **CONDITION:**

Young's modulus \*ELASTMOD and Poisson's ratio \*POISSRATIO must be given.

### **EXPLANATION:**

In this constitutive model, the bulk modulus is defined as a function of mean effective stress under the loading condition. The bulk modulus at a certain mean effective stress can be written as:

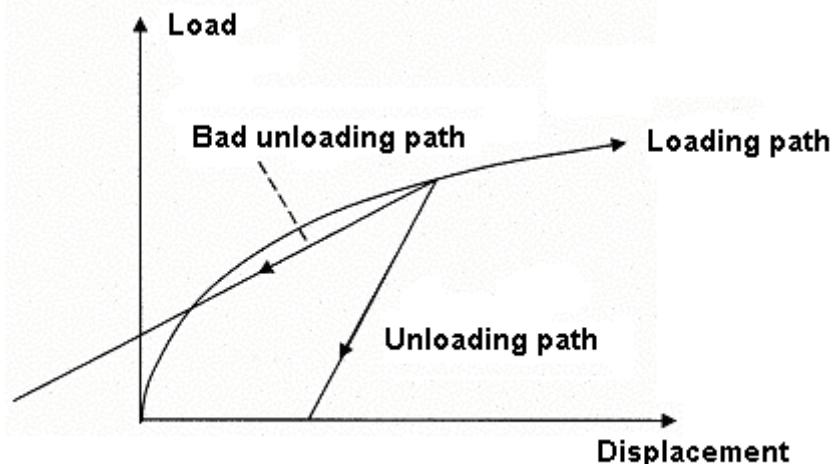
$$K = \text{gamma} * (\sigma'_m)^n$$

Where: the coefficient ‘gamma’ and power ‘n’ are defined above,  $\sigma'_m$  is the mean effective stress which is given by:

$$\sigma'_m = (\sigma'_{11} + \sigma'_{22} + \sigma'_{33})/3$$

In the above formula,  $\sigma'_{11}, \sigma'_{22}, \sigma'_{33}$  are effective stresses in X, Y, and Z directions respectively.

When unloading occurs, the bulk modulus is assumed to be constant and has a value as described by the keyword \*GULBULKMOD and its value (*bulkmod*). Care should be taken when assign a value of unloading bulk modulus so that the unloading path totally lies in the area bounded by the loading path and the displacement axis as shown. In other words when the unloading path is given, the value of *bulkmod* is selected carefully so that the unloading path will not cut the loading path. If the keyword \*GULBULKMOD is not entered for a rock type, the unloading path is coincident to the loading path. The loading path and unloading path are distinguished by the work done criteria.



### Example

```

*GEOMECH
*NLINEAR 1           **Keyword and number for the hypoelastic model
*ELASTMOD 3.95E5    **(kPa | psi)
*POISSRATIO 0.3      **unitless
*GAMMA 6125.0        **(kPa | psi)
*GEXPONENTN 0.4      **unitless
*GULBULKMOD 2.5E+4   **(kPa | psi)

```

---

**Nonlinear Elastic Constitutive Model 2**      \*ECOEF, \*BCOEF, \*GPATM,  
\*FRATIO, \*NE, \*NB, \*NTE, \*NTB. \*EXPN1, \*EXPN2, \*FRICANGMN, \*FRICANGMX,  
\*DFRICANGLE, \*URECOEF, \*URBCOEF, \*URNE, \*URNB, \*URNTE, \*URNTB, \*UREXPN1,  
\*UREXPN2, \*COHESION, \*FRICANGLE, \*MCOEF

**PURPOSE:**

Define coefficients for a hyperelastic constitutive model # 2 as described in the keyword  
\*NLINER.

**FORMAT:**

*ECOEF	<i>Ke</i>
*BCOEF	<i>Kb</i>
*GPATM	<i>Patm</i>
*FRATIO	<i>Rf</i>
*NE	<i>ne</i>
*NB	<i>nb</i>
*NTE	<i>nte</i>
*NTB	<i>ntb</i>
*EXPN1	<i>n1</i>
*EXPN2	<i>n2</i>
*FRICANGMN	<i>phimin</i>
*FRICANGMX	<i>phimax</i>
*DFRICANGLE	<i>dphi</i>
*URECOEF	<i>Keur</i>
*URBCOEF	<i>Kbur</i>
*URNE	<i>neur</i>
*URNB	<i>nbur</i>
*URNTE	<i>nteur</i>
*URNTB	<i>ntbur</i>
*UREXPN1	<i>n1ur</i>
*UREXPN2	<i>n2ur</i>
*COHESION	<i>c</i>
*FRICANGLE	<i>phi</i>
*MCOEF	<i>m</i>

**DEFINITION:**

**Ke**

Dimensionless loading modulus number

**Kb**

Dimensionless loading bulk modulus number

**Patm**

The atmospheric pressure (kPa | psi)

Rf	Failure ratio which lies between zero and one
ne	Loading tangential modulus exponent for confining stress
nb	Loading bulk modulus exponent for confining stress
nte	Loading modulus exponent for temperature
ntb	Loading bulk modulus exponent for temperature
n1,n2	Loading tangential modulus exponents for failure ratio
phimin	Minimum friction angle (degrees)
phimax	Maximum friction angle (degrees)
dphi	Reduction in friction angle (degrees) for one log circle
Keur	Dimensionless unloading-reloading modulus number
Kbur	Dimensionless unloading-reloading bulk modulus number
neur	Unloading-reloading modulus exponent for confining stress
nbur	Unloading-reloading bulk modulus exponent for confining stress
nteur	Unloading-reloading modulus exponent for temperature
ntbur	Unloading-reloading bulk modulus exponent for temperature

n1ur,n2ur

Unloading-reloading modulus exponents for failure ratio

c

Cohesion for Mohr-Coulomb and Drucker-Prager materials (kPa | psi)

phi

Angle of internal friction for Mohr-Coulomb and Drucker-Prager materials (degrees) at a confining pressure of 1 atm.

m

Coefficient factor used to compute the reference stress. Its value is 0 or 1.

Note: the confining stress used here corresponds to the minimum principle stress.

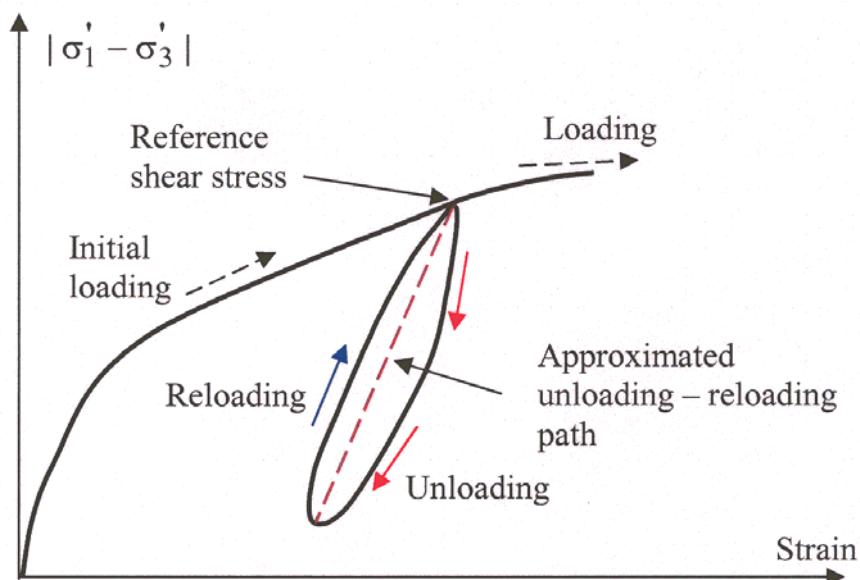
#### DEFAULT:

*GPATM	1 atm
*COHESION	0
*FRICANGLE	30
*NE, *NB	0.5
*URNE, *URNB	0.5
*EXP1, *UREXP1	2
*EXP2, *UREXP2	1
*NTE, *NTB	0
*URNTE, *URNTB	0
*FRICANGMN	0
*DFRICANGLE	0
*FRICANGLMX	30
*URECOEF	0
*URBCOEF	0
*MCOEF	1

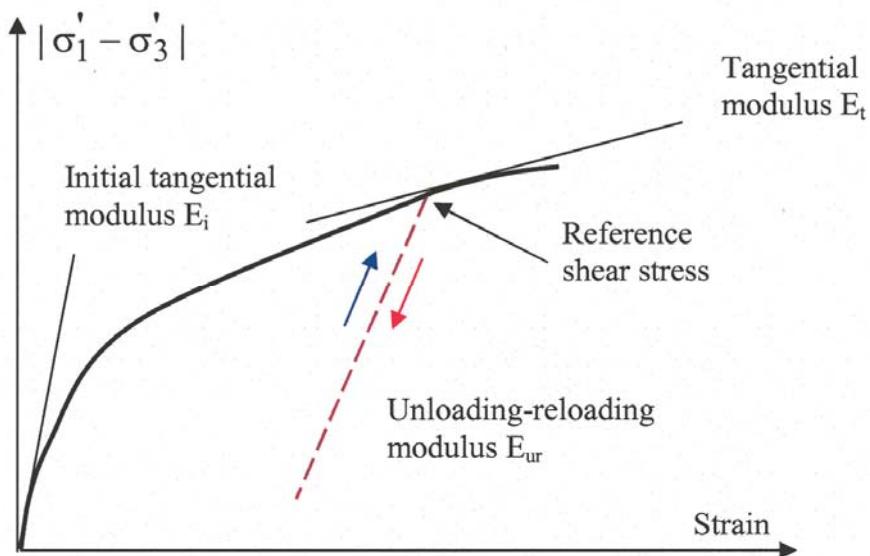
#### EXPLANATION:

The nonlinear elasticity constitutive model # 2 or hyperbolic model implemented in the module is based on the works of Duncan and Chang and Settari et al. In this model, tangential modulus as well as bulk modulus varies with minimum principle effective stress  $\sigma_3$  and temperature. Poisson's ratio in this model is, thus, also varied. The model also has a loading path and an unloading-reloading path which are distinguished by the reference shear stress criteria. When keywords \*URECOEF and \*URBCOEF disappear, there is no loading path. In this case, the unloading path coincides with the loading path. Furthermore, in this model, the unloading and reloading paths are the same.

In this hyperbolic model, the Mohr-Coulomb failure criterion is used for the material failure due to shear stress to compute the stress level. The stress level is also limited by one so that the shear stress can not exceed the shear failure of the Mohr-Coulomb model. More details on this model can be seen in related references.



The material follows a loading path when it is subjected to shear stress higher than it has previously experienced. Along this path, the constitutive model is governed by the tangential modulus. The material follows the unloading-reloading path when its shear stress is lower than the reference shear stress. In this case, the unloading-reloading modulus can either be affected by the shear stress level or not depending on the selection of the exponents related to the model.



In the above figure, there are three type of modulus:

Initial tangential modulus  $E_i$ : the slope of the tangential line passing the zero shear stress  $|\sigma'_1 - \sigma'_3| = 0$ . The initial tangential modulus  $E_i$  is computed by:

$$E_i = K_e P_{atm} \left( \frac{\sigma_r}{P_{atm}} \right)^{ne} \left( \frac{T}{T_o} \right)^{nte}$$

$$\sigma_r = \sigma'_3 + m \frac{c}{\tan(\phi_{imax})}$$

Where:

$c$ ,  $K_e$ ,  $P_{atm}$ ,  $\phi_{imax}$ ,  $ne$ ,  $nte$ ,  $m$  are defined above by the related keywords.

$T$  Current temperature

$T_o$  Initial temperature

$\sigma_r$  Reference stress

$\sigma'_3$  Minimum principle effective stress

When  $m = 0$ , the reference stress is the minimum principle effective stress which was proposed by Duncan and Chang.

Tangential modulus  $E_t$ :

$$E_t = E_i \left\{ 1 - \left[ R_f \frac{(\sigma'_1 - \sigma'_3)(1 - \sin(\phi))}{2ccos(\phi) + 2\sigma'_3 \sin(\phi)} \right]^{n2} \right\}^{n1}$$

$$\phi = \phi_{imax} - dphi * \log_{10} \left( \frac{\sigma'_3}{P_{atm}} \right)$$

Where:

$E_i(\sigma_r)$ : Initial tangential modulus defined in the above equation

$R_f$ : Failure ratio defined by keyword \*FRATIO

$L_\sigma$ : Stress level which is limited to one when the deviatoric effective stress reaches to the failure envelope. Its value is determined by:

$$L_\sigma = \frac{\sigma'_1 - \sigma'_3}{(\sigma'_1 - \sigma'_3)_f} \leq 1$$

Wherein, for sands, the failure envelope is normally defined by Mohr-Coulomb failure surface such as:

$$(\sigma'_d)_f = (\sigma'_1 - \sigma'_3)_f = \frac{2ccos(\phi) + 2\sigma'_3 \sin(\phi)}{1 - \sin(\phi)}$$

In the above equation, the friction angle  $\phi$  for a given  $\sigma'_3$  is given by:

$$\phi = \text{phi} - dphi * \log_{10} \left( \frac{\sigma'_3}{P_{\text{atm}}} \right)$$

and  $\phi$  is limited within the range:  $\text{phimax} \geq \phi \geq \text{phimin}$ .

Where:

$R_f$ ,  $c$ ,  $n_1$ ,  $n_2$ ,  $\text{phi}$ ,  $dphi$ ,  $\text{phimax}$ ,  $\text{phimin}$  are defined above.

If  $n_1 = 2$  and  $n_2 = 1$ , the above equation has the same form as that of the Duncan and Chang model.

Since the stress level  $L_\sigma$  is limited by one, the hyperelastic model used in this module considers the Mohr-Coulomb failure envelope as its asymptote.

#### Unloading-Reloading modulus $E_{ur}$

$$E_{uri} = K_{eur} P_{\text{atm}} \left( \frac{\sigma_r}{P_{\text{atm}}} \right)^{neur} \left( \frac{T}{T_o} \right)^{nteur}$$

$$E_{ur} = E_{uri} \left\{ 1 - \left[ R_f \frac{(\sigma'_1 - \sigma'_3)(1 - \sin(\phi))}{2c\cos(\phi) + 2\sigma'_3 \sin(\phi)} \right]^{n2ur} \right\}^{n1ur}$$

Where:

$n1ur$  and  $n2ur$  are defined above.

If the value of  $n2ur$  is set to be zero, the unloading-reloading modulus is unaffected by the shear stress level.

#### Loading bulk modulus $B_m$

$$B_m = K_b P_{\text{atm}} \left( \frac{\sigma_r}{P_{\text{atm}}} \right)^{nb} \left( \frac{T}{T_o} \right)^{ntb}$$

#### Unloading bulk modulus $B_{ur}$

$$B_{ur} = K_{bur} P_{\text{atm}} \left( \frac{\sigma_r}{P_{\text{atm}}} \right)^{nbur} \left( \frac{T}{T_o} \right)^{ntbur}$$

Poisson ratio  $\nu$  for loading or unloading-reloading case is computed by the equation:

$$\nu = \frac{3B - E}{6B}$$

Where:

B	Bulk modulus
E	Tangential modulus

The computed Poisson ratio should lie between the range:  $0.01 \leq \nu \leq 0.49$ .

Example: For a rock type which has a nonlinear elastic constitutive model 2, its parameters are given as follows:

```
*GEOMECH
*NLINEAR      2      ** keyword and number for the
                      hyperelastic model
*ECOEF        485.0   ** unitless
*BCOEF        404.2   ** unitless
*GATM         14.7    ** ambient pressure (psi)
*FRATIO        0.85   ** fraction ratio (unitless)
*NE            0.3    ** unitless
*NB            0.3    ** unitless
*NTE           0.0    ** unitless
*NTB           0.0    ** unitless
*EXPN1         2.0    ** unitless
*EXPN2         1.0    ** unitless
*FRICANGMN   1.0    ** degrees
*FRICANGMX   60     ** degrees
*DFRICANGLE   0.1    ** degrees
*URECOEF       700   ** unit less
*URBCOEF      583.3  ** unit less
*URNE          0.6    ** unitless
*URNB          0.6    ** unitless
*URNTE         0.0    ** unitless
*URNTB         0.0    ** unitless
*UREXPN1       2.0    ** unitless
*UREXPN2       1.0    ** unitless
*COHESION      14.5   ** psi
*FRICANGLE    40     ** degrees
*MCOEF          0     ** unitless
```

## References

- Settari, A., Ito, Y., Fukushima, N. and Vaziri, H.: “*Geotechnical Aspects of Recovery Processes in Oil Sands*”, Can. Geotech. J., Vol. 30, pp. 22-33, 1993.
- Settari, A. and Walters, D.: “*Advances in Coupled Geomechanical and Reservoir Modeling With Application to Reservoir Compaction*”, SPE 51927, 1999.
- Duncan, J.M. and Chang, C-Y.: “*Nonlinear Analysis of Stress and Strain in Soils*”, ASCE Journal of the Soil Mechanics and Foundations Division, Vol. 96, pp. 1629-1653, 1970.

---

## Creep Model

\*GMCREEP

### PURPOSE:

Assign a creep constitutive model for a type of rock. Creep is defined as the time-dependent strain that happens in a material at constant stress.

### FORMAT:

\*GMCREEP *cmodel*

### DEFINITION:

\*GMCREEP *cmodel*

Specify a creep constitutive model for a material, and indicate which plastic yield surface to use via *cmodel*.

\*GMCREEP 1 → Drucker-Prager type

\*GMCREEP 2 → Mises type

### CONDITIONS:

Keyword \*GMCREEP must be followed by others keywords related to the plastic yield surface indicated by *cmodel*. See manual page “Creep Model 1,2”, below.

### EXPLANATION:

There are many constitutive models used to express creep in materials. In CMG’s geomechanics module, the elasto-viscoplastic constitutive model based on Perzyna’s theory is selected. The total strain rate is decomposed into elastic strain rate and viscoplastic strain rate as follows:

$$\dot{\epsilon} = \dot{\epsilon}_e + \dot{\epsilon}_{vp}$$

Here

$\dot{\epsilon}_e$

Elastic strain rate estimated by the elastic constitutive relation.

$\dot{\epsilon}_{vp}$

Viscoplastic strain rate based on Perzyna’s theory wherein the viscous response is estimated by a time-rate flow rule. The viscoplastic strain rate is expressed by:

$$\dot{\epsilon}_{vp} = \gamma \cdot \langle \Phi(F) \rangle \cdot \frac{\partial F}{\partial \sigma}$$

Here

$\gamma$

The fluidity parameter that defines the relative rate of viscoplastic strain.

$\langle\Phi(F)\rangle$

Flow function that determines the current magnitude of viscoplastic strain rate.

$$\langle\Phi(F)\rangle = \begin{cases} \Phi(F) & \text{For } F > 0 \\ 0 & \text{For } F \leq 0 \end{cases}$$

F

Plastic yield function.

---

## Creep Model 1, 2

\*VISPARA, \*VISFLOWR, \*VISPOWER, \*VISSCHEME,  
\*VISTIME, \*VISTEP, \*VISINIT, \*ELASTMOD, \*POISSRATIO, \*FRICANGLE, \*COHESION,  
\*YLDSTRESS

### PURPOSE:

Define coefficients for creep models \*GMCREEP 1 (Drucker-Prager yield surface) and \*GMCREEP 2 (Mises yield surface).

### FORMAT:

*VISPARA	<i>gamma</i>
*VISFLOWR	<i>function</i>
*VISPOWER	<i>delta</i>
*VISSCHEME	<i>scheme</i>
*VISTIME	<i>para</i>
*VISTEP	<i>increment</i>
*VISINIT	<i>initime</i>
*ELASTMOD	<i>elastmod</i>
*POISSRATIO	<i>poissratio</i>
*FRICANGLE	<i>fricangle</i>
*COHESION	<i>cohesion</i>
*YLDSTRESS	<i>yldstress</i>

### DEFINITION:

*gamma*

Fluidity parameter  $\gamma$  (1/days | 1/days | 1/mins).

*function*

Viscoplastic flow function type.

*delta*

Parameter  $\delta$  used in viscoplastic flow function.

*scheme*

Scheme implicitness.

*scheme* = 1      explicit  
*scheme* = 2      semi-implicit  
*scheme* = 3      implicit

*para*

Timestep increment parameter.

For explicit scheme     $0.01 < para < 0.15$

For implicit scheme     $0.01 < para < 10$

*increment*

Increase the value of timestep from the previous timestep.

$$\Delta t_{n+1} = \text{increment} * \Delta t_n$$

*initime*

Initial timestep length (day | day | min).

*elastmod*

Young's elastic modulus (kPa | psi | kPa).

*poissratio*

Poisson's ratio.

*cohesion*

Cohesion for **Drucker-Prager** materials (kPa | psi | kPa).

*fricangle*

Friction angle for **Drucker-Prager** materials (degrees).

*yldstress*

Yield stress for **von Mises** materials (kPa | psi).

**DEFUALTS:**

*COHESION	0
*FRICANGLE	30
*VISFLOWR	2
*VISPOWER	1.0
*VISSCHEME	3
*VISTIME	0.01
*VISSTEP	0.1
*VISINIT	1.0

**CONDITIONS:**

Young's modulus \*ELASTMOD, Poisson's ratio \*POISSRATIO and fluidity parameter \*VISCOPARA must be given.

For a von Mises material, the yield stress (\*YLDSTRESS) is used, whereas for a Drucker-Prager material cohesion (\*COHESION) and friction angle (\*FRICANGLE) must be used.

**EXPLANATION:**

This type of viscoplastic constitutive model has the following assumptions:

1. Viscoplastic flow occurs only when the yield function F is greater than a uniaxial yield stress  $F_0$ .
2. The viscoplastic strain rate depends upon the current stresses only.

Therefore the viscoplastic flow rule can be written as:

$$\dot{\varepsilon}_{vp} = \gamma \cdot \langle \Phi(F) \rangle \cdot \frac{\partial F}{\partial \sigma} \quad (1)$$

Where:

$\gamma$

Fluidity parameter controlling the plastic flow rate (\*VISPARA).

The flow function  $\Phi$  can be one of the forms:

*function = 1:*

$$\Phi(F) = \exp \left\{ \delta \left( \frac{F - F_0}{F_0} \right) \right\} - 1 \quad (2)$$

*function = 2:*

$$\Phi(F) = \left( \frac{F - F_0}{F_0} \right)^\delta \quad (3)$$

In a time interval  $\Delta t$  equation (1) can be expressed as:

$$\Delta \dot{\varepsilon}_{vp} = \Delta t_n \left\{ (1 - \eta) \dot{\varepsilon}_{vp}^n + \eta \dot{\varepsilon}_{vp}^{n+1} \right\} \quad (4)$$

For the explicit scheme (*scheme = 1*)  $\eta = 0$ . For the semi-implicit scheme (*scheme = 2*)  $\eta = 0.5$ . For the implicit scheme (*scheme = 3*)  $\eta = 1$ .

When a value of  $\eta$  is greater than or equal to 0.5, the time integration scheme (4) is unconditionally stable; whereas the scheme (4) is only conditionally stable when the value of  $\eta$  is less than 0.5. For the latter case, the timestep length must be controlled. The timestep length can be computed as:

$$\Delta t_n \leq para * \sqrt{\frac{\dot{\varepsilon}_v^n}{\left( \dot{\varepsilon}_v^n \right)_{vp}}}$$

where

para:

Time increment parameter *para* as defined by the keyword \*VISTIME.

$\dot{\varepsilon}_v$ :

Volumetric strain

$(\dot{\varepsilon}_v)_{vp}$ :

Volumetric strain rate of viscoplasticity.

The change in timestep length between two consecutive calculations is also limited by:

$$\Delta t_{n+1} \leq \text{Increment} * \Delta t_n$$

where

Increment:

the *increment* parameter as defined by the keyword \*VISSTEP.

Note that the timestep length computed in the viscoplastic model is completely different from the timestep size used for fluid flow in the reservoir simulator. However, the sum of timestep length in the geomechanics module for one loading can not exceed the flow timestep. If the sum exceeds the flow timestep, the steady state is not reached within the allowed time. A warning message in the file \*.geo will appear such as “*Steady state time is larger than the flow timestep*”. If such a case occurs, some parameters should be changed; for instance, fluidity parameter which mostly influents the plastic flow rate.

### Example

Creep model for a Drucker-Prager material in field units:

*GMCREEP	1	** For Drucker-Prager material
*ELASTMOD	1.5E5	** Young's modulus (psi)
*POISSRATIO	0.3	** Poisson's ratio
*FRICANGLE	15.0	** Friction angle (degrees)
*COHESION	14.5	** Cohesion (psi)
*VISPARA	1.0E-5	** Fluidity parameter (1/day)
*VISPOWER	0.01	** Parameter used in flow function
*VISFLOWR	2	** Flow function type
*VISSCHEME	1	** Explicit scheme
*VISTIME	0.01	** Time increment parameter
*VISSTEP	0.1	** Timestep increment
*VISINIT	0.5	** Initial timestep length (day)

### References

Perzyna, P.: “*Fundamental Problems in Viscoplasticity*”, Advances in Applied Mechanics, Academic, New York, Vol. 9, pp. 244-368, 1966.

Zienkiewicz, O.C. and Cormeau, I.C.:” *Visco-plasticity, Plasticity and Creep in Elastic Solids – A Unified Numerical Solution Approach*”, Int. Journal for Numerical Methods in Engineering, Vol. 8, pp. 821-845, 1974.

---

## Pseudo Dilation Model

\*PGDILA, \*PGPACT, \*PGPDMAX

### PURPOSE:

Use a pseudo dilation model to mimic the STARS \*DILATION option.

### FORMAT:

```
*PGDILA      pgdila      young1
*PGPACT      pgpact      young2
*PGPDMAX    pgpdmax    young3
```

### DEFINITIONS:

*pgdila*

Threshold pressure beyond which the material behavior changes from an elastic to a dilation state (kPa | psi).

*young1*

Young's modulus in a dilation state (kPa | psi).

*pgpact*

Threshold pressure below which the material behavior changes from elastic state to a recompaction state (kPa | psi).

*young2*

Young's modulus in a recompaction state (kPa | psi).

*pgpdmax*

Threshold pressure above which the material behavior changes from dilation state to an elastic state (kPa | psi).

*young3*

Young's modulus in an elastic state beyond the dilation state (kPa | psi).

### CONDITIONS:

These keywords are assigned to the current rock type specified via \*GEOROCK, or to rock type #1 if \*GEOROCK is absent.

To use the pseudo dilation model the minimum required data is \*PGDILA and \*PGPACT. Also required are \*ELASTMOD (Young's modulus), \*POISSRATIO (Poisson's ratio) and a very large value for \*YLDSTRESS (yield stress) so the material is always in elastic mode.

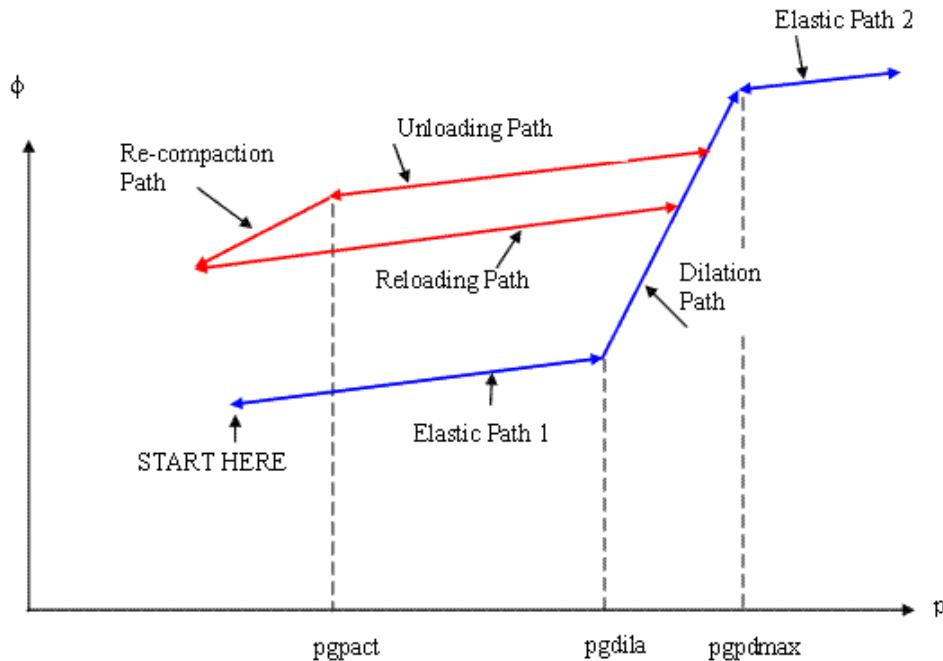
*pgpdmax* must be greater than *pgdila* which must be greater than *pgpact*.

### DEFAULTS:

If keyword \*PGPDMAX does not appear, *pgpdmax* has a very high value.

## EXPLANATION:

The pseudo dilation model is implemented in CMG's geomechanics module to produce a pressure-porosity curve similar to that of the STARS \*DILATION empirical model. In the pseudo dilation model, the change of Young's modulus is controlled by threshold pressures specified by \*PGDILA, \*PGPACT and \*PGPDMAX as shown in the below figure.



In the above figure, the six paths can be grouped as follows:

1. Elastic Path 1, Unloading Path, Reloading Path, Elastic Path 2: The material behavior is elastic so the load and deformation is reversible. A single Young's modulus specified via keyword \*ELASTMOD is used for all these paths except Elastic Path 2.
2. Dilation Path: Young's modulus will change to *young1* when the pressure is larger than *pgdila*. On this path the load and deformation cannot be reversed; if pressure decreases at some point, the material behavior starts down an Unloading Path instead.
3. Recompaction Path: If while on the unloading path the pressure continues decreasing until it is less than *pgpact*, the material behavior changes to a recompaction state with Young's modulus *young2*. On the Recompaction Path the load and deformation is not reversible. If the pressure increases then the material behavior will go along the Reloading Path instead.

## **Porosity Coupling**

With \*PGDILA there are two porosity coupling options available. In each case keywords \*PGDILA, \*PGPACT and \*PGPDMAX control the Young's modulus for each path. See the EXPLANATION for keyword \*GCOUPLING.

### **\*GCOUPLING 0**

Use this coupling option with \*PGDILA to obtain more accurate deformation and stress estimates while still getting the fluid-flow pressure-porosity behavior from the STARS \*DILATION model. In this coupling, porosity is a function of pressure and temperature only.

### **\*GCOUPLING 2**

This type of coupling is more compact than the previous one since the porosity is not only a function of pressure and temperature but also a function of mean total stress.

#### **Example**

```
*GEOROCK 1      ** Rock type # 1
*ELASTMOD 5e4   ** Young's modulus for elastic path 1
*POISSRATIO 0.3 ** Poisson's ratio
*YLDSTRESS 1e10  ** Very large yield stress
** dilation keyword    threshold p    Young's mod
*PGDILA          1536.0        1e4
*PGPACT          500.0         2e4
```

---

**Generalized Plasticity Model**      \*GENPLAST, \*ELASTMOD, \*POISSRATIO,  
\*PEAKCOHES, \*RESDUCOHES, \*EPSCUTOFF, \*COHESEXPN, \*FRICHARDEN,  
\*INITFANGLE, \*PEAKANGLE, \*CVFANGLE, \*COHESHARD, \*FRANGSOFT

**PURPOSE:**

Use the Generalized Plasticity constitutive model which allows cohesion, friction angle and dilation angle to change with effective plastic strain.

**FORMAT:**

*GENPLAST	<i>option</i>
*ELASTMOD	<i>elastmod</i>
*POISSRATIO	<i>poissratio</i>
*PEAKCOHES	$c_p$
*RESDUCOHES	$c_{res}$
*EPSCUTOFF	$\varepsilon_c$
*COHESEXPN	$n$
*FRICHARDEN	$a$
*INITFANGLE	$\phi_0$
*PEAKANGLE	$\phi_p$
*CVFANGLE	$\phi_{cv}$
*COHESHARD	
*FRANGSOFT	

**DEFINITION:**

**\*GENPLAST *option***

Specify that the Generalized Plasticity model along with the sub-option.  
Keywords \*ELASTMOD and \*POISSRATIO will be required.

<i>option</i>	Sub-option
1	Associated Drucker-Prager
2	Non-associated Drucker-Prager
3	Associated Mohr-Coulomb
4	Non-associated Mohr-Coulomb

**\*ELASTMOD *elastmod***

Young's modulus (kPa | psi | kPa).

**\*POISSRATIO *poissratio***

Poisson's ratio.

**\*PEAKCOHES  $c_p$**

Peak cohesion (kPa | psi | kPa).

**\*RESDUCOHES  $c_{res}$**

Residual value of cohesion (kPa | psi | kPa).

**\*EPSCUTOFF  $\varepsilon_c$**

Effective plastic strain cut-off for cohesive softening.

**\*COHESEXPN  $n$**

Exponent number for cohesive softening.

**\*FRICHARDEN  $a$**

Coefficient for friction hardening.

**\*INITFANGLE  $\varphi_0$**

Initial friction angle (degrees).

**\*PEAKANGLE  $\varphi_p$**

Maximum or peak friction angle (degrees).

**\*CVFANGLE  $\varphi_{cv}$**

Friction angle at constant volume (degrees).

**\*COHESHARD**

Condition for hardening the cohesion.

**\*FRANGSOFT**

Condition for softening the friction angle.

#### **DEFAULT:**

If \*PEAKCOHES is absent then  $c_p = 0$  is assumed.

If \*RESDUCOHES is absent then  $c_{res} = 0$  is assumed.

If \*EPSCUTOFF is absent then  $\varepsilon_c = 100$  is assumed.

If \*COHESEXPN is absent then  $n = 0$  is assumed.

If \*FRICHARDEN is absent then  $a = 0$  is assumed.

If \*INITFANGLE is absent then  $\varphi_0 = 30$  is assumed.

If \*PEAKANGLE is absent then  $\varphi_p = 30$  is assumed.

If \*CVFANGLE is absent the  $\varphi_{cv} = 30$  is assumed.

If \*COHESHARD is absent, softening the cohesion is assumed.

If \*FRANGSOFT is absent, hardening the friction angle is assumed.

#### **CONDITION:**

If \*GENPLAST appears then \*ELASTMOD and \*POISSRATIO are required.

## EXPLANATION:

Geo-materials normally exhibit pressure sensitivity and dilation behavior. These behavioral aspects can be described within the framework of non-associated isotropic hardening and softening and some convenient yield function which encapsulates all the basic ingredients of a granular material. The generalized plasticity model is based on either Mohr-Coulomb or Drucker-Prager yield surface. With this model, cohesion and friction angle are allowed changing with effective plastic strain according to the rules:

1. By default, cohesion decays exponentially with effective plastic strains until it reaches a residual value. In order to keep a constant cohesion, the values n of COHESEXPN keyword should be zero. When the keyword \*COHESHARD is used, the cohesion increases with the effective plastic strain until it reaches its maximum value via the keyword \*PEAKCOHES. In practice, the value of peak cohesion should not be zero.
2. By default, friction angle is increasing with effective plastic strains until it reaches a peak value. However, when the keyword \*FRANGSOFT is applied to a rock type, the friction angle starts decaying along with increasing the effective plastic strain from the peak angle to the value of  $\phi_0$ .
3. Dilation angle derived by Row is a function of friction angle. Therefore, when a non-associated plastic flow rule is used, the dilation angle also varies along with the change of effective plastic strain.

The physics behind the trend of decreasing cohesion with increasing effective plastic strains is based on the fact that there is usually a gradual loss of cementation as the material structure starts microcracking due to shearing. On the other hand, with increasing effective plastic strain, more friction is mobilized thus causing hardening towards a peak angle or an ultimate friction angle value. Hence, the dilation angle increases along with the friction angle.

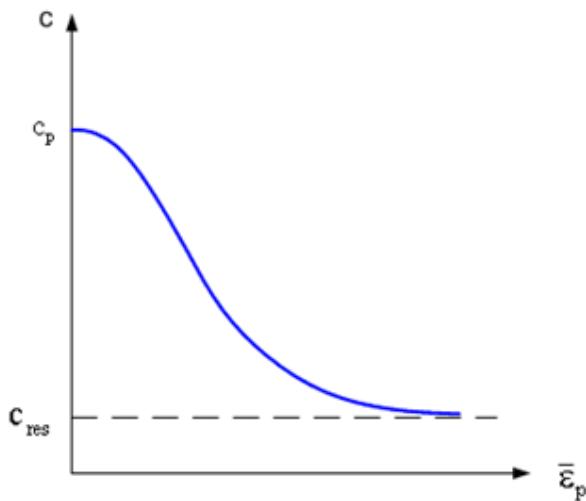
Softening and hardening cohesion may not be simultaneously combined in one material.

By default, cohesion, friction angle and dilation angle are expressed as:

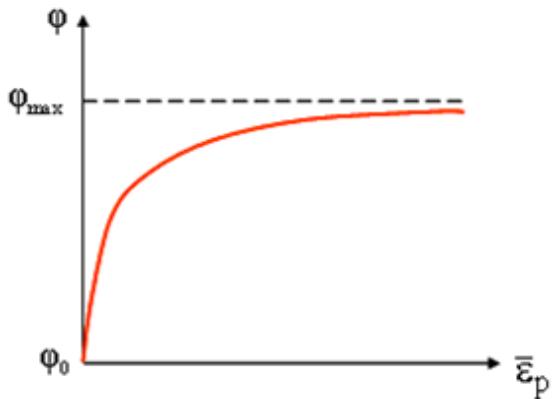
$$c = (c_p - c_{res}) \exp \left[ - \left( \frac{\bar{\varepsilon}_p}{\varepsilon_c} \right)^n \right] + c_{res}$$

$$\varphi = \sin^{-1} \left( 2 \sin(\varphi_p - \varphi_0) \frac{\sqrt{a \bar{\varepsilon}_p}}{a + \bar{\varepsilon}_p} \right) + \varphi_0$$

$$\sin \psi = \frac{\sin \varphi - \sin \varphi_{cv}}{1 - \sin \varphi \sin \varphi_{cv}}$$



*Nonlinear Softening Cohesion vs. Effective Plastic Strain*



*Hardening Friction Angle  $\phi$  Versus Effective Plastic Strain*

### Example 1

```

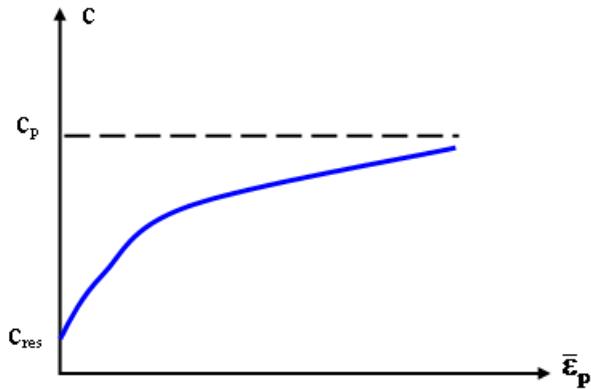
** Non-linear Cohesion Softening & Hardening Friction Angle
*GENPLAST 1    ** Generalized plasticity
                ** Drucker-Prager associated flow
*ELASTMOD 9.83e5      ** Young's modulus
*POISSRATIO 0.21       ** Poisson's ratio
*PEAKCOHES 500.38     ** Peak cohesion
*RESDUCOHES 0.0        ** Residual cohesion
*EPSCUTOFF 100.0       ** Effective plastic strain cut-off
*COHESEXPN 0.0         ** Exponent of cohesion softening
*FRICHARDEN 0.0        ** Coefficient for friction hardening
*INITFANGLE 30          ** Initial friction angle
*PEAKFANGLE 30          ** Peak friction angle
*CVFANGLE 0.0           ** Friction angle at constant volume

```

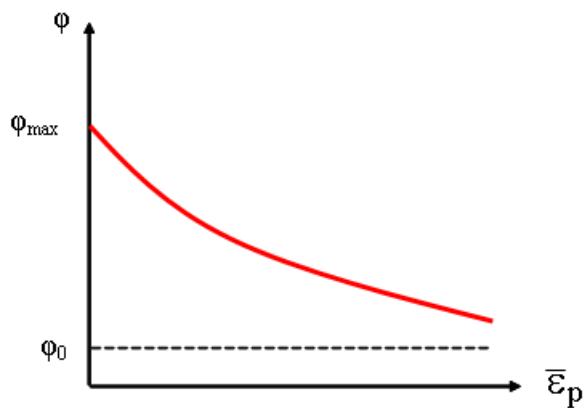
When keywords \*COHESHARD and \*FRANGSOFT appear, the cohesion and friction angle can be expressed as:

$$c = - (c_p - c_{res}) \exp \left[ - \left( \frac{\bar{\varepsilon}_p}{\varepsilon_c} \right)^n \right] + c_p$$

$$\varphi = - \sin^{-1} \left( 2 \sin(\varphi_p - \varphi_0) \frac{\sqrt{a \bar{\varepsilon}_p}}{a + \bar{\varepsilon}_p} \right) + \varphi_p$$



*Nonlinear Hardening Cohesion Versus Effective Plastic Strain*



*Softening Friction Angle  $\varphi$  Versus Effective Plastic Strain*

## Example 2

```
** Non-linear Cohesion Softening & Hardening Friction Angle
*GENPLAST 1    ** Generalized plasticity
               ** Drucker-Prager associated flow
*ELASTMOD 1.00e5      ** Young's modulus (psi)
*POISSRATIO 0.30      ** Poisson's ratio
*PEAKCOHES 14.5       ** Peak cohesion (psi)
*RESDUCOHES 0.0        ** Residual cohesion (psi)
*EPSCUTOFF 1E-02       ** Effective plastic strain cut-off
*COHESEXPN 0.2         ** Exponent of cohesion softening
*FRICHARDEN 5E-02      ** Coefficient for friction hardening
*INITFANGLE 20          ** Initial friction angle (deg)
*PEAKFANGLE 40          ** Peak friction angle (deg)
*CVFANGLE 20.           ** Friction angle at constant volume (deg)
*COHESHARD             ** Hardening cohesion
*FRANGSOFT              ** Softening friction angle
```

## Reference

Vermee, P.A. and R. De Borst: "Non-Associated Plasticity for Soils, Concrete and Rock," Heron, Vol 29(3), pp.5-64, 1984.

---

**Single Surface Failure Model**      \*SSMODEL, \*ELASTMOD, \*POISSRATIO,  
\*SGAMMA, \*SBETA, \*SM, \*SN, \*SNG, \*SA1, \*SETHA1, \*SKAPA1, \*SKAPA2, \*STEN,  
\*SPATM

**PURPOSE:**

The Single Surface Failure constitutive model captures the main behaviors of geomaterials under complex states of stress.

**FORMAT:**

*SSMODEL	<i>option</i>
*ELASTMOD	<i>elastmod</i>
*POISSRATIO	<i>poissratio</i>
*SGAMMA	$\gamma$
*SBETA	$\beta$
*SM	$m$
*SN	$n$
*SNG	$g$
*SA1	$a_1$
*SETHA1	$\eta_1$
*SKAPA1	$\kappa_1$
*SKAPA2	$\kappa_2$
*STEN	$T$
*SPATM	$p_a$

**DEFINITION:**

**\*SSMODEL *option***

Specify the Single Surface failure model and sub-option. Keywords \*ELASTMOD and \*POISSRATIO will be required.

<i>option</i>	Sub-option
1	Associated plastic flow
2	Non-associated plastic flow

**\*ELASTMOD *elastmod***

Young's modulus (kPa | psi | kPa).

**\*POISSRATIO *poissratio***

Poisson's ratio.

**\*SGAMMA  $\gamma$**

Material parameter related to the yield surface.

**\*SBETA  $\beta$**

Material parameter used to control the shape of failure surface on the octahedral plane.

*SM <i>m</i>	Power of stress ratio
*SN <i>n</i>	Parameter used to control the phase change when the volume changes from compaction to dilation or vanishes.
*SNG <i>g</i>	Power used to control the shape of failure surface in a meridian plane.
*SA1 <i>a<sub>1</sub></i>	Parameter used to compute a hardening value
*SETHA1 <i>η<sub>1</sub></i>	Exponential parameter for effective plastic strain.
*SKAPA1 <i>κ<sub>1</sub></i>	Material constant used to compute a hardening value for a non-associated flow
*SKAPA2 <i>κ<sub>2</sub></i>	Material constant used to compute a hardening value for a non-associated flow
*STEN <i>T</i>	Tensile strength of rock (kPa   psi   kPa).
*SPATM <i>p<sub>a</sub></i>	Atmospheric pressure (kPa   psi   kPa).

**DEFAULT:**

- If \*SGAMMA is absent then  $\gamma = 1$  is assumed.
- If \*SBETA is absent then  $\beta = 0$  is assumed.
- If \*SM is absent then  $m = 0$  is assumed.
- If \*SN is absent then  $n = 0$  is assumed.
- If \*SNG is absent then  $g = 2$  is assumed.
- If \*SA1 is absent then  $a_1 = 0$  is assumed.
- If \*SETHA1 is absent then  $\eta_1 = 0$  is assumed.
- If \*SKAPA1 is absent then  $\kappa_1 = 0$  is assumed.
- If \*SKAPA2 is absent then  $\kappa_2 = 0$  is assumed.
- If \*STEN is absent then  $T = 0$  is assumed.
- If \*SPATM is absent then  $p_a = 14.7$  psi (101.3 kPa) is assumed.

**CONDITION:**

- If \*SSMODEL appears then \*ELASTMOD and \*POISSRATIO are required.

## EXPLANATION:

The Single Surface Failure model discussed here is based on the work of Desai<sup>1</sup>. This model, also called the hierarchical model, captures the main features of geomechanical behavior: isotropic and dynamic hardening, softening, and associated and non-associated plasticity under complex states of stress. In comparison with multi-surface models, the Single Surface model has many advantages, including:

1. No numerical instabilities caused by operation at the non-smooth intersection of two surfaces in a multi-surface model.
2. The model allows different strengths under different stress paths.
3. The model allows change in the shape and size of failure envelope, tension and cohesion of a material.

The yield function F of the model<sup>2</sup> is:

$$F = \frac{J_2}{p_a^2} - \left[ -\alpha \left( \frac{I_1 + T}{p_a} \right)^n + \gamma \left( \frac{I_1 + T}{p_a} \right)^g \right] (1 - \beta S_r)^m \quad (1)$$

The potential function Q is:

$$Q = \frac{J_2}{p_a^2} - \left[ -\alpha_Q \left( \frac{I_1 + T}{p_a} \right)^n + \gamma \left( \frac{I_1 + T}{p_a} \right)^g \right] (1 - \beta S_r)^m \quad (2)$$

Where:

Compressive stress is positive.

$J_2$ : The second invariant of deviatoric stress tensor

$I_1$  : The first invariant of stress tensor

$p_a$ : The atmospheric pressure

$S_r$  : Stress ratio

$$S_r = \frac{\sqrt{27}}{2} \frac{J_3}{(J_2)^{1.5}} \quad (3)$$

$J_3$ : The third invariant of deviatoric stress tensor

$\alpha$ : The hardening function in F

$$\alpha = \frac{a_1}{\xi^{\eta_1}} \quad (4)$$

The increments of  $\xi$  and  $\xi_w$  are computed by:

$$d\xi = \lambda \left( \frac{\partial Q}{\partial \sigma_{ij} \partial \sigma_{ij}} \right)^{1/2} \quad (5)$$

$$d\xi_v = \lambda \left( \frac{\partial Q}{\partial \sigma_{ii}} \right) / \sqrt{3} \quad (6)$$

$\alpha_Q$ : The hardening function in Q

$$\alpha_Q = \alpha + \kappa(\alpha_0 - \alpha)(1 - r_v) \quad (7)$$

$$\kappa = \kappa_1 + \kappa_2 S_r \quad (8)$$

$$r_v = \frac{\xi_v}{\xi} \quad (9)$$

$\lambda$ : A scalar also called plastic multiplier

If we let  $T = 0$ ,  $m = 0$ ,  $g = 2$  and  $\alpha = 0$ , Equation (1) becomes:

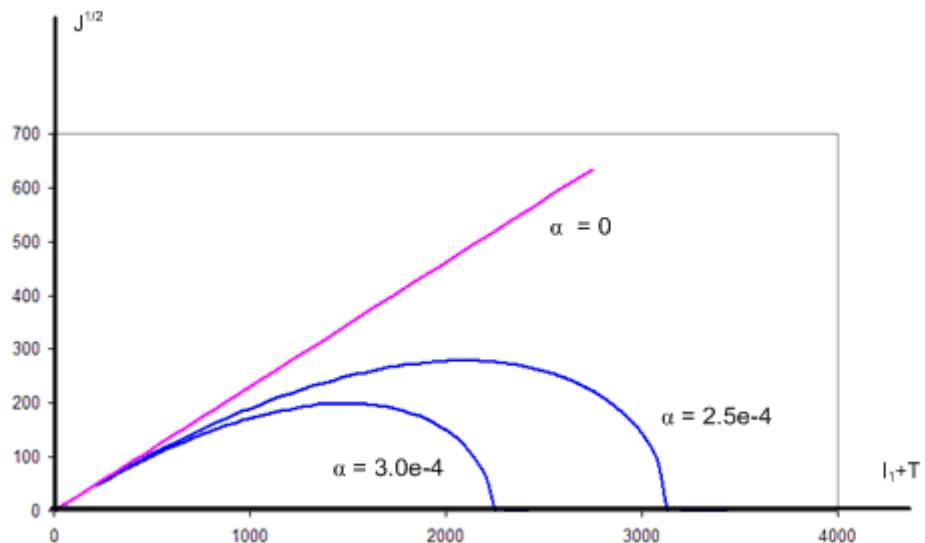
$$F = \frac{J_2}{p_a^2} - \gamma \left( \frac{I_1}{p_a} \right)^2 = 0 \quad (10)$$

Equation (10) is an approximation to the Drucker-Prager constitutive model when:

$$\sqrt{\gamma} \approx \frac{2 \sin \phi}{\sqrt{3}(3 - \sin \phi)} \quad (11)$$

Where  $\phi$  is friction angle. For some problems, the value of  $\gamma$  should be slightly adjusted to obtain equivalent results.

The following figure is an example to illustrate how the parameter  $\alpha$  changes the shape of failure envelope. Data used for plotting the curves are:  $g = 2$ ,  $\gamma = 0.0533$ ,  $n = 3$ ,  $m = 0$ ,  $T = 25.1$  psi and  $p_a = 14.7$  psi.



Other parameters such as  $g$ ,  $n$  and  $\beta$  can also change the shape of failure envelope. More details are discussed in the references 1 and 2.

When  $\alpha = 0$ , the failure envelope reaches the ultimate state.

### Example #1

This is an associated flow case in field units.

```
*SSMODEL 1          ** Associated flow
*ELASTMOD 1.5E4    ** Young's modulus (psi)
*POISSRATIO 0.3    ** Poisson's ratio
*SGAMMA 0.0533     ** Gamma
*SN   3.0           ** Power n
*SNG  2.0           ** Power g
*STEN -25.1         ** Tensile strength (psi)
*SPATM 14.7         ** Ambient pressure (psi)
```

### Example #2

This is a non-associated flow case in field units.

```
*SSMODEL 2          ** Non-associated flow
*ELASTMOD 1.5E4    ** Young's modulus (psi)
*POISSRATIO 0.3    ** Poisson's ratio
*SGAMMA 0.0533     ** Gamma
*SN   3.0           ** Power n
*SNG  2.0           ** Power g
*SKAPA1 0.05        ** Hardening parameter 1
*SKAPA2 -0.01       ** Hardening parameter 2
*STEN -25.1         ** Tensile strength (psi)
*SPATM 14.7         ** Ambient pressure (psi)
```

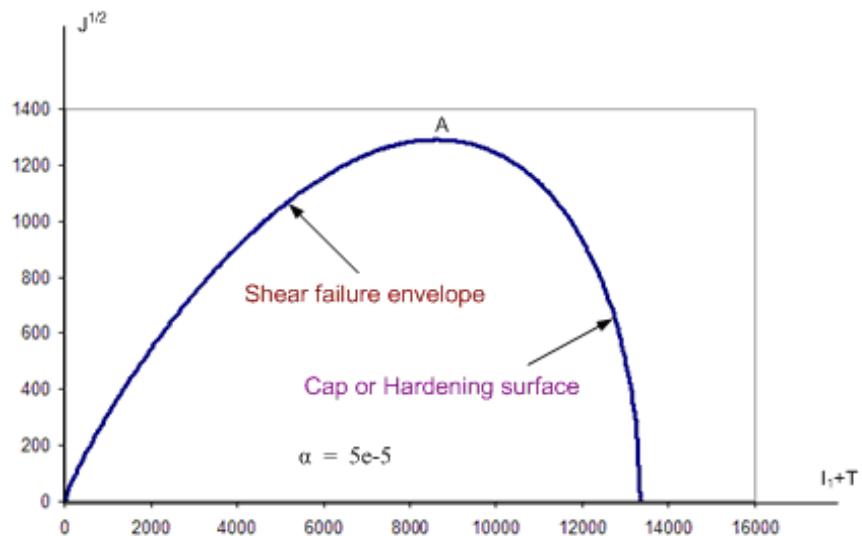
### Example #3

This is a hardening surface case in field units. This case is equivalent to a cap model because there is a dynamically hardening surface which encloses the shear failure envelope on the compressive stress axis.

```
*SSMODEL 1          ** Associated flow
*ELASTMOD 1.5E4    ** Young's modulus (psi)
*POISSRATIO 0.3    ** Poisson's ratio
*SGAMMA 0.35        ** Gamma
*SN   3.0           ** Power n
*SNG  1.7           ** Power g
*SA1   5E-5         ** Hardening parameter
*SETHA1 4E-1        ** Plastic strain parameter
*STEN -25.1         ** Tensile strength (psi)
*SPATM 14.7         ** Ambient pressure (psi)
```

For the purpose of representing a cap model situation, the key data of the Single Surface model is the plastic strain parameter \*SETHA1 which must be non-zero.

Using the above data with  $\alpha = 5e-5$  or  $\eta_1 = 0$ , the following single failure surface is plotted.



It should be noted that on the left of the maximum point A on the curve is the shear failure envelope and on the right of it is the cap or hardening surface.

When using the keyword \*YLDSTATE to illustrate failure modes of a material on RESULTS®, there are 3 states for the single surface failure model:

1. If its value is 0, the material is in an elastic state lying inside the curve.
2. If its value is 1, the material is in a plastic state lying on the shear failure envelope.
3. If its value is 2, the material is in a plastic state lying on the cap surface.

## References

Desai, S.C., : **Mechanics of Materials and Interfaces: The Disturbed State Concept**, CRC Press, 2001.

Liu, X. et al., : "Numerical modelling of nonlinear response of soil. Part 1: Constitutive model", **Int. J. Solids and Structures**, Vol. 42. pp. 1849-1881, 2005.

---

## **Modified Cam Clay Model**

\*MCCMODEL, \*SHEARMOD, \*POISSRATIO,  
\*FRICANGLE, \*MCSWINDX, \*MCOMINDX, \*MCRCSDL, \*MCPREHD, \*MCETHA, \*MCOCR

### **PURPOSE:**

The Modified Cam Clay (MCC) constitutive model captures the behavior of soft geomaterials such as soil or sand.

### **FORMAT:**

*MCCMODEL	<i>option</i>
*SHEARMOD	<i>shearmod</i>
*POISSRATIO	<i>poissratio</i>
*FRICANGLE	<i>angle</i>
*MCSWINDX	<i>swellindx</i>
*MCOMINDX	<i>comindx</i>
*MCRCSDL	<i>ratio</i>
*MCPREHD	<i>hard</i>
*MCETHA	<i>etha</i>
*MCOCR	<i>ocr</i>

### **DEFINITION:**

#### **\*MCCMODEL *option***

Specify the Modified Cam Clay failure model and sub-option.

<i>option</i>	Sub-option
1	Associated plastic flow
2	Non-associated plastic flow

#### **\*SHEARMOD *shearmod***

Shear modulus (kPa | psi | kPa).

#### **\*POISSRATIO *poissratio***

Poisson's ratio.

#### **\*FRICANGLE *angle***

Friction angle (degrees)

#### **\*MCSWINDX *swellindx***

Swelling index or gradient of swelling line

#### **\*MCOMINDX *comindx***

Compression index or gradient of virgin consolidation line (VCL)

#### **\*MCRCSDL *ratio***

Ratio of critical state lines in compression and tension

#### **\*MCPREHD *hard***

Initial hardening or preconsolidated pressure (kPa | psi | kPa)

**\*MCETHA  $\eta$** 

Factor for potential failure surface function – Nonassociated plastic flow

**\*MCOCR  $ocr$** 

Over consolidated ratio

**DEFAULT:**

If \*SHEARMOD is absent, shear modulus will be calculated at every timestep.

If \*FRICANGLE is absent then  $angle = 30$  degrees is assumed.

If \*MCRCSL is absent then  $ratio = 1$  is assumed.

If \*MCETHA is absent then  $\eta = 0$  is assumed.

If \*MCOCR is absent then  $ocr = 0$  is assumed.

**CONDITIONS:**

If \*MCCMODEL appears then \* MCSWINDX and \* MCOMINDX are required. Besides, other requirements are also needed for the model as follows:

If \*SHEARMOD appears, the Poisson's ratio is computed at every timestep.

If \*POISSRATIO appears, the shear modulus is computed at every timestep.

If \*MCPREHD appears, the initial hardening is maximum of the value of *hard* and the initial mean effective stress  $\sigma'_m$ .

If \*MCOCR appears, the value of *ocr* must be equal or greater than unity. The initial hardening is the maximum of (*hard* and  $ocr \cdot \sigma_0$ );  $\sigma_0$  is computed in the next section.

If \*MCOCR appears, the keyword \*MCPREHD may be absent. In this case, the initial hardening is equal to  $ocr \cdot \sigma_0$ .

The value of swelling index (*swellindx*) is always less than the value of compression index (*comindx*).

If \*MCETHA appears, the value of  $\eta$  must be equal or greater than zero.

**EXPLANATION:**

For soft geomaterials such as soil or sand, this modified cam clay model can be used for simulation. In this model, Young's modulus continues changing via the bulk modulus. Poisson's ratio can be changed or kept constant depending on the user's choice. If the Poisson's ratio is kept constant, shear modulus will change with time. In this case, the keyword \*SHEARMOD must be absent in the data. If the shear modulus is kept constant, Poisson's ratio is allowed to change with time. In this case, the keyword \*SHEARMOD must appear in the data and the keyword POISSRATIO can be absent. Even though the keyword POISSRATIO appears in the data, the keyword \*SHEARMOD will dominate.

The failure surface function of this model has a form as:

$$F(I, J, \alpha, \sigma_0) = I^2 + \left(\frac{J}{N}\right)^2 - 3\sigma_0 I = 0$$

Where:

$I$  : First invariant of effective stress tensor

$J$  : Square root of second invariant of effective stress tensor

$3\sigma_0$  : Intersection between the failure surface and I axis.

$$N := \frac{M}{3\sqrt{3}}$$

$$M := g(\text{ratio}, \theta) Mc = \frac{2\text{ratio}}{1 + \text{ratio} - (1 - \text{ratio})\sin 3\theta} Mc$$

$$\theta : \text{Lode's angle defined as: } -\frac{\pi}{6} \leq \theta \leq \frac{1}{3} \arcsin \left[ \frac{3\sqrt{3}}{2} \frac{J_3}{J^{3/2}} \right] \leq \frac{\pi}{6}$$

$$J_3 : \text{Third invariant of effective stress tensor} = \frac{1}{3} s_{ij} s_{jk} s_{ki}$$

$$s_{ij} : \text{Deviatoric stress} = \sigma'_{ij} - \frac{1}{3} I \delta_{ij}$$

$\sigma'_{ij}$  : Effective stress

$\delta_{ij}$  : Kronecker delta = 1 if  $i = j$  and = 0 if  $i \neq j$

$$Mc : \text{Slope of critical state line (CSL) in compression} = \frac{6 \sin \varphi}{2 - \sin \varphi}$$

$\varphi$  : Friction angle

The value of  $\sigma_0$  which is defined above can also be called as hardening. At the initial state, it has a value as.

If  $I \neq 0$ .

$$\sigma_0 = \max(\text{hard}, \frac{1}{3I} \left( I^2 + \left( \frac{J}{N} \right)^2 \right)) \quad \text{if *MCOCR is absent}$$

$$\sigma_0 = \max(\text{hard}, \frac{\text{ocr}}{3I} \left( I^2 + \left( \frac{J}{N} \right)^2 \right)) \quad \text{if *MCOCR appears}$$

If  $I = 0$ ,

$\sigma_0 = \text{hard}$

Mobilized value of  $\sigma_0$  can be calculated as:

$$\sigma_0^{n+1} = \sigma_0^n \exp \left( C_1 \left( \Delta \varepsilon_v - \frac{\Delta \sigma'_m}{K_n} \right) \right)$$

Where:

$\Delta\sigma'_m$  : Change in mean effective stress between timestep n+1 and n

$\Delta\varepsilon_v$  : Change in volumetric strain between timestep n+1 and n

$K_n$  : Bulk modulus at timestep n =  $\frac{(1+e_0)}{\text{swellindx}} \frac{I}{3}$

$e_0$  : Initial void ratio =  $\frac{\phi_0}{1-\phi_0}$

$\phi_0$  : Initial porosity

Shear modulus  $G_n$  at time n can be computed if Poisson's ratio  $v_0$  is given initially as follows:

$$G_n = \frac{K_n(3 - 6v_0)}{2(1 + v_0)}$$

If shear modulus  $G_0$  is given initially, Poisson's ratio at time n is computed by:

$$v_n = \frac{3K_n - 2G_0}{2G_0 + 6K_n}$$

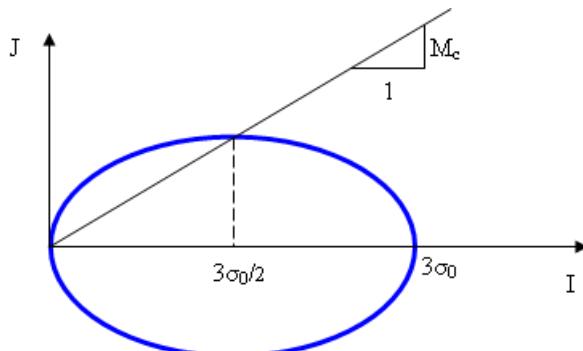
Young's modulus at time n is computed by:

$$E_n = 3K_n(1 - v_0) \quad \text{or} \quad E_n = 3K_n(1 - v_n)$$

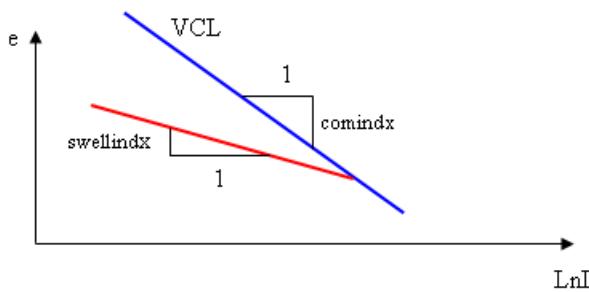
For non-associated plastic flow, the potential function has a form:

$$Q(I, J, \alpha, \sigma_0) = I^2 + \left(\frac{J}{N}\right)^2 - 3\eta\sigma_0 I = 0$$

Where:  $\eta$  via the keyword MCETHA is defined above.



*Yield Surface of Modified Cam Clay Model in IJ Plane*



In the Void Ratio  $e$  and  $\ln I$  Plane

The input of material properties for a modified cam clay model is given as follows:

Example 1: No shear modulus is used

```
*MCCMODEL 1      ** Associated flow
*MPOISSRATIO 0.3 ** Poisson's ratio
*MCSWINDX 0.018 ** Swelling index
*MCOMINDX 0.13  ** Compression index
*FRICANGLE 26.0  ** Friction angle (degrees)
*MCRCSSL 1.0    ** Ratio between tension and compression indices
*MCPREHD 20.0   ** Initial hardening (psi)
```

Example 2: With shear modulus and OCR are used. Non-associated plastic flow is applied.

```
*MCCMODEL 2      ** Non-associated flow
*SHEARMOD 2.0E5  ** Shear modulus (psi)
*MCSWINDX 0.018 ** Swelling index
*MCOMINDX 0.13  ** Compression index
*FRICANGLE 26.0  ** Friction angle (degrees)
*MCRCSSL 1.0    ** Ratio between tension and compression indices
*MCPREHD 20.0   ** Initial hardening (psi)
*MCOCR 3.0     ** Overconsolidated ratio
*MCETHA 0.0     ** Factor in non-associated flow potential
```

### References:

1. Hashash, Y.M.A. and Whittle, A.J. "Integration of the Modified Cam-Clay Model in Non-Linear Finite Element Analysis", **Computers and Geotechnics**, Vol 14, pp. 59-83, 1992.
2. Borja, R.I. and Tamagnini, C. "Cam-Clay Plasticity. Part III: extension of the Infinitesimal Model to Include Finite Strains", **Comput. Methods. Appl. Mech. Eng.**, Vol. 155, pp. 73-95, 1998.

---

## Thermal Expansion Coefficient

\*THEXPCOEF

### PURPOSE:

Specify the thermal expansion coefficient for a rock type due to the thermal effect.

### FORMAT:

\*THEXPCOEF    *thexpcoef*

### DEFINITION:

*thexpcoef*

Linear thermal expansion coefficient (1/C | 1/F | 1/C).

### CONDITION:

The keyword \*THEXPCOEF and its value must be entered when the thermal effect is taken into account in the geomechanical model. When the keyword is absent, changes in thermal do not affect to the rock at all.

### EXPLANATION:

When the temperature changes, the deformation as well as stress of rock also changes. Such changes strongly depend on the thermal expansion coefficient of a rock type. When the keyword \*THEXPCOEF is absent in the geomechanical section, the thermal expansion coefficient is zero by default; therefore, the source term due to the change in temperature is zero in the equilibrium equation. That means the temperature does not play a part in calculation of stresses via the rock deformation. The value of thermal expansion coefficient can be measured or taken from a material handbook.

### Example

```
*GEOROCK 1          ** rock type # 1
*ELASTMOD 1.0e6    *POISSRATIO .3   *YLDSTRESS 1000000
*THEXPCOEF 1.0e-6
```

In this example, the value of linear thermal expansion coefficient is  $10^{-6}$  (unit used) for rock type # 1.

---

## **Matrix Permeability Option**

**\*GPERMVL, \*GULOGINT**

**\*GPERMLC, \*GPERMES, \*GPERMTS,**

**PURPOSE:**

Compute permeability multiplier due to geomechanical responses.

**FORMAT:**

**\*GPERMLC  $C_{nl}$**

or

**\*GPERMES**

{  $\Delta\sigma_E$   $k_x/k_x^0$   $k_y/k_y^0$   $k_z/k_z^0$  }

or

**\*GPERMTS**

{  $\Delta\sigma_T$   $k_x/k_x^0$   $k_y/k_y^0$   $k_z/k_z^0$  }

or

**\*GPERMVL**

{  $\varepsilon_v$   $k_x/k_x^0$   $k_y/k_y^0$   $k_z/k_z^0$  }

**\*GULOGINT**

**DEFINITIONS:**

**\*GPERMLC  $C_{nl}$**

Use the empirical model of Li and Chalaturnyk to specify a matrix permeability multiplier, where  $C_{nl}$  (dimensionless) is an experimental parameter. Multiplier  $k/k^0 = \exp[ C_{nl} \cdot \varepsilon_v ]$  is applied to a block's permeability in all three directions, where  $\varepsilon_v$  is the volumetric strain.

**\*GPERMES**

Use a table to specify matrix permeability multiplier as a function of mean effective stress change from initial value  $\Delta\sigma_E$  (kPa | psi). The stress change entries must be monotonically increasing.

**\*GPERMTS**

Use a table to specify matrix permeability multiplier as a function of mean total stress change from initial value  $\Delta\sigma_T$  (kPa | psi). The stress change entries must be monotonically increasing.

**\*GPERMVL**

Use a table to specify matrix permeability multiplier as a function of volumetric strain  $\varepsilon_v$  (dimensionless). The strain entries must be monotonically increasing.

$k_x/k_x^0$   $k_y/k_y^0$   $k_z/k_z^0$

Permeability multipliers in the X, Y and Z directions, respectively (dimensionless). These multiplier entries must be monotonic. Multipliers can be different in the three grid directions.

#### \*GULOGINT

Use logarithmic interpolation for table look-up of the permeability-ratio columns. Let  $x$  be the independent variable ( $\varepsilon_v$ ,  $\Delta\sigma_E$  or  $\Delta\sigma_T$ ) and let  $y$  be the dependent variable ( $k_x/k_x^0$ ,  $k_y/k_y^0$  or  $k_z/k_z^0$ ); let subscripts 1 and 2 denote two adjacent table rows and let \* denote values between the table rows. The logarithmic interpolation algorithm is

$$[\log(y^*)-\log(y_1)]/[\log(y_2)-\log(y_1)] = [x^*-x_1]/[x_2-x_1]$$

whereas the linear interpolation algorithm is

$$[y^*-y_1]/[y_2-y_1] = [x^*-x_1]/[x_2-x_1]$$

#### DEFAULTS:

If none of these permeability multiplier options is specified for a rock type, then blocks in that rock type will experience no permeability change due to geomechanical effects.

If keyword \*GULOGINT is absent then table look-up used linear interpolation.

#### CONDITIONS:

These keywords are assigned to the current rock type specified via \*GEOROCK, or to rock type #1 if \*GEOROCK is absent.

#### EXPLANATION:

In order to include the geomechanical responses in the reservoir fluid flow, the permeability multiplier can depend on geomechanical information such as volumetric strain or mean stress. Two different approaches are available: empirical formula and three table look-up options.

#### Table Look-up

This approach is more flexible than the empirical formula since the permeability multipliers may differ in the three directions. The first column of the table can be either volumetric strain, mean effective stress change, or mean total stress change. The stress change is current stress minus initial stress on a per-node basis. The curves of multipliers ( $k_x/k_x^0$ ), ( $k_y/k_y^0$ ) and ( $k_z/k_z^0$ ) versus stress change or volumetric strain can be linear or nonlinear, but they must be monotonic. To ensure that permeabilities entered as data are used as initial values, the table must contain a row with  $k_x/k_x^0 = k_y/k_y^0 = k_z/k_z^0 = 1$  at  $\varepsilon_v$ ,  $\Delta\sigma_E$  or  $\Delta\sigma_T = 0$ .

## Example

```
*GEOROCK 1
  *GPERMLC 100 ** Empirical formula is used
*GEOROCK 2
  *GULOGINT    ** Logarithmic interpolation
  *GPERMES     ** Mean effective stress change
    ** mean eff
    ** stress dif   (kx/kx0) (ky/ky0) (kz/kz0)
      -600.        1.5      1.5      2.0
      -500.        1.4      1.4      1.8
      -200.        1.1      1.1      1.2
      0.          1.0      1.0      1.0  ** Initial
      300.        0.8      0.8      0.7
      500.        0.75     0.75     0.6
      800.        0.7      0.7      0.5

*GEOROCK 3
  *GPERMVL     ** Volumetric strain
    ** Vol strain   (kx/kx0) (ky/ky0) (kz/kz0)
      -0.005       1.5      1.5      1.6
      -0.001       1.4      1.4      1.5
      -0.0005      1.3      1.3      1.4
      -0.0001      1.2      1.2      1.3
      -0.00005     1.1      1.1      1.2
      -0.00001     1.05     1.05     1.02
      0.0          1.0      1.0      1.0  ** Initial
      0.0001      0.8      0.8      0.7
      0.0005      0.7      0.7      0.6
```

Note that positive stress means compression and negative volumetric strain means expansion.

### Empirical Formula

Based on the work of Li and Chalaturnyk (Li, P. and Chalaturnyk, R.J., “*Permeability Variations Associated with Shearing and Isotropic Unloading during the SAGD Process*”, CIPC Paper 2004-240), the following multiplier is applied to a block’s permeability in all three directions

$$k/k^0 = \exp[ C_{nl} \cdot \varepsilon_v ]$$

where

$C_{nl}$       experimental parameter for the block’s rock type, and  
 $\varepsilon_v$         volumetric strain of the block

---

## Barton-Bandis Fracture Permeability

### GFRACBLK

\*GPERMBB, \*UNFIXFDIR,

#### PURPOSE:

Use Barton-Bandis fracture permeability model.

#### FORMAT:

```
*GPERMBB e0 kni frs khf kccf krcf  
*UNFIXFDIR  
*GFRACBLK *IJK  
 { i1:i2 j1:j2 k1:k2 }
```

#### DEFINITIONS:

##### \*GPERMBB

Enable the Barton-Bandis fracture permeability model, for the current rock type specified by \*GEOROCK or its default.

*e<sub>0</sub>*

Initial fracture aperture (m | ft).

*kni*

Initial normal fracture stiffness (kPa/m | psi/ft).

*frs*

Fracture opening stress (kPa | psi).

*khf*

Hydraulic fracture permeability (md).

*kccf*

Fracture closure permeability (md).

*krcf*

Residual value of fracture closure permeability (md).

##### \*UNFIXFDIR

Allow the fracture direction to vary with time, for the current rock type specified by \*GEOROCK or its default.

##### \*GFRACBLK \*IJK

For the specified blocks, start fracture permeability on the Barton-Bandis model Path FG (see Figure below) instead of Path AB. This allows fracture permeability to use the “open fracture” model from the beginning without passing through the fracture opening path.

*i1:i2 j1:j2 k1:k2*

Range of grid indices in the I, J and K directions. Each index ranges from 1 to the maximum number of blocks in that direction as specified by keyword \*GRID. Each lower index must not exceed the corresponding upper index. Enter a single index as a range, for example, 5:5.

### DEFAULTS:

This option is disabled for each rock type for which keyword \*GPERMBB is not defined.

If \*UNFIXFDIR is absent, the fracture direction is fixed when the fracture opening occurs.

For each block not specified by \*GFRACBLK but using the \*GPERMBB option, fracture permeability starts on Path AB in below Figure.

### CONDITIONS:

After \*GPERMBB all six values must be entered and each value must be positive.

\*GPERMBB may be used only with natural fracture grid options \*DUALPERM and \*DUALPOR.

\*GPERMBB model parameters can be modified in recurrent data.

### EXPLANATION:

A natural-fracture grid option for fluid flow consists of the usual grid system for the porous rock matrix couple together with a second grid system consisting of fracture blocks that coexist with the matrix blocks on a one-to-one basis. Geomechanics calculations are coupled only to the matrix blocks. However, fracture opening and closing can depend upon stresses in the matrix blocks. Keyword \*GPERMBB allows for calculation of the fracture block permeability from normal fracture effective stress via the Barton-Bandis model.

Fracture closure permeability is computed by:

$$k_f = k_{ccf} \left( \frac{e}{e_0} \right)^4 \geq k_{rcf} \quad (1)$$

where:

$k_f$ : Fracture closure permeability (m | md)

$$e = e_0 - V_j \quad (2)$$

$$V_j = \frac{\sigma'_n}{k_{ni} + \sigma'_n / V_m} \quad (3) \text{ Joint closure under a normal fracture effective stress } \sigma'_n$$

$$V_m = e_0 \left[ 1 - \left( \frac{k_{rcf}}{k_{ccf}} \right)^{1/4} \right] \quad (4) \text{ Maximum fracture closure (m|ft)}$$

Fracture permeability depends on the value and history of normal fracture effective stress  $\sigma'_n$  as illustrated in below Figure. It should be noted that the normal fracture effective stress  $\sigma'_n$  is equivalent to the minimum principle effective stress.

Path AB: Initially  $\sigma'_n$  is greater than opening fracture stress  $f_{rs}$ . On this path initial fracture permeability is very small and behavior is reversible.

Path BC: When  $\sigma'_n$  becomes less than  $f_{rs}$ , the fracture opens suddenly and fracture permeability jumps from the initial value to hydraulic fracture permeability  $k_{hf}$ .

Path DCE: As long as  $\sigma'_n$  is less than zero, fracture permeability remains at  $k_{hf}$ .

Paths EF and FG: When  $\sigma'_n$  becomes greater than zero, fracture permeability jumps from  $k_{hf}$  to fracture closure permeability  $k_{ccf}$  and then follows the Barton-Bandis model (curve FG) specified by equation (1). Path FG has an asymptotic value at  $k_{rcf}$  as shown by the dotted line. Thereafter, fracture permeability varies reversibly with  $\sigma'_n$  on path GFED.

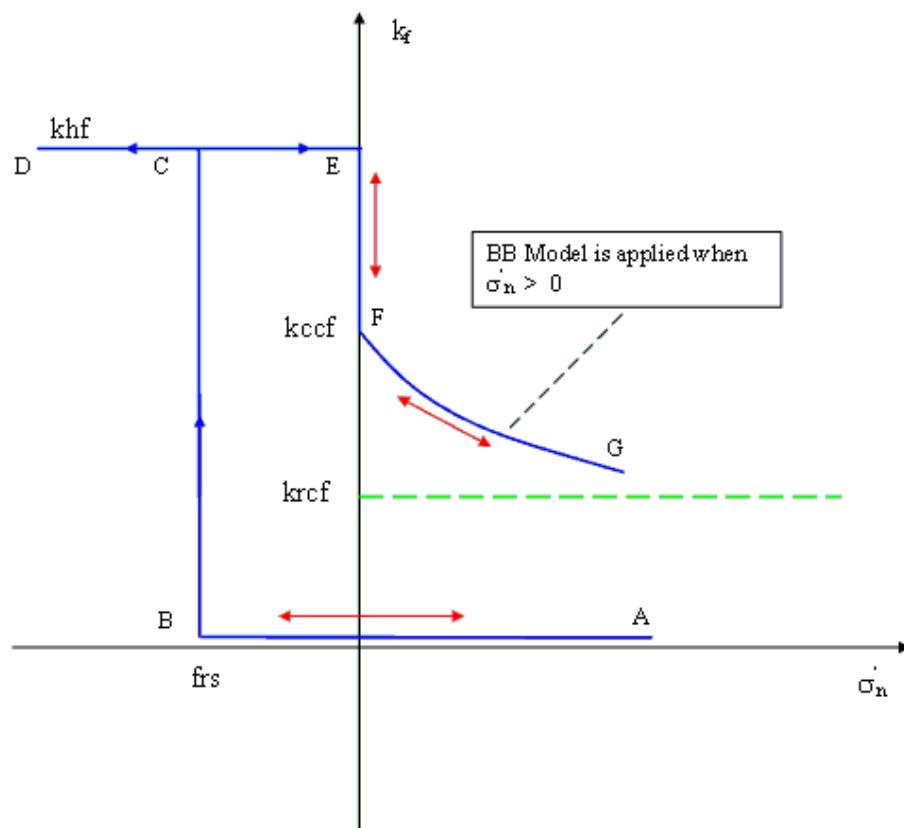


Diagram of Fracture Permeability

## Example

```
*GEOROCK 1  
.  
.  .  
** keyword e0      kni      frs      khf      kccf      krcf  
*GPERMBB 6.5e-5   3e6     -100.    100.    50.     5  
.  .  
*GEOROCK 2  
.  
.  .  
** keyword e0      kni      frs      khf      kccf      krcf  
*GPERMBB 7.5e-5   4e6     -110.    120.    60.     3  
*UNFIXFDIR  
.  .  
*GEOTYPE *KVAR 6*1 4*2 ** 10 layers  
*GFRACBLK *IJK  
  1:2 1:1 1:6 ** Fracture open from beginning  
  1:2 1:2 7:10
```

## Fracture Direction

\*FRACANGLE

### PURPOSE:

Specify direction of fracture when the fracture permeability option \*GPERMBB is used.

### FORMAT:

\*FRACANGLE

{ \*IJK ( iI(:i2)jI(:j2)kI(:k2) | \*ALL )  $\theta_x$   $\theta_y$  }

### DEFINITIONS:

iI(:i2)jI(:j2)kI(:k2)

I-J-K address range specification.

\*ALL

Specify all blocks.

$\theta_x$

Angle in degrees between fracture and I (X) coordinate direction. The allowed range is from 0 to 90 degrees.

$\theta_y$

Angle in degrees between fracture and J (Y) coordinate direction. The allowed range is from 0 to 90 degrees.

### DEFAULTS:

For each grid block that is not explicitly assigned a fracture direction via \*FRACANGLE, its fracture direction will be determined by the stress response on it.

### CONDITIONS:

Each pair of angles  $\theta_x$  and  $\theta_y$  must satisfy the condition  $\cos^2(\theta_x) + \cos^2(\theta_y) \leq 1$ .

### EXPLANATION:

When rock cracks the fracture direction depends strongly upon the direction of maximum principle effective stress acting on it. Use keyword \*FRACANGLE to assign explicitly the fracture direction independent of stress response in the block. When this option is used, the fracture direction will remain as specified, until it is replaced with a new fracture direction via keyword \*FRACANGLE in the recurrent data.

Multiple \*IJK lines can be used after \*FRACANGLE to specify a non-rectangular region or different angles for different regions.

### Examples

```
*FRACANGLE
  *IJK 1:10 5 1:2 60 40 ** Valid (sum cos2 = 0.837 < 1)
  *IJK 1:10 5 3:5 60 25 ** Invalid (sum cos2 = 1.071 > 1)
*FRACANGLE
  *IJK *ALL 60 60
```

## Dilation Relative Permeabilities

\*RPWTABD, \*RPLTABD

### PURPOSE:

Define relative permeabilities for the dilated region.

### FORMAT:

\*RPWTABD

sw	krw	krow
:	:	:

\*RPLTABD

sl	krg	krog
:	:	:

### DEFINITIONS:

sw

Water saturation. Table entries must increase, and must be separated by at least 0.001.

kr

Relative permeability to water in the water/oil system. Table entries must increase.

krow

Relative permeability to oil in the water/oil system. Table entries must decrease.

sl

Liquid saturation. Table entries must increase, and must be separated by at least 0.001.

krg

Relative permeability to gas in the liquid/gas system. Table entries must decrease.

krog

Relative permeability to liquid in the liquid/gas system. Table entries must increase. The last entry must be equal to krow at critical water saturation.

### DEFAULTS:

If the keywords \*RPWTABD and \*RPLTABD do not appear, the effect of remoulding on the relative permeability will not be taken into account.

## **CONDITIONS:**

Both tables are optional if the plastic deformation option is used. See the \*GEOMECH keyword.

These keywords are rock-type dependent, and are applied to the rock type number in effect at the time they are read. See the \*GEOROCK and \*GEOTYPE keywords.

## **EXPLANATION:**

The relative permeability curves for the dilation zone may be used to account for the remoulding of the sand matrix as a result of plastic deformation. A grid block which has been subjected to plastic yielding will be considered to consist of the remoulded zone and the original rock matrix with a relatively low level of disturbances. It may be reasonable to assume that the remoulded zone tends to have a more linear relative permeability relationship. However, to avoid stability problems the relative permeability increase should not exceed about 20 times that of the original rock matrix.

By using a set of relative permeability curves identical to that of the original rock, the effect of remoulding on relative permeability will be removed.

---

## Other Dilation Properties

\*FPVOLM, \*TDMAX, \*TDMIN

### PURPOSE:

Assign limits of pore volume and transmissibility changes due to volumetric dilatation which includes the shear dilation component.

### FORMAT:

*FPVOLM	<i>fpvolm</i>
*TDMAX	<i>tdmax</i>
*TDMIN	<i>tdmin</i>

### DEFINITIONS:

*fpvolm*

Maximum fractional change in the pore volume due to volumetric dilatation.

*tdmax*

Maximum transmissibility increases due to volumetric dilatation.

*tdmin*

Minimum transmissibility increases due to volumetric dilatation.

### DEFAULTS:

\*FPVOLM 0.05 \*TDMAX 1 \*TDMIN 1

### CONDITIONS:

*tdmin* must not exceed *tdmax*.

### EXPLANATION:

The volumetric dilatation of each grid block is calculated numerically by the finite-element stress-deformation analysis at each timestep. Keyword \*FPVOLM allows the user to limit the amount of volumetric dilatation.

Increase in transmissibility due to increase in the dilated zone is modelled via permeability multipliers of which *tdmin* and *tdmax* are the upper and lower bounds. Specifying \*TDMAX 1 and \*TDMIN 1 will result in no transmissibility increase. The relationship between transmissibility increase and pore-volume increase is as follows.

Let R be a block's fractional change in fluid pore volume due to volumetric dilatation, that is,  $R = (V - V_i)/V_i$  where  $V_i$  is the initial pore volume and V is the current pore volume; *fpvolm* is the maximum allowed value of R. Let  $x = R/fpvolm$  be the block's scaled fractional change in pore volume, limited to the range [0,1]. The block's permeability multiplier is

$$tdmin + (tdmax - tdmin)(3x^2 - 2x^3)$$

which gives  $tdmin$  at  $x = 0$ ,  $tdmax$  at  $x = 1$  and zero slopes at both points. The multiplier of each block pair is applied to their respective permeabilities which are combined to give the transmissibility of the inter-block connection.

Note that no transmissibility variation is applied to diagonal directions if the \*NINEPOINT grid option is used.

---

## Well Radius

\*WRADIUS

### PURPOSE:

Assign well radius where boundary conditions are to be applied for the axisymmetric radial grid. The desired model can be either the plasticity model or plasticity with well boundary unloading.

### FORMAT:

\*WRADIUS wrad

### DEFINITIONS:

wrad

Well radius (m | ft).

### DEFAULTS:

\*WRADIUS 0

### CONDITIONS:

Required only if the radial grid option is used.

### EXPLANATION:

The plastic deformation model may be used with axisymmetric radial grid. In this case the well boundary is considered to be a rigid boundary, for example, a cased wellbore. When the well boundary unloading model is selected by using the \*UNLOADSTR keyword, the boundary stress is assumed to be unloaded at the well radius by the amount specified using the \*UNLOADSTR keyword.

---

## **Stiffness Matrix Calculation Option**

**\*STIFFCOM1, \*STIFFCOM2**

**\*STIFFINIT, \*STIFFTANG,**

### **PURPOSE:**

Assign stiffness matrix calculation option.

### **FORMAT:**

**\*STIFFINIT | \*STIFFTANG | \*STIFFCOM1 | \*STIFFCOM2**

### **DEFINITIONS:**

#### **\*STIFFINIT**

Calculate the elasto-plastic stiffness matrix only once at the beginning of the timestep.

#### **\*STIFFTANG**

Calculate the elasto-plastic stiffness matrix at every iteration of every load increment of every timestep.

#### **\*STIFFCOM1**

Calculate the elasto-plastic stiffness matrix once at the beginning of every load increment of every timestep and also whenever unloading is detected.

#### **\*STIFFCOM2**

Calculate the elasto-plastic stiffness matrix at the beginning of the second iteration of each load increment of every timestep, except for the first increment when the stiffness matrix is also calculated at the first iteration.

Unloading will also trigger the recalculation of the stiffness matrix.

### **DEFAULTS:**

#### **\*STIFFTANG**

### **CONDITIONS:**

Only one of the stiffness matrix calculation option may be in use, i.e. they are mutually exclusive.

---

## **Deformation Solution Control**      \*PRINTGEO, \*NODE4, \*NODE8, \*NODE9, \*GAUSSPNT, \*NINCS, \*FORCETOL, \*DISPLACTOL, \*NITERGEO

### **PURPOSE:**

Override default control parameters for plastic deformation option.

### **FORMAT:**

\*NODE4 | \*NODE8 | \*NODE9

*GAUSSPNT	<i>ngaus</i>
*NINCS	<i>nincs</i>
*FORCETOL	<i>tol1</i>
*DISPLACTOL	<i>tol2</i>
*NITERGEO	<i>niter</i>
*PRINTGEO	<i>n</i>

### **DEFINITIONS:**

\*NODE4

Use 4-node finite elements.

\*NODE8

Use 8-node finite elements.

\*NODE9

Use 9-node finite elements.

\*GAUSSPNT *ngaus*

Specify the number of Gaussian quadrature points for each local coordinate direction. The allowed range is 2 to 4.

\*NINCS *nincs*

Subdivide the load increment for each timestep into *nincs* equal sub-increments. This will increase the accuracy of the plastic analysis at an increased computational cost. The allowed range for *nincs* is 1 to 10.

\*FORCETOL *tol1*

Relative convergence tolerance (expressed as a percentage) for force balance equations. The range is  $10^{-10}$  to 1. *tol1* is the rms residual of force balance at all nodes acceptable as converged to an equilibrium condition. The unbalanced force is carried forward into the next timestep. A value of 1 represents a rather loose tolerance whereas  $10^{-10}$  is very tight.

**\*DISPLACTOL *tol2***

Convergence tolerance (m | ft) for the nodal displacement vector. The allowed range is  $10^{-20}$  m to  $10^{-2}$  m. The choice should be representative of the dimension of the model and should not be bigger than one percent of the length of the smallest element. A large value effectively turns off the displacement convergence criterion.

**\*NITERGEO *niter***

Maximum number of iterations allowed for solving the force balance equations. The minimum is 30.

**\*PRINTGEO *n***

Control the amount of output to the geomechanics output (.geo) file. The minimum that is written is a summary of input data and finite element information. *n* may be any number from 0 to 13, to produce the indicated additional output. *n* = 0 - 3 produce the same output as previous versions.

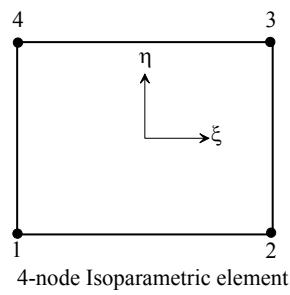
<i>n</i>	<i>Additional output</i>
0	None
1, 2, 3, 4, 9, 11	Principle stresses
1, 2, 3, 7, 8, 10	Stress components
2, 3, 5, 8, 9, 12	Displacements
3, 6, 10, 11, 12	Reactions at constrained nodes
13	Mapping reservoir grids and geomechanical grids

**DEFAULTS:**

\*NODE4  
\*GAUSSPNT 2  
\*NINCS 1  
\*FORCETOL 0.1  
\*DISPLACTOL 0.0001  
\*NITERGEO 30  
\*PRINTGEO 0

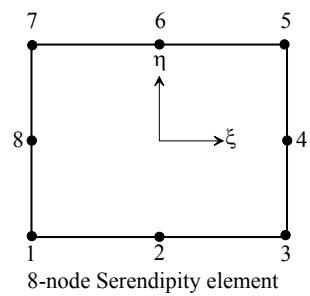
**EXPLANATION:**

The nodal locations for each of the element types \*NODE4, \*NODE8 and \*NODE9 and the associated element shape functions are shown in below Figure.



(a) 4-node element

$$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i)$$



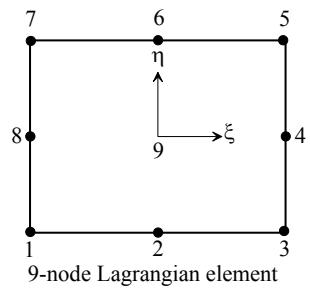
(b) 8-node element

for corner nodes

$$N_i^{(e)} = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i)(\xi\xi_i + \eta\eta_i - 1), \quad i = 1, 3, 5, 7$$

for midside nodes

$$N_i^{(e)} = \frac{\xi_i^2}{2}(1 + \xi\xi_i)(1 - \eta^2) + \frac{\eta_i^2}{2}(1 + \eta\eta_i)(1 - \xi^2), \quad i = 2, 4, 6, 8$$



(c) 9-node element

for corner nodes

$$N_i^{(e)} = \frac{1}{4}(\xi^2 + \xi\xi_i)(\eta^2 + \eta\eta_i), \quad i = 1, 3, 5, 7$$

for midside nodes

$$N_i^{(e)} = \frac{1}{2}\eta_i^2(\eta^2 - \eta\eta_i)(1 - \xi^2) + \frac{1}{2}\xi_i^2(\xi^2 - \xi\xi_i)(1 - \eta^2), \quad i = 2, 4, 6, 8$$

for central nodes

$$N_i^{(e)} = (1 - \xi^2)(1 - \eta^2)$$

### Element Types and Shape Functions Available for Geomechanical Analysis

---

## **Geomechanics AIMSOL Control**

\*SOLVERG, \*PRECCG, \*PRECABG,  
\*NORTHG, \*SORDERG, \*SDEGREEG, \*PIVOTG, \*ITERMAXG, \*ORTHOGG, \*JDUMPG,  
\*SITERPG

### **PURPOSE:**

Indicate and control AIMSOL as the matrix solver for the geomechanics equations.

### **FORMAT:**

*SOLVERG	( *AIMSOL   *AIMSOLN   *PGSOLV   *PGSOLN )
*PRECCG	<i>preccg</i>
*PRECABG	<i>precabs</i>
*NORTHG	<i>north</i>
*SORDERG	( *NATURAL   *REDBLACK   *RCM   *RCMRB )
*SDEGREEG	( *GAUSS   <i>ideg</i> )
*PIVOTG	( *ON   *OFF )
*ITERMAXG	<i>nitmax</i>
*ORTHOGG	( *CGS   *MGS ) ( *SINGLE   *DOUBLE )
*JDUMPG	
*SITERPG	

### **DEFINITIONS:**

#### **\*SOLVERG ( \*AIMSOL | \*AIMSOLN | \*PGSOLV | \*PGSOLN )**

Specify which matrix solver to use on the linearized geomechanics equations.

\*AIMSOL and \*AIMSOLN use the AIMSOL matrix solution package. This option is required for \*GEOM3D and recommended for large grids not using \*GEOM3D. The AIMSOL package implemented here has control keywords that are separate but similar (with "G" appended) to the corresponding fluid-flow keywords found in the Numerical Control section. This allows you to specify AIMSOL control parameters for solving geomechanics equations that differ from the corresponding fluid-flow values.

\*PGSOLV and \*PGSOLN use a frontal solution technique in which the system is solved non-iteratively in a way which is equivalent to Gaussian elimination.

When a non-associated flow rule is used (see \*DILANGLE) then \*AIMSOL must be used instead of \*AIMSOL and \*PGSOLN must be used instead of \*PGSOLV. See **Non-Associated Flow** below.

#### **\*PRECCG *preccg***

Specify relative convergence tolerance for GMRES iterations (AIMSOL only).

**\*PRECABG** *precabs*

Specify absolute convergence tolerance for the GMRES iterations (AIMSOL only).

**\*NORTHG** *north*

Specify maximum number of GMRES orthogonalizations before iteration restart (AIMSOL only).

**\*SORDERG** ( \*NATURAL | \*REDBLACK | \*RCM | \*RCMRB )

Specify equation ordering strategy (AIMSOL only). Natural, red-black, reverse Cuthill-McKee, and reverse Cuthill-McKee followed by Red-Black are supported.

**\*SDEGREEG** ( \*GAUSS | *ideg* )

Specify degree of LU factorization (AIMSOL only). \*GAUSS indicates Gaussian elimination; otherwise, off-diagonal sub-matrices of degree greater than *ideg* are ignored. The initial non-zero off-diagonal sub-matrices are designated as degree 1.

**\*PIVOTG** ( \*ON | \*OFF )

Specify that pivoting is allowed in the inversion of diagonal sub-matrices (AIMSOL only).

**\*ITERMAXG** *nitmax*

Specify maximum number of GMRES iterations allowed before returning to calling routine (AIMSOL only).

**\*ORTHOGG** ( \*CGS | \*MGS ) ( \*SINGLE | \*DOUBLE )

Specify orthogonalization method used in GMRES iterations: Classical (\*CGS) or modified Gram-Schmidt (\*MGS); and single or double orthogonalization (AIMSOL only).

**\*JDUMPG**

The geomechanical linear system matrix is dumped to the “.geo” geomechanical output file (AIMSOL only).

**\*SITERPG**

Information about GMRES dimensions and iterations is printed to the “.geo” output file (AIMSOL only; not for \*CHECKONLY). See the tutorial **Optimizing Memory Requirements**.

**DEFAULTS:**

Absent	Action
*SOLVERG	*AIMSOL (*GEOM3D present, *DILANGLE absent); *AIMSOLN (*GEOM3D present, *DILANGLE present); *PGSOLV (*GEOM3D absent, *DILANGLE absent); *PGSOLN (*GEOM3D absent, *DILANGLE present).
*PRECCG	$10^{-6}$
*PRECABG	$5 \cdot 10^{-9}$
*NORTHG	30
*SORDERG	*NATURAL
*SDEGREEG	2
*PIVOTG	*OFF
*ITERMAXG	30
*ORTHOOG	*CGS *SINGLE
*JDUMPNG	(no dumping)
*SITERPG	(no printing)

**CONDITIONS:**

All keywords except \*SOLVERG are valid only when \*AIMSOL or \*AIMSOLN is used.

\*SOLVERG subkeywords \*PGSOLV and \*PGSOLN may not be used with \*GEOM3D.

\*SOLVERG subkeyword \*AIMSOLN or \*PGSOLN must be used when \*DILANGLE is present.

**EXPLANATION:**

When a non-associated flow rule is used for an elasto-plastic material the stiffness matrix is unsymmetrical and requires an appropriate matrix solver. \*SOLVERG options \*PGSOLN and \*AIMSOLN handle these unsymmetrical matrices. Use of another matrix solver option for unsymmetrical matrices may not give the correct result. Like \*PGSOLV, \*PGSOLN is available only for the 2D plane strain option (\*GEOM3D absent). Like \*AIMSOL, \*AIMSOLN can handle 2D plane strain or 3D strain (\*GEOM3D).

\*PGSOLN and \*AIMSOLN can solve associated-flow problems as well as the other matrix solution options but the storage requirement is larger. Therefore, these matrix solution options should be used only for cases with non-associated flow, as indicated by the default.

---

## **Dimension Over-Rides (Optional)**      \*MPLNE, \*MCONNG, \*MDICLU\_PG

### **PURPOSE:**

Over-ride default dimension estimates obtained from the preliminary data scan.

### **FORMAT:**

*MPLNE	<i>mplne</i>
*MCONNG	<i>mconng</i>
*MDICLU_PG	<i>mdiclu_pg</i>

### **DEFINITIONS:**

*mplne*

Maximum dimension for geomechanics array IPFRE.

*mconng*

Maximum number of finite element connections for AIMSOL in Geomechanics.

*mdiclu\_pg*

Maximum number of AIMSOL LU submatrices in Geomechanics.

### **DEFAULTS:**

Each of these keywords defaults to the value obtained from the data scan.

### **EXPLANATION:**

Run-time dimensioning in STARS is designed to obtain all its needed information for storage allocation from a preliminary scan of the data. However, it is possible that several dimensioning parameters may be insufficient after this scan, in which case the user may enter values directly.

See also **Optimizing Memory Requirements** in the TUTORIAL chapter.

---

## **Initial Stress Distribution (2D)**

**\*STRESS, \*STRESSGRAD, \*STRESSALL,  
\*STRESI, \*STRESJ, \*STRESK, \*STRESSH**

### **PURPOSE:**

Assign the initial stress distribution for 2D plane strain approach.

### **FORMAT:**

*STRESS	<i>sigma_y sigma_z sigma_yz sigma_x</i>
*STRESSGRAD	<i>strgrd_y strgrd_z strgrd_yz strgrd_x</i>
or	
*STRESS *IJK	{ <i>il:i2 j1:j2 kl:k2</i> <i>sigma_y sigma_z sigma_yz sigma_x</i> }
or	
*STRESSALL	{ <i>sigma_y sigma_z sigma_yz sigma_x</i> } (N lines)
or	
*STRESI *ALL	<i>sigma_x</i> ... (N values)
*STRESJ *ALL	<i>sigma_y</i> ... (N values)
*STRESK *ALL	<i>sigma_z</i> ... (N values)
*STRESSH *ALL	<i>sigma_yz</i> ... (N values)

### **DEFINITIONS:**

*sigma\_y*      Y direction effective stress (kPa | psi).

*sigma\_z*      Z direction or vertical effective stress (kPa | psi).

*sigma\_yz*      YZ plane shear stress (kPa | psi).

*sigma\_x*      X direction effective stress (kPa | psi).

*strgrd\_y*      Y direction effective stress gradient (kPa/m | psi/ft).

*strgrd\_z*      Z direction effective stress gradient (kPa/m | psi/ft).

*strgrd\_yz*      YZ plane shear stress gradient (kPa/m | psi/ft).

*strgrd\_x*

X direction effective stress gradient (kPa/m | psi/ft).

*il:i2 j1:j2 k1:k2*

Specify a group of fundamental blocks with I-J-K ranges.

### DEFAULTS:

There is no default for initial stress.

### CONDITIONS:

Initial stress is required if the plastic deformation option is used. See \*GEOMECH.

### EXPLANATION:

There are a number of methods for specifying initial stress distributions in the reservoir.

#### Uniform Distribution

Use \*STRESS without \*IJK to specify an initially uniform stress state to the entire reservoir.

#### Linear Variation with Depth

Use \*STRESS without \*IJK together with \*STRESSGRAD to specify a linear variation of the individual stress components with depth. The values entered via \*STRESS are assumed to be located at the centre of the stress reference block. This reference block is (1,1,1) by default but another block may be specified via keyword \*GEORBLOCK. The initial Y-direction stress of a block is  $\sigma_y + (\text{strgrd}_y) \cdot \Delta h$ , where  $\Delta h$  is the depth of the block minus the depth of the reference block. Other stress components are computed in the same manner.

#### Uniform by Group

Use \*STRESS \*IJK to assign stresses by one or more groups of blocks. The syntax is similar to the \*IJK array input option described in chapter Keyword Data Entry System, except that each I-J-K range is followed by four values. Each grid block that is not referenced by the \*IJK ranges will have zero stress values.

#### Values for All Blocks, Stress Components Together

Use \*STRESSALL to assign all four stress component values together on a line. There must be one line for each of  $N = ni \cdot nj \cdot nk$  fundamental grid blocks as specified via keyword \*GRID \*CART or \*GRID \*RADIAL. The order of blocks is natural order, that is, starting at (1,1,1), increasing the I index most frequently and the K index least frequently. For example: (1,1,1), (2,1,1), ... (ni-1,nj,nk), (ni,nj,nk).

#### Values for All Blocks, Stress Components Separate

Use individual keywords \*STRESI (I-direction), \*STRESJ (J-direction), \*STRESK (K-direction) and \*STRESSH (shear) with \*ALL to assign the corresponding stress component to each grid block. The number and ordering of blocks are the same as described for \*STRESSALL. The syntax is similar to the \*ALL array input option.

## **Radial Grid**

The following associations exist between the radial and Cartesian stress components.

Radial	Y axis	sigma_y	*STRESJ *ALL
Axial (Z)	Z axis	sigma_z	*STRESK *ALL
Tangential	X axis	sigma_x	*STRESI *ALL
RZ plane	YZ plane	sigma_yz	*STRESSH *ALL

## **Initial Stress Distribution (3D)**    \*STRESS3D, \*STRESSGRAD3D, \*STRESI, \*STRESJ, \*STRESK, \*STRESSHIJ, \*STRESSHJK, \*STRESSHIK

## **PURPOSE:**

Assign the initial stress distribution for 3D finite elements.

## **FORMAT:**

\*STRESS3D  
 $\sigma_x \sigma_y \sigma_z \sigma_{xy} \sigma_{yz} \sigma_{xz}$   
 \*STRESSGRAD3D  
 $\text{strgrd}_x \text{strgrd}_y \text{strgrd}_z \text{strgrd}_{xy} \text{strgrd}_{yz} \text{strgrd}_{xz}$   
 or  
 \*STRESS3D  
 { \*IJK  $i_1:i_2 j_1:j_2 k_1:k_2$   
 $\sigma_x \sigma_y \sigma_z \sigma_{xy} \sigma_{yz} \sigma_{xz}$  }  
 or  
 \*STRESS3D \*ALL  
 {  $\sigma_x \sigma_y \sigma_z \sigma_{xy} \sigma_{yz} \sigma_{xz}$  } (N lines)  
 or  
 \*STRESS3D  
 \*STRESI \*ALL  $\sigma_{xy} \dots$  (N values)  
 \*STRESJ \*ALL  $\sigma_{xy} \dots$  (N values)  
 \*STRESK \*ALL  $\sigma_{xy} \dots$  (N values)  
 \*STRESSHIJ \*ALL  $\sigma_{xy} \dots$  (N values)  
 \*STRESSHJK \*ALL  $\sigma_{yz} \dots$  (N values)  
 \*STRESSHIK \*ALL  $\sigma_{xz} \dots$  (N values)

## **DEFINITIONS:**

*sigma\_x* X direction effective stress (kPa | psi).

*sigma\_y* Y direction effective stress (kPa | psi).

*sigma\_z* Z direction or vertical effective stress (kPa | psi).

*sigma\_xy* XY plane shear stress (kPa | psi).

*sigma\_yz* YZ plane shear stress (kPa | psi).

*sigma\_xz* XZ plane shear stress (kPa | psi)

*strgrd\_x*

X direction effective stress gradient (kPa/m | psi/ft).

*strgrd\_y*

Y direction effective stress gradient (kPa/m | psi/ft).

*strgrd\_z*

Z direction effective stress gradient (kPa/m | psi/ft).

*strgrd\_xy*

XY plane shear stress gradient (kPa/m | psi/ft).

*strgrd\_yz*

YZ plane shear stress gradient (kPa/m | psi/ft).

*strgrd\_xz*

XZ plane shear stress gradient (kPa/m | psi/ft).

*i1:i2 j1:j2 k1:k2*

Specify a group of fundamental blocks with I-J-K ranges.

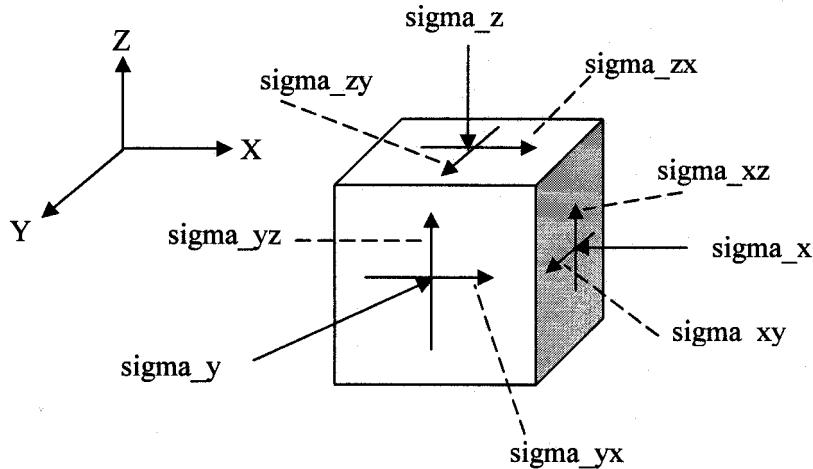
## DEFAULTS:

There is no default for initialize stress.

## CONDITIONS:

Initial stress is required if the plastic deformation option is used. See \*GEOMECH.

Keyword \*STRESS3D and subsequent values which indicate initial values of stresses on finite elements must be entered in a coupled data set. The convention of stresses on a block is shown in the figure.



In the above figure,  $\sigma_x$  means normal stress on a surface perpendicular to the X axis,  $\sigma_{xy}$  means that shear stress on a surface is perpendicular to the X axis and its traction direction is parallel to the Y axis. In addition, the shear stresses are symmetry, therefore:

$$\sigma_{yx} = \sigma_{xy}$$

$$\sigma_{zx} = \sigma_{xz}$$

$$\sigma_{zy} = \sigma_{yz}$$

### **EXPLANATION:**

There are a number of methods for specifying initial stress distributions in the reservoir.

#### **Uniform Distribution**

Use \*STRESS3D without \*IJK to specify an initially uniform stress state to the entire reservoir.

#### **Linear Variation with Depth**

Use \*STRESS3D without \*IJK but with \*STRESSGRAD3D to specify a linear variation of the individual stress components with depth. The values entered via \*STRESS3D are assumed to be located at the centre of the stress reference block. This reference block is (1,1,1) by default but another block may be specified via keyword \*GEORBLOCK. The initial Y-direction stress of a block is  $\sigma_y + (\text{strgrd}_y) \cdot \Delta h$ , where  $\Delta h$  is the depth of the block minus the depth of the reference block. Other stress components are computed in the same manner.

#### **Uniform by Group**

Use \*STRESS3D \*IJK to assign stresses by one or more groups of blocks. The syntax is similar to the \*IJK array input option described in chapter Keyword Data Entry System, except that each I-J-K range is followed by six values. Each grid block that is not referenced by the \*IJK ranges will have zero stress values.

#### **Values for All Blocks, Stress Components Together**

Use \*STRESS3D \*ALL to assign all six stress component values together on a line. There must be one line for each of  $N = ni \cdot nj \cdot nk$  fundamental grid blocks as specified via keyword \*GRID \*CART or \*GRID \*RADIAL. The order of blocks is natural order, that is, starting at (1,1,1), increasing the I index most frequently and the K index least frequently. For example: (1,1,1), (2,1,1), ... (ni-1,nj,nk), (ni,nj,nk).

#### **Values for All Blocks, Stress Components Separate**

Use \*STRESS3D followed by individual keywords \*STRESI (I-direction), \*STRESJ (J-direction), \*STRESK (K-direction), \*STRESSHIJ (XY plane shear), \*STRESSHJK (YZ plane shear) and \*STRESSHIK (XZ plane shear) with \*ALL to assign the corresponding stress component to each grid block. The number and ordering of blocks are the same as described for \*STRESS3D \*ALL. The syntax is similar to the \*ALL array input option.

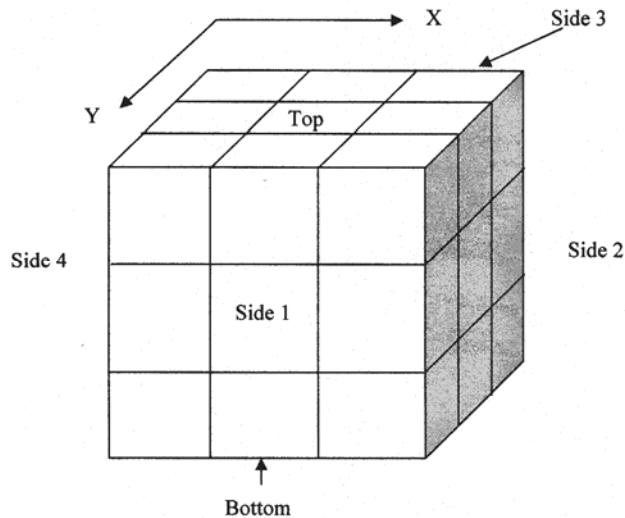
#### **Sign of Stress Gradient**

When the grid is defined with \*KDIR UP, stress gradients must be positive; for \*KDIR DOWN they must be negative.

### Example #1

Consider a reservoir having 3x3x3 grids as shown in the below figure. Assuming that normal effective stresses in three directions on the first layer of the reservoir are 5000 psi, stresses on the second layer are 5500 psi and stresses on the third layer are 6000 psi. The keywords entered in a data set are expressed as:

```
*STRESS3D
*IJK 1:3 1:3 1      ** layer 1
 5000 5000 5000    0 0 0
*IJK 1:3 1:3 2      ** layer 2
 5500 5500 5500    0 0 0
*IJK 1:3 1:3 3      ** layer 3
 6000 6000 6000    0 0 0
```



The above stresses can also be expressed as:

```
*STRESS3D *ALL
 5000 5000 5000 0 0 0      **Start first layer
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0
 5000 5000 5000 0 0 0      **End first layer
 5500 5500 5500 0 0 0      **Start second layer
 5500 5500 5500 0 0 0
 5500 5500 5500 0 0 0
 5500 5500 5500 0 0 0
 5500 5500 5500 0 0 0
 5500 5500 5500 0 0 0
 5500 5500 5500 0 0 0      **End second layer
```

```

6000 6000 6000 0 0 0      **Start third layer
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0
6000 6000 6000 0 0 0      **End third layer

```

The above example can also be expressible as:

```

*STRESS3D
*STRESI *ALL
5000 5000 5000 5000 5000 5000 5000 5000 5000
5500 5500 5500 5500 5500 5500 5500 5500 5500
6000 6000 6000 6000 6000 6000 6000 6000 6000
*STRESJ *ALL
5000 5000 5000 5000 5000 5000 5000 5000 5000
5500 5500 5500 5500 5500 5500 5500 5500 5500
6000 6000 6000 6000 6000 6000 6000 6000 6000
*STRESK *ALL
5000 5000 5000 5000 5000 5000 5000 5000 5000
5500 5500 5500 5500 5500 5500 5500 5500 5500
6000 6000 6000 6000 6000 6000 6000 6000 6000

```

### **Example #2**

If all the grid blocks have the same normal stresses, for instance 5000 psi in each direction.

The keyword is entered in the data set as:

```
*STRESS3D 5000 5000 5000 0 0 0
```

### **Example #3**

Assuming the stress gradient is 5 psi/ft in the normal directions. The keyword is entered as:

```

*KDIR *UP ...
*STRESSGRAD3D      5   5   5   0   0   0
** OR
*KDIR *DOWN ...
*STRESSGRAD3D     -5  -5  -5   0   0   0

```

Since shear stresses are zero the corresponding keywords are absent.

---

## Geomechanical Reference Block

\*GEORBLOCK

### PURPOSE:

Specify the reference block for stress gradient option.

### FORMAT:

```
*GEORBLOCK  
*IJK  i1  j1  k1
```

### DEFAULT:

If keyword \*GEORBLOCK is absent, the first grid block of the reservoir is used as a reference block from which stress gradients are computed.

### DEFINITION:

*i1*

Index of the reference block in X direction.

*j1*

Index of the reference block in Y direction.

*k1*

Index of the reference block in Z direction.

### EXPLANATION:

Keyword \*GEORBLOCK is designed so that the stress gradient can be determined when a reference grid block is selected among the reservoir grid blocks. Each direction index must lie within the fundamental grid indices defined by keyword \*GRID in the Reservoir Description chapter.

### Example

```
*STRESSGRAD  5  5  0  5  
*GEORBLOCK  
  *IJK    1    5    1    **Reference block is:  i=1, j=5, k=1
```

## Prescribed Boundary Conditions (2D)

\*PRESCBC

### PURPOSE:

Prescribe displacement-type boundary conditions on nodal points of a finite element.

### FORMAT:

\*PRESCBC  
{ node1(:node2) direction displacement }

or

\*PRESCBC  
{ \*PLANE p1(:p2)  
{ node1(:node2) direction displacement } }

or

\*PRESCBC  
{ \*IJK i1:i2 j1:j2 k1:k2  
{local node1(:local node2) direction displacement}}

### DEFINITIONS:

\*PRESCBC

Enable prescribed displacement boundary conditions. This keyword removes the default “essential” boundary conditions where nodes are constrained on the sides and bottom of the reservoir, and prepares for the user’s prescribed displacement-type boundary conditions table.

node1(:node2)

Single nodal point (or range of nodal points) of a finite element to which the boundary condition is assigned. For the case of \*KDIR \*DOWN (as shown in the figure below) the nodal points on finite elements are numbered from **top to bottom** and from left to right. In the case of \*KDIR \*UP, the nodal points are numbered from **bottom to top** and from left to right.

Note that an element node that is not associated with any active element (grid block) is not included in the node ordering. This occurs when a node lies in the interior of a region of null blocks. Since the distribution of null blocks can differ from plane to plane, node ordering and total nodes may differ also between planes.

\*IJK

Keyword for reservoir grid blocks

i1:i2

Starting block index *i1* to ending block index *i2* in X direction

*j1:j2*

Starting block index *j1* to ending block index *j2* in Y direction

*k1:k2*

Starting block index *k1* to ending block index *k2* in Z direction

*local node1(:local node2)*

Local nodal point (or a range of local nodal points) of a 2D finite element on which the prescribed boundary conditions are given. These local nodal points are strongly dependent on keywords \*KDIR UP, \*KDIR \*DOWN and \*PLSTRAINY. The convention of nodal points is given in the explanation.

*direction*

Direction of the displacement at a nodal point, given as follows:

- 01 displacement along the horizontal direction
- 02 displacement along the vertical direction
- 01:02 displacements in both directions

*displacement*

Value of the displacement (m | ft) at the indicated boundary nodal point(s) in the indicated direction(s).

\*PLANE *p1(:p2)*

Specify to which plane the following set of boundary conditions are to be applied. When this keyword is absent, the prescribed boundary conditions are applied to all planes.

Plane number *p1* (or each number *p1* to *p2*) corresponds to the grid index in the direction normal to the computed strain planes. There are two cases:

1. Keyword \*PLSTRAINY is absent: the computed strain planes are normal to the I (X) direction; *p1* is the I index and ranges from 1 to *ni* (see keyword \*GRID).
2. Keyword \*PLSTRAINY is present: the computed strain planes are normal to the J (Y) direction; *p1* is the J index and ranges from 1 to *nj* (see keyword \*GRID).

## CONDITIONS:

If the keyword \*PRESCBC is entered in a data set, prescribed boundary conditions must be given.

## DEFAULTS:

If keyword \*PRESCBC is absent, constraints on the left, right and bottom of a plane are used.

If sub-keyword \*PLANE is absent, all the planes have the same prescribed boundary conditions.

## EXPLANATION:

Keyword \*PRESCBC allows you to assign prescribed boundary conditions at any node on any finite element and on any plane which is cut from the reservoir either along the I or along the J direction.

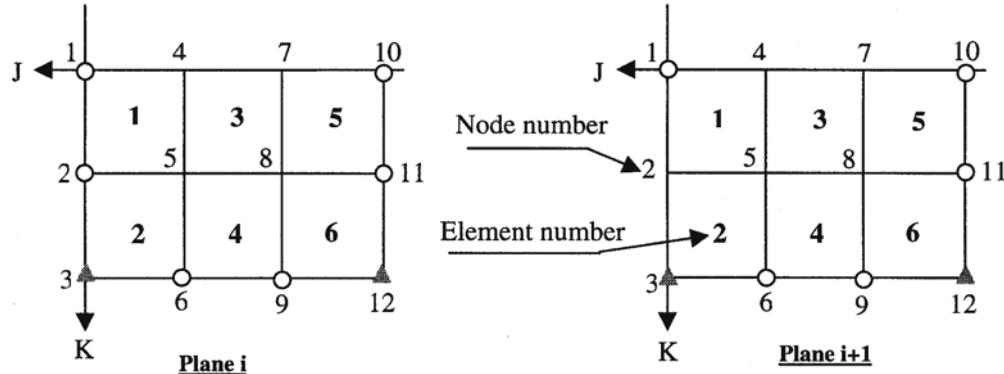
If keyword \*PRESCBC is present, prescribed boundary conditions must be supplied for appropriate nodes on each plane; otherwise the boundary nodes of finite elements are not constrained by any conditions. There are two approaches:

First approach: Using global nodal points.

In this approach, finite elements as well as their nodal points on a plane must be determined in advance before the prescribed boundary conditions can be assigned on those nodal points. This approach is quite uncomfortable when counting elements and node numbers for a reservoir having complicated geometry. This approach is used for 2D finite elements only.

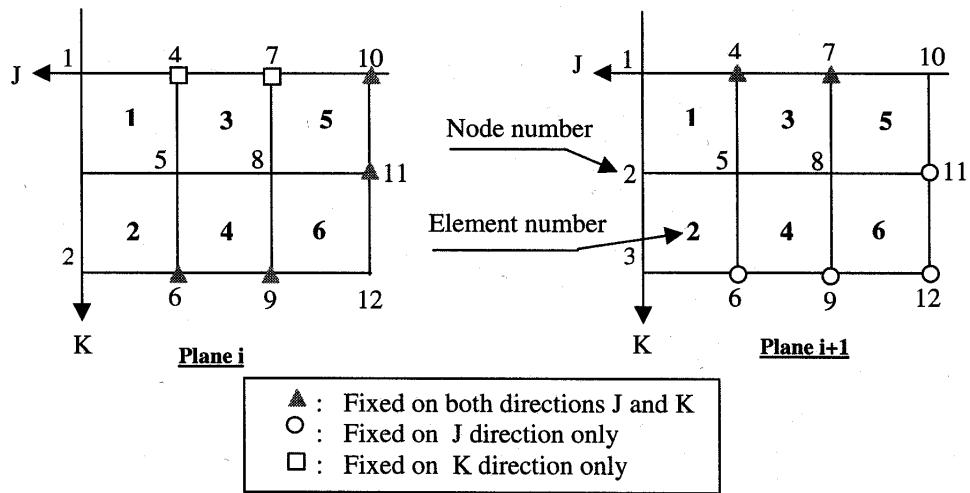
### Example

Consider two planes i and i+1. Without keyword \*PRESCBC, default boundary conditions are shown in the following figure.



With the following \*PRESCBC data, prescribed boundary conditions are given on plane i and plane i+1 as shown in the following figure.

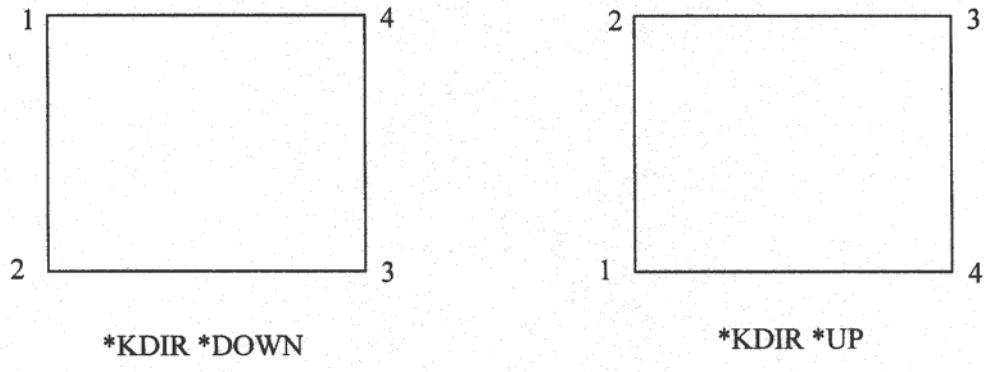
```
*PRESCBC
*PLANE    i
** Node      Direction   Displacement
        4          02           0.0
        6          01:02         0.0
        7          02           0.0
    9:11        01:02         0.0
*PLANE    i+1
** Node      Direction   Displacement
        4          01:02         0.0
        6          01           0.0
        7          01:02         0.0
        9          01           0.0
    11:12       01           0.0
```



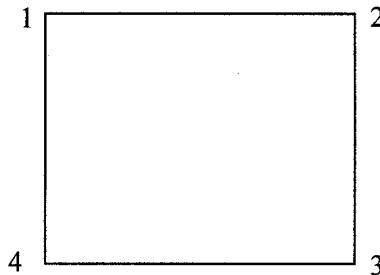
Second approach: Using local nodal points of a finite element.

This approach uses the reservoir block address in terms of I, J, K to assign the boundary conditions on local nodal points of a finite element. The approach is an easy and straightforward way compared to the first approach. In this approach, the user does not have to count the node and element number in advance. As mentioned above, numbering the local nodal points on a 2D finite element are strongly dependent on keywords \*KDIR \*UP, \*KDIR DOWN and \*PLSTRAINY. The nodal points for each case are illustrated in following figures.

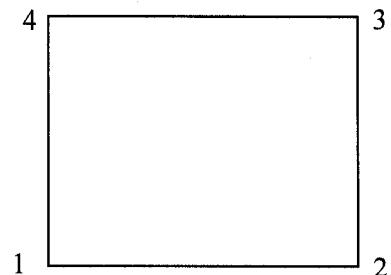
Without keyword \*PLSTRAINY



With keyword \*PLSTRAINY



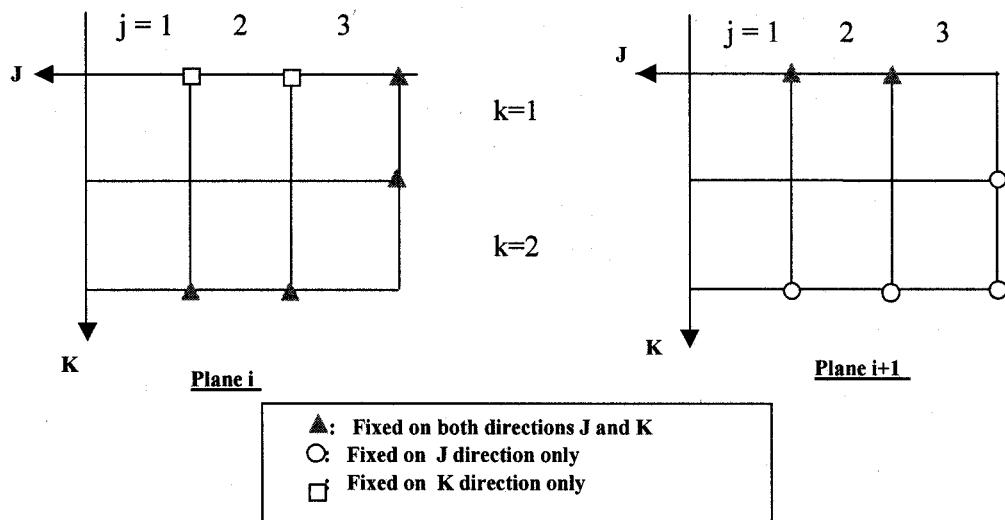
\*KDIR \*DOWN



\*KDIR \*UP

**It should be noted that when \*GRID \*RAD is used, the keyword \*PLSTRAINY is automatically applied. Therefore, the notation of local nodal points is shown above must be taken into account.**

Taking the above example, on plane i and plane  $i+1$ ,  $NJ = 3$  and  $NK = 2$ . Keyword \*KDIR DOWN is used and keyword \*PLSTRAINY is absent that means the plane is perpendicular to the I direction. The prescribed boundary conditions are expressed as:



```

*PRESCBC
** For plane i
*IJK    i    1    1    ** j = 1, k = 1
          **local node      direction      displacement
          4                  02              0.0
*IJK    i    2    1    ** j = 2, k = 1
          **local node      direction      displacement
          1                  02              0.0
          4                  02              0.0
*IJK    I    3    1    ** j = 3, k = 1
          **local node      direction      displacement
          1                  02              0.0
          3:4                01:02           0.0
*IJK    I    1    2    ** j = 1, k = 2
          **local node      direction      displacement
          3                  01:02           0.0
*IJK    I    2    2    ** j = 2, k = 2
          **local node      direction      displacement
          2:3                01:02           0.0
*IJK    I    3    2    ** j = 3, k = 2
          **local node      direction      displacement
          2                  01:02           0.0
          4                  01:02           0.0
** For plane i+1
*IJK    i+1   1    1    ** j = 1, k = 1
          **local node      direction      displacement
          4                  01:02           0.0
*IJK    i+1   2    1    ** j = 2, k = 1
          **local node      direction      displacement
          1                  01:02           0.0
          4                  01:02           0.0
*IJK    i+1   3    1    ** j = 3, k = 1
          **local node      direction      displacement
          1                  01:02           0.0
          3                  01              0.0
*IJK    i+1   1    2    ** j = 1, k = 2
          **local node      direction      displacement
          3                  01              0.0
*IJK    i+1   2    2    ** j = 2, k = 2
          **local node      direction      displacement
          2:3                01              0.0
*IJK    i+1   3    2    ** j = 3, k = 2
          **local node      direction      displacement
          2:4                01              0.0

```

### Casing Treatments with Radial Grids

See “Casing Treatments with Radial Grids” in section “Summary of Geomechanic Model” at the beginning of this chapter.

---

## Prescribed Boundary Conditions (3D)

\*PRESCBC3D

### PURPOSE:

Prescribed displacement-type boundary conditions on a nodal point of a 3D finite element.

### FORMAT:

```
*PRESCBC3D  
*IJK  i1:i2  j1:j2  k1:k2  
      node1(:node2)  direction  displacement
```

### DEFAULT:

If keyword \*PRESCBC3D is absent, the default ‘essential’ boundary condition is applied.

### DEFINITION:

\*PRESCBC3D

Similar to the keyword \*PRESCBC, the keyword \*PRESCBC3D is applied for a case of 3D finite elements. The keyword removes all the essential boundary constraints on a reservoir and prepares for the input of prescribed displacement-type boundary constraints on finite element nodes.

\*IJK

Keyword for reservoir grid blocks.

i1:i2

Starting block index *i1* to ending block index *i2* in X direction

j1:j2

Starting block index *j1* to ending block index *j2* in Y direction

k1:k2

Starting block index *k1* to ending block index *k2* in Z direction

node1(:node2)

Local nodal point (or a range of nodal points) of a finite element to which the boundary condition is assigned. \*KDIR \*UP and \*KDIR \*DOWN are taken into account when a local nodal point is used for prescribed boundary conditions.

direction

Direction of the displacement at a local nodal point is given as:

- 01 displacement along I direction
- 02 displacement along J direction
- 03 displacement along K direction
- 01:03 displacement in three directions.

### *displacement*

Value of displacement (m | ft) at the indicated boundary local nodal point(s) in the indicated direction(s). A positive value means the displacement has the same directions as that of the coordinate, A negative value means the displacement has the opposite direction to the direction of the coordinate.

### **CONDITIONS:**

Keyword \*PRESCBC3D may be used only if \*GEOM3D is present.

Keyword \*PRESCBC3D may not appear more than once, so all prescribed displacement-type boundary conditions must be specified with one \*PRESCBC3D keyword.

### **DEFAULTS:**

If keyword \*PRESCBC3D is absent, the following grid boundaries are constrained:

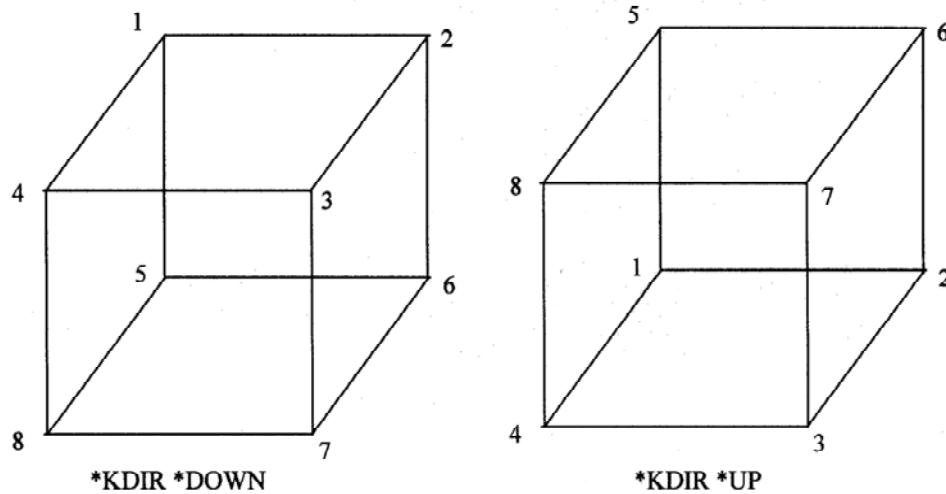
Radial grid: Bottom only

All other grids: Bottom, left, right, front, back (all except top)

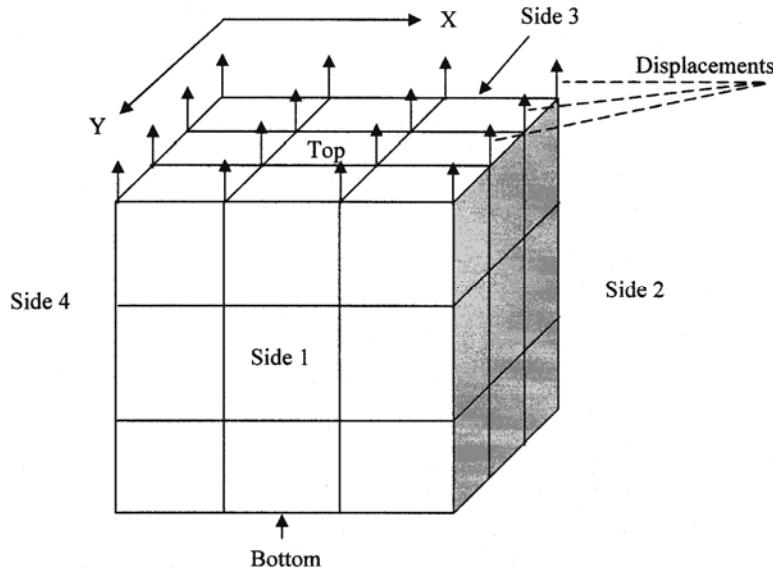
Individual nodes that are not constrained will move freely in space.

### **EXPLANATION:**

Beside the essential boundary conditions which are default, the optional keyword \*PRESCBC3D is designed so that users can select any grid block of a reservoir and assign a displacement value on it. When the keyword is entered in a data set, sub keyword \*IJK must also be accompanied to indicate the constrained block. Following that local node number, direction of the displacement and value of the displacement must be given. If the keyword \*PRESCBC3D appears and the sub keyword \*IJK as well as prescribed displacement are not entered, the reservoir is moving freely. The solution, therefore, can not be predicted. The local node numbers of a finite element also depend on the keyword \*KDIR \*UP or \*KDIR \*DOWN. The local node numbers used in the module are shown as follows:



Example: Consider a reservoir field which has NI = 3, NJ = 3, NK = 3 as shown in the figure.



Assuming the reservoir is constrained all its sides except the top side; the boundary conditions are expressed as follows:

For a case of \*KDIR \*DOWN

```

*PRESCBC3D      ** Keyword for prescribed boundary condition
    **Bottom
*IJK 1:3 1:3 3  ** Keyword for grid block and I,J,K block
                    ** local node   direction   displacement
                    5:8 03 0.0  ** nodes 5, 6, 7, 8 are constrained in Z direction
    ** Side 1
*IJK 1:3 3 1:3      ** local node   direction   displacement
                    3:4        02        0.0
                    7:8        02        0.0
    ** Side 2
*IJK 3 1:3 1:3      ** local node   direction   displacement
                    2:3        01        0.0
                    6:7        01        0.0
    ** Side 3
*IJK 1:3 1 1:3      ** local node   direction   displacement
                    1:2        02        0.0
                    5:6        02        0.0
    ** Side 4
*IJK 1 1:3 1:3      ** local node   direction   displacement
                    1          01        0.0
                    4          01        0.0
                    5          01        0.0
                    8          01        0.0

```

```

For a case of *KDIR *UP
*PRESCBC3D          ** Keyword for prescribed boundary condition
    **Bottom
*IJK 1:3 1:3 1 ** Keyword for grid block and I,J,K block
    ** local node direction displacement
    1:4 03 0.0 ** nodes 1, 2, 3, 4 are constrained in Z direction
    ** Side 1
*IJK 1:3 3 1:3          ** local node direction displacement
    3:4      02      0.0
    7:8      02      0.0
    ** Side 2
*IJK 3 1:3 1:3          ** local node direction displacement
    2:3      01      0.0
    6:7      01      0.0
    ** Side 3
*IJK 1:3 1 1:3          ** local node direction displacement
    1:2      02      0.0
    5:6      02      0.0
    ** Side 4
*IJK 1 1:3 1:3          ** local node direction displacement
    1      01      0.0
    4      01      0.0
    5      01      0.0
    8      01      0.0
Assuming that the top of reservoir is pulled by an amount of 0.5 ft (or m)
indicated by the arrows on the top, the prescribed boundary conditions at
those nodal points are given as:
For *KDIR *DOWN
*IJK 1:3 1:3 1          ** local node direction displacement
    1:4      03      -0.5
    ** minus sign since its direction is opposite to Z direction
For *KDIR *UP
*IJK 1:3 1:3 3          ** local node direction displacement
    5:8      03      0.5
    ** plus sign not minus sign

```

### Casing Treatments with Radial Grids

See “Casing Treatments with Radial Grids” in section “Summary of Geomechanic Model” at the beginning of this chapter.

## Point Loads (2D)

\*PLOADBC

### PURPOSE:

Specify external loads on any nodal points of a finite element.

### FORMAT:

```
*PLOADBC  
{ node1(:node2) I/J_load K_load }
```

or

```
*PLOADBC  
{ *PLANE p1(:p2)  
{ node1(:node2) I/J_load K_load } }
```

or

```
*PLOADBC  
{ *IJK i1:i2 j1:j2 k1:k2  
{local node1(:local node2) I/J_load K_load } }
```

### DEFINITIONS:

#### \*PLOADBC

This keyword specifies external loads applied to nodal points.

#### node1(:node2)

Single nodal point (or range of nodal points) of a finite element to which the boundary condition is assigned. See the description for \*PRESCBC.

#### \*IJK

Keyword for reservoir grid blocks

#### i1:i2

Starting block index *i1* to ending block index *i2* in X direction

#### j1:j2

Starting block index *j1* to ending block index *j2* in Y direction

#### k1:k2

Starting block index *k1* to ending block index *k2* in Z direction

#### local node1(:local node2)

Local nodal point (or a range of local nodal points) of a 2D finite element on which the loads are applied. These local nodal points are strongly dependent on keywords \*KDIR UP, \*KDIR \*DOWN and \*PLSTRAINY. The convention of nodal points is given in the explanation.

### *I/J\_load*

Load (kN | tonf) applied to the indicated node(s) in (1) the I direction when strains are computed in planes normal to the J direction (\*PLSTRAINY is present), and (2) the J direction when strains are computed in planes normal to the I direction (\*PLSTRAINY is absent). The load value is positive or negative depending on the actual load direction relative to the grid direction. Enter a zero value if no load is applied to the node(s) on this direction (e.g., only *K\_load* is non-zero).

### *K\_load*

Load (kN | tonf) applied to the indicated node(s) in the K direction. The load value is positive or negative depending on the actual load direction relative to the grid direction. Enter a zero value if there is no load applied to the node(s) (e.g., only *I/J\_load* is zero).

### \*PLANE *p1(:p2)*

See the description for \*PRESCBC.

### **DEFAULTS:**

If keyword \*PLOADBC is absent then all external loads are zero.

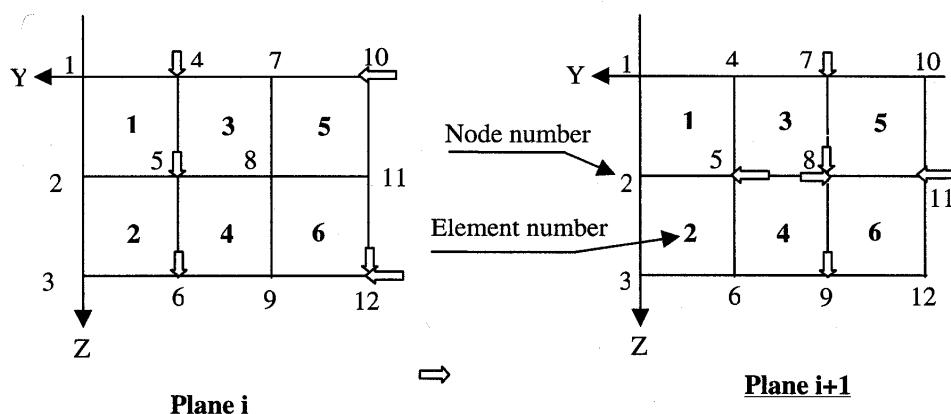
If sub-keyword \*PLANE is absent, all the planes have the same prescribed boundary conditions.

### **EXPLANATION:**

Keyword \*PLOADBC allows you to assign external loads at any node on any finite element and on any plane which is cut from the reservoir either along the I or along the J direction. Similar to the optional keyword \*PRESCBC, there are two different approaches which are used to describe the loading boundary as follows:

First approach: Using global nodal points

### **Example**



The point loads at nodes as showed in the above figures is specified as follows.

```
*PLOADBC
  *PLANE i
    ** Node      Y Load      Z Load
      4:6        0          100
      10         15          0
      12         10          10
  *PLANE i+1
    ** Node      Y Load      Z Load
      5          100         0
      7          0          10
      8         -10          10
      9          0          15
     11         15          0
```

Note that a zero value is entered when no load is applied.

Second approach: Using I, J, K block and local nodal points of a finite element. For convention of local nodal points of a 2D finite element, please see keyword \*PRESCBC.

The point loads as seen in the above figure are expressed as:

```
*PLOADBC
  ** on plane i

  *IJK   I   1   1   ** j = 1, k = 1
           **local node      Y load      Z load
           3:4                 0          100
  *IJK   I   2   1   ** j = 2, k = 1
           **local node      Y load      Z load
           1:2                 0          100
  *IJK   I   3   1   ** j= 3, k = 1
           **local node      Y load      Z load
           4                 15          0
  *IJK   I   1   2   ** j= 1, k = 2
           **local node      Y load      Z load
           3:4                 0          100
  *IJK   I   3   1   ** j= 2, k = 2
           **local node      Y load      Z load
           1:2                 0          100
  *IJK   I   3   2   ** j= 3, k = 2
           **local node      Y load      Z load
           3                 10          10
  ** on plane i+1
  *IJK   i+1  1   1   ** j = 1, k = 1
           **local node      Y load      Z load
           3                 100         0
  *IJK   i+1  2   1   ** j = 2, k = 1
           **local node      Y load      Z load
           2                 100         0
           3                -10          10
           4                  0          10
```

```

*IJK    i+1   3   1   ** j= 3, k = 1
        **local node      Y load      Z load
          1                 0           10
          2                -10          10
          3                 15           0
*IJK    i+1   1   2   ** j= 1, k = 2
        **local node      Y load      Z load
          4                 100          0
*IJK    i+1   3   1   ** j= 2, k = 2
        **local node      Y load      Z load
          1                 100          0
          3                 0           15
          4                -10          10
*IJK    i+1   3   2   ** j= 3, k = 2
        **local node      Y load      Z load
          1                -10          10
          2                 0           15
          4                 15           0

```

## Point Loads (3D)

\*PLOADBC3D

### PURPOSE:

Specify external load on any nodal points of a 3D finite element.

### FORMAT:

```
*PLOADBC3D
  {*IJK      i1:i2  j1:j2  k1:k2
    node1(:node2) I_load   J_load   K_load }
```

### DEFINITIONS:

\*PLOADBC3D

The keyword indicates external loads applied to nodal points.

\*IJK

Keyword for reservoir grid blocks

i1:i2

Starting block index *i1* to ending block index *i2* in X direction

j1:j2

Starting block index *j1* to ending block index *j2* in Y direction

k1:k2

Starting block index *k1* to ending block index *k2* in X direction

node1(:node2)

Single local nodal point (or range of nodal points) of a finite element to which the load is assigned. See the keyword \*PRESCBC3D for details of local nodal point.

I\_load

Load (kN | tonf) applied in the I direction.

J\_load

Load (kN | tonf) applied in the J direction.

K\_load

Load (kN | tonf) applied in the K direction.

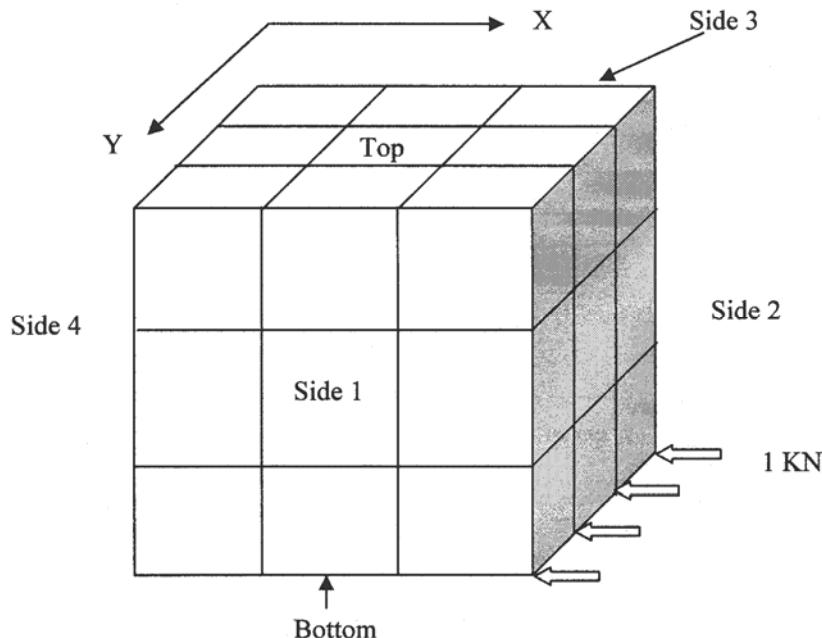
### DEFAULTS:

If the keyword \*PLOADBC3D is absent, there is no external load.

## EXPLANATION:

Similar to the keyword \*PLOADBC for 2D plane strain, the optional keyword \*PLOADBC3D is applied for 3D finite element. With this keyword, the user can assign external loads on any nodal points of a 3D finite element. The load value is positive or negative in one direction depending on the actual load direction relative to the grid direction. Values of I\_load, J\_load and K\_load must be entered in the table even though they are zero. External loads expressed on a finite element also strongly depends on \*KDIR \*UP and \*KDIR \*DOWN.

### Example



Assuming external loads 1 KN are applied at the bottom edge on side 2 of the above reservoir as illustrated by arrows. Expressing the above loads in a data set is shown as:

```

For *KDIR *DOWN
*PLOADBC3D
*IJK 3      1:3      3      ** I = 3, J=1 to J=3, K = 3
    ** node   I_load   J_load   K_load
        6:7     -1.0     0.0     0.0
    ** negative load due to opposite direction to X
For *KDIR *UP
*PLOADBC3D
*IJK 3      1:3      1      ** I = 3, J=1 to J=3, K = 1
    ** node   I_load   J_load   K_load
        2:3     -1.0     0.0     0.0
    ** negative load due to opposite direction to X

```

## Distributed Edge Loads (2D)

\*DLOADBC, \*DLOADBC2D

### PURPOSE:

Specify distributed loads along an edge of a finite element.

### FORMAT:

```
*DLOADBC or *DLOADBC2D
{ *ELEMENT ele
  { en1:en2  NL1  TL1  NL2  TL2 } }
```

or

```
*DLOADBC or *DLOADBC2D
{ *PLANE pl
  { *ELEMENT ele
    { en1:en2  NL1  TL1  NL2  TL2 } } }
```

or

```
*DLOADBC or *DLOADBC2D
{ *IJK  i1:i2  j1:j2  k1:k2
  { en1:en2  NL1  TL1  NL2  TL2 } } }
```

### DEFINITIONS:

#### \*DLOADBC

Specify distributed loads for finite element edges where 2D plane strain is considered as a thin plate. The unit of loading quantities is (kN/m | tonf/ft).

#### \*DLOADBC2D

Specify distributed loads for finite element edges for 2D axi-symmetric problems. The specified edge loading is multiplied by the entire circumference to get distributed loading per unit area. The unit of loading quantities is (kPa/m | psi/ft).

#### \*ELEMENT ele

Element number. The allowed range is 1 to the maximum number of non-null elements in a plane. The definition of null elements is the same as that of null grid blocks in the reservoir section. Null elements are treated as non-existing elements.

#### en1:en2 NL1 TL1 NL2 TL2

en1:en2 are the starting and ending nodes, respectively, of an edge of finite element ele. An edge is defined by two nodes on the element. More than one of these data lines may follow \*ELEMENT ele.

The remaining four quantities are load per unit length. Enter zero for a quantity for which no load is specified.

$NL1$	normal load at $en1$
$TL1$	tangential load at $en1$
$NL2$	normal load at $en2$
$TL2$	tangential load at $en2$

#### \*PLANE $p1$

See the description for \*PRESCBC. Only one plane number is allowed.

#### DEFAULTS:

If sub-keyword \*PLANE is absent, all planes have the same distributed load.

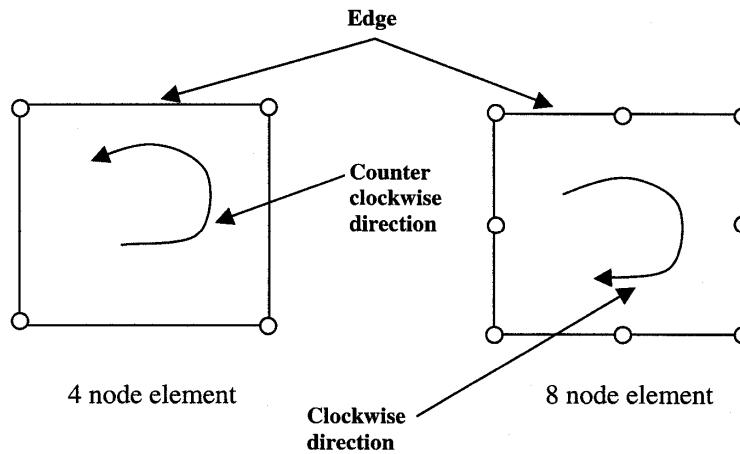
#### EXPLANATION:

Keyword \*DLOADBC allows you to distribute loads on edges of finite elements. An edge is determined by two sequential nodes on the element. There are 4 edges on an element constituted by 4 nodes, and there are 8 edges on an element constituted by 8 nodes or 9 nodes. Similar to prescribed boundary conditions and point loads, there are also two different approaches to define the distributed load on the edges of a finite element.

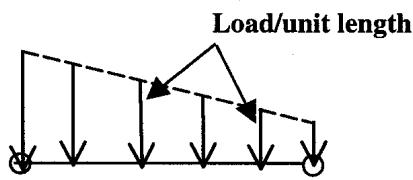
First approach: Based on global nodal points on a plane.

In this approach, rules for assigning starting nodes and ending nodes correctly are:

- i) For plane strains along I direction: starting nodes and ending nodes must be entered in the counter clockwise direction.
- ii) For plane strains along J direction: (i.e. when keyword \*PLSTRAINY appears): starting nodes and ending nodes must be entered in the clockwise direction.
- iii) For axis-symmetric case: starting nodes and ending nodes must be entered in the clockwise direction.

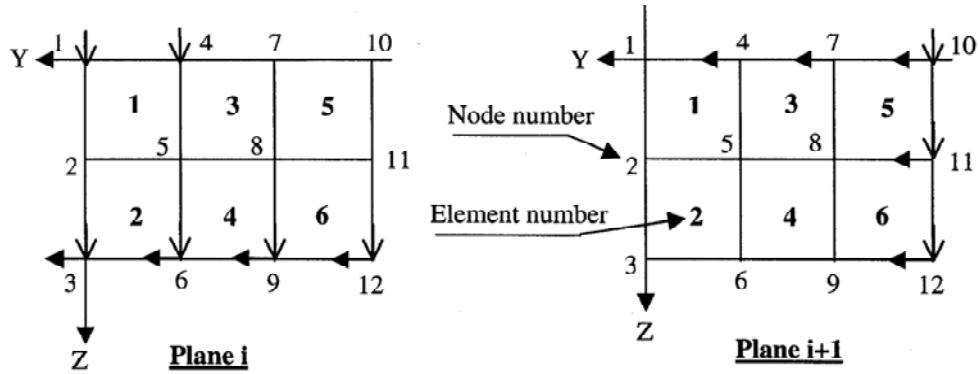


An illustration of distributed edge loads on one edge is shown in the following figure.



If the load is positive its direction should be the same as the coordinate direction, and if the load is negative its direction is opposite to the coordinate direction.

### Example



Input formats corresponding to the above figures are shown below:

```

*DLOADBC
*PLANE i
*ELEMENT 1
** SN1:EN2    NL1    TL1    NL2    TL2
        4:1      10     0      5      0
*ELEMENT 2
** SN1:EN2    NL1    TL1    NL2    TL2
        3:6      1      2      5      5
*ELEMENT 4
** SN1:EN2    NL1    TL1    NL2    TL2
        6:9      10     1      10     1
*ELEMENT 6
** SN1:EN2    NL1    TL1    NL2    TL2
        9:12     2      1      2      1
*PLANE i+1
*ELEMENT 3
** SN1:EN2    NL1    TL1    NL2    TL2
        7:4      0      1      0      1
*ELEMENT 5
** SN1:EN2    NL1    TL1    NL2    TL2
        10:7     4      4      3      3
        11:10    7      8      1      2
*ELEMENT 6
** SN1:EN2    NL1    TL1    NL2    TL2
        12:11    4      5      3      1

```

Note the zero values in some column corresponding to no force. In plane  $i+1$ , there is one null element; thus, the total number of elements in that plane is only 5 instead of 6 as shown in plane  $i$ . Since the strains are computed in plane **normal to the I direction**, the starting node and ending node are entered in the counter clockwise direction as shown in the example.

Second approach: Based on the local nodal points which are defined in the section of keyword \*PRESCBC.

In this approach the direction of local nodes is not as important as it is for the first approach. The value (sign) of normal load determines whether the edge is compressive (positive) or tensile (negative). In addition, users need to know only the grid block location which is based on (I,J,K), the local nodes of a finite element and the values of loads on assigned nodes.

Using this approach, the above example of distributed loads can be expressed as:

```
*DLOADBC
** on plane i
*IJK    i    1    1
      ** local node1:local node2      NL1      TL1      NL2      TL2
          4:1           10        0        5        0
*IJK    i    1    2
      ** local node1:local node2      NL1      TL1      NL2      TL2
          2:3           1         2        5        5
*IJK    i    2    2
      ** local node1:local node2      NL1      TL1      NL2      TL2
          2:3           10        1       10        1
*IJK    i    3    2
      ** local node1:local node2      NL1      TL1      NL2      TL2
          3:2           2         1        2        1

** on plane i+1
*IJK    i    2    1
      ** local node1:local node2      NL1      TL1      NL2      TL2
          4:1           0         1        0        1
*IJK    i    3    1
      ** local node1:local node2      NL1      TL1      NL2      TL2
          4:1           4         4        3        2
          3:4           7         8        1        2
*IJK    i    3    2
      ** local node1:local node2      NL1      TL1      NL2      TL2
          3:4           4         5        3        1
```

---

## Distributed Surface Loads (3D)

\*DLOADBC3D, \*DLOADIJK

### PURPOSE:

Specify distributed load on a surface of a 3D finite element.

### FORMAT:

```
*DLOADBC3D
  {*IJK    i1:i2 j1:j2 k1:k2
   node1 node2 node3 node4 load }
*DLOADIJK
  {*IJK    i1:i2 j1:j2 k1:k2
   *Face
   node loadx loady loadz }
```

### DEFINITIONS:

#### \*DLOADBC3D

Specify distributed normal loads on a surface of a 3D finite element.

#### \*IJK

Keyword for reservoir grid blocks.

*i1*

Starting block index in I direction.

*i2*

Ending block index in I direction.

*j1*

Starting block index in J direction.

*j2*

Ending block index in J direction.

*k1*

Starting block index in K direction.

*k2*

Ending block index in K direction.

*node1*

First nodal point on a surface.

*node2*

Second nodal point on a surface.

<i>node3</i>	Third nodal point on a surface.
<i>node4</i>	Fourth nodal point on a surface.
<i>load</i>	Uniform load ( $\text{kN/m}^2$   $\text{tonf/m}^2$ ); ( $\text{tonf/m}^2 = 13.88 \text{ psi}$ ).
<b>*DLOADIJK</b>	Specify distributed loads on three directions at each node of an element.
<b>*Face</b>	Face of a local element. It may be *LEFT, *RIGHT, *FRONT, *BACK, *TOP, *BOTTOM.
<i>node</i>	Node number of a local 3D element.
<i>loadx</i>	Distributed load in the I direction (kPa   psi).
<i>loady</i>	Distributed load in the J direction (kPa   psi).
<i>loadz</i>	Distributed load in the K direction (kPa   psi).

#### **CONDITIONS:**

When using \*DLOADBC3D, four nodal points on a surface of a 3D finite element must be entered.

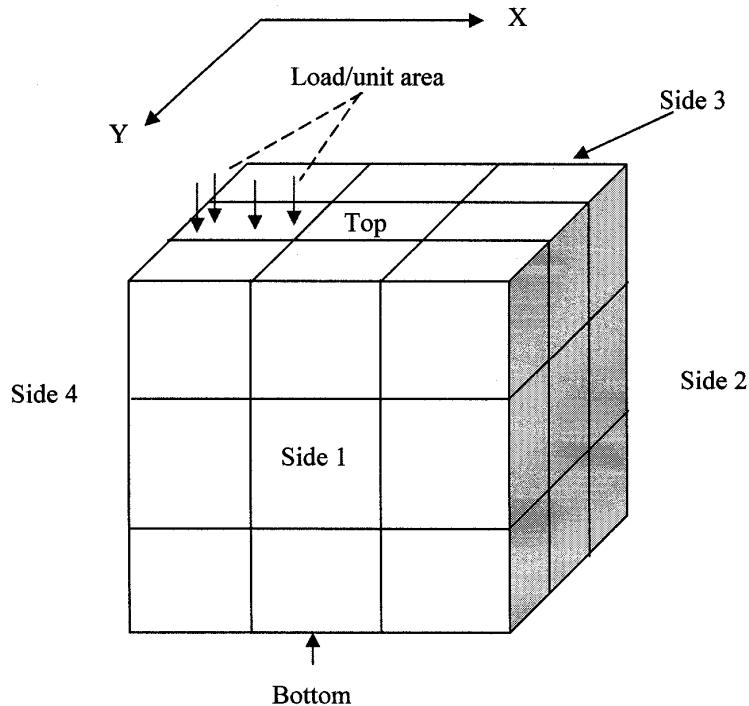
When using \*DLOADIJK, face of the element must be defined before the nodes are assigned. The local nodes must lie on the surface of the element.

#### **EXPLANATION:**

##### **Distributed Normal Loads**

Keyword \*DLOADBC3D allows the user to distribute loads evenly on a surface of a 3D finite element. As mentioned in the keyword \*GEOM3D, a 3D finite element consists of 8 nodal points and each face of it is composed by 4 nodal points. Therefore, to indicate which surface of a finite element is subject to distributed loads, a set of four nodal points must be entered but is not necessarily in order. A uniform load is given after the nodal points. If the load value is positive, the surface of a finite element is compressed. If the load value is negative, the surface is stretched. Again, \*KDIR \*DOWN and \*KDIR \*UP must be taken into account.

Example: Consider the reservoir shown below. A uniform distributed load of 1kN/m<sup>2</sup> is applied on one block of the reservoir. Since the load is positive, the arrows are pointing on the surface as illustrated.



For \*KDIR \*DOWN:

Since the load is distributed on surface of the block I = 1, J = 2 and K = 1, the data entry for such loads is given as (unit load in tonf/m<sup>2</sup>):

```
*DLOADBC3D
*IJK      1     2     1
          ** node1    node2    node3    node4    load
                  1         2         3         4       1.0
```

For \*KDIR \*UP:

In this case, the load is distributed on surface of the block I = 1, J = 2 and K = 3, the data entry for such loads is given as (unit load in tonf/m<sup>2</sup>):

```
*DLOADBC3D
*IJK      1     2     3
          ** node1    node2    node3    node4    load
                  5         6         7         8       1.0
```

## Distributed Arbitrary Loads

Keyword \*DLOADIJK can handle arbitrarily distributed loads on a face of a finite element. When using this keyword, the following procedures must be followed:

1. Enter keyword \*DLOADIJK in the geomechanical section.
2. Enter keyword \*IJK to determine the region on which distributed loads are applied.
3. Enter a face of the element on which the load is acting.
4. Enter node numbers with loads in directions. The load is positive when it points to the same direction of the coordinate.

Using the above figure, facing names are denoted as:

<b>LEFT</b>	is equivalent to side 4
<b>RIGHT</b>	is equivalent to side 2
<b>FRONT</b>	is equivalent to side 1
<b>BACK</b>	is equivalent to side 3
<b>TOP</b> and <b>BOTTOM</b> are defined in the figure	

Local node numbers can be seen in the section of keyword \*PRESCBC3D.

### Example 1

Using the above example of DLOADABC3D, the equivalent data for this keyword are given as follows:

For \*KDIR \*DOWN:

```
*DLOADIJK
*IJK 1 2 1
*TOP **loadx  loady  loadz
      1      0      0    13.88    ** Unit load in psi
      2      0      0    13.88    ** Unit load in psi
      3      0      0    13.88    ** Unit load in psi
      4      0      0    13.88    ** Unit load in psi
```

The positive load in this case indicates that the surface is compressed.

For \*KDIR \*UP:

```
*DLOADIJK
*IJK 1 2 1
*TOP **loadx  loady  loadz
      1      0      0   -13.88    ** Unit load in psi
      2      0      0   -13.88    ** Unit load in psi
      3      0      0   -13.88    ** Unit load in psi
      4      0      0   -13.88    ** Unit load in psi
```

The negative load in this case indicates that the surface is compressed.

### Example 2

When the shearing loads along the K direction are applied to side 2 and side 4, the data entry is given as:

For KDIR DOWN:

```
*DLOADIJK
*IJK 1 1:3 1:3
*LEFT ** equivalent to side 4 of the figure
 1      0      0     -13.88 ** Stretching this face
 4      0      0     -13.88 ** along K direction
 5      0      0     -13.88
 8      0      0     -13.88
*IJK 3 1:3 1:3
*RIGHT ** equivalent to side 2 of the figure
 1      0      0      13.88 ** Compressing this face
 4      0      0      13.88 ** along K direction
 5      0      0      13.88
 8      0      0      13.88
```

### Example 3

When the normal distributed loads along the I direction are applied to side 4, the data entry is given as:

For KDIR DOWN:

```
*DLOADIJK
*IJK 1 1:3 1:3
*LEFT ** equivalent to side 4 of the figure
 1      13.88     0      0 ** Compressing this face
 4      13.88     0      0 ** along I direction
 5      13.88     0      0
 8      13.88     0      0
```

### Example 4

Another feature of the keyword \*DLOADIJK is that the number of nodes may vary from 1 to 4. For instance, if only three nodes at the corner of the top surface of the element (1,1,1) in the figure with KDIR DOWN are loaded, the data entered in the geomechanical section is:

```
*DLOADIJK
*IJK 1 1 1
*TOP
 1      0      0      13.88
 2      0      0      13.88
 4      0      0      13.88
```

Here there are only three nodes on a TOP face subject to distributed loads. In this case, the loads are distributed on the area defined by nodes 1, 2 and 4.

It should be noted that the above distributed loads are different from these distributed loads.

```
*DLOADIJK
*IJK 1 1 1
*TOP
 1      0      0      13.88
 2      0      0      13.88
 4      0      0      13.88
 3      0      0      0.0
```

In this case the loads are distributed on the area defined by 4 nodes 1, 2, 3 and 4.

---

## Gravity Loads (2D)

\*GLOADBC, \*SPECGRAV

### PURPOSE:

Specify direction and magnitude of body forces on a per-plane basis.

### FORMAT:

```
*GLOADBC  
  { p1(:p2) theta }  
*SPECGRAV spec_grav
```

### DEFINITIONS:

#### \*GLOADBC

Specify gravity load direction on a per-plane basis.

#### p1(:p2)

Single plane number, or range of plane numbers. See the description for \*PRESCBC.

#### theta

Angle in degrees between the gravity direction (down) and K direction.

#### spec\_grav

Specify gravity (density divided by water density of 9806.650 kg/m<sup>3</sup>) of the material in the planes. The body force acting on a plane is computed from the material density and each grid block volume.

### CONDITIONS:

Keywords \*GLOADBC and \*SPECGRAV must appear together if at all, that is, they must be either both absent or both present.

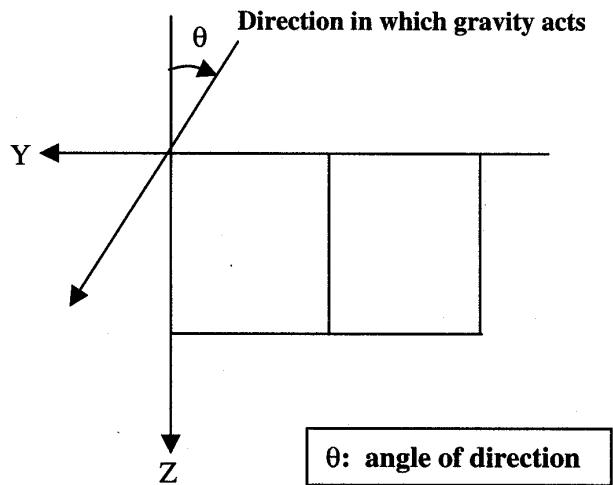
### DEFAULTS:

If keyword \*GLOADBC is present but does not refer to a particular plane, then the angle is zero for that plane.

There is no default for \*SPECGRAV. If \*SPECGRAV is absent then \*GLOADBC must be absent as well.

### EXPLANATION:

Keyword \*GLOADBC allows you to assign a body force on any plane. Specific gravity controlled by keyword \*SPECGRAV is used to compute the body force per unit volume for each element in the plane. Keyword \*GLOADBC is active only when \*SPECGRAV is given.



### Example

```

*GLOADBC
  ** plane #      ** theta (degrees)
    1                  10
    2:5                20
*SPECGRAV 1.2  ** density is 1.2x9806.65 = 11767 kg/m3

```

On plane # 1 the K axis is 10 degrees from vertical, whereas on planes 2, 3, 4 and 5 the K axis is 20 degrees from vertical.

---

## Gravity Loads (3D)

\*GLOADBC3D, \*SPECGRAV

### PURPOSE:

Specify direction and magnitude of body forces on a per-plane basis in 3D.

### FORMAT:

\*GLOADBC3D  
{ *p1(:p2) theta1 theta2* }

\*SPECGRAV *spec\_grav*

### DEFINITIONS:

\*GLOADBC3D

Specify gravity load direction on a per-plane basis when 3D finite element is used in computation.

*p1(:p2)*

Single plane number, or range of plane numbers. The plane number for 3D finite element is equivalent to the index I on X axis.

*theta1*

Angle in degrees between the gravity vector and X direction.

*theta2*

Angle in degrees between the gravity vector and Y direction.

*spec\_grav*

Specify gravity (density divided by water density of 9806.650 kg/m<sup>3</sup>) of the material in the planes. The body force acting on a plane is computed from the material density and each grid block volume.

### CONDITIONS:

Keywords \*GLOADBC3D and \*SPECGRAV must appear together if at all, that is, they must be either both absent or both present.

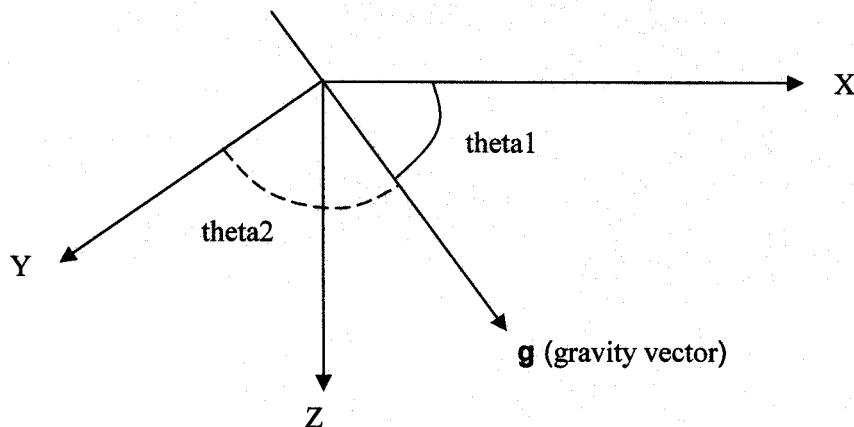
### DEFAULTS:

If keyword \*GLOADBC3D is present but does not refer to a particular plane, then the angles theta1 and theta2 are 90 degrees.

There is no default for \*SPECGRAV. If \*SPECGRAV is absent then \*GLOADBC must be absent as well.

## EXPLANATION:

Keyword \*GLOADBC3D allows you to assign a body force in space. Specific gravity controlled by keyword \*SPECGRAV is used to compute the body force per unit volume for each element in the plane. Keyword \*GLOADBC3D is active only when \*SPECGRAV is given.



Note: The angles with respect to X axis and Y axis must satisfy the following condition:

$$\{\cos(\text{theta1})\}^2 + \{\cos(\text{theta2})\}^2 \leq 1$$

### Example

```
*GLOADBC3D
  ** plane #           ** theta1 (degrees)   ** theta2
    1                  70                      80
    2:5                90                      60
  *SPECGRAV 1.2  ** density is 1.2x9806.65 = 11767 kg/m3
```

On plane # 1 (i.e. I = 1), the gravity vector made an angle 70 degrees with respect to X axis and an angle 80 degrees with respect to Y axis. Whereas, on plane 2:5 (I=2 to I=5), the angles are 90 degrees and 60 degrees with respect to X axis and Y axis.

---

## Fixed Null Block

\*RIGIDNULL

### PURPOSE:

Assign fixed boundary to nodes around a null block.

### FORMAT:

```
*RIGIDNULL *ALL  
or  
*RIGIDNULL *ELEMENT  
{ plane element }
```

### DEFINITIONS:

\*RIGIDNULL \*ALL

Specifies fixed boundary for nodes around all null blocks.

\*RIGIDNULL \*ELEMENT

Specifies fixed boundary for nodes around null blocks associated with the indicated elements.

*plane element*

Plane and element number, respectively, of the element corresponding to the null block. See the description of plane and element numbering for keyword \*PRESCBC.

### CONDITIONS:

If keyword \*RIGIDNULL is present, only one of sub-keyword \*ALL or \*ELEMENT may be used.

### DEFAULT:

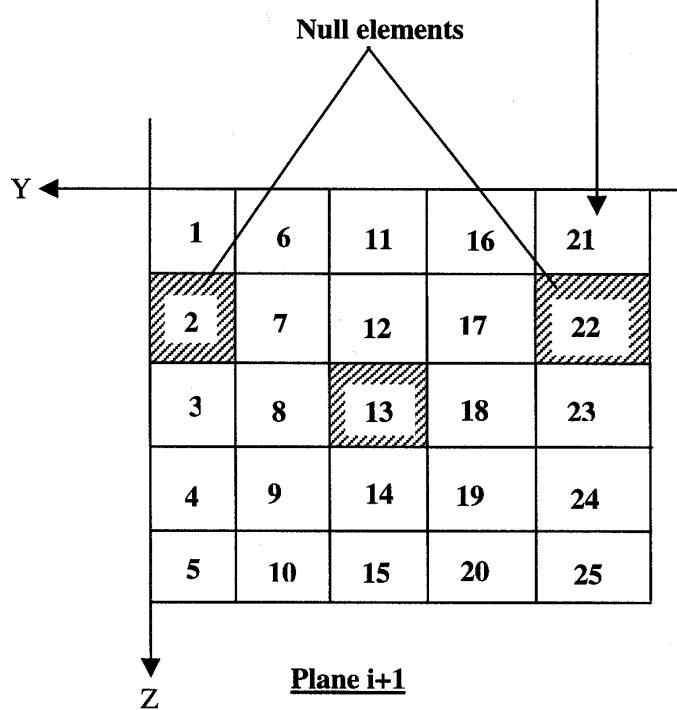
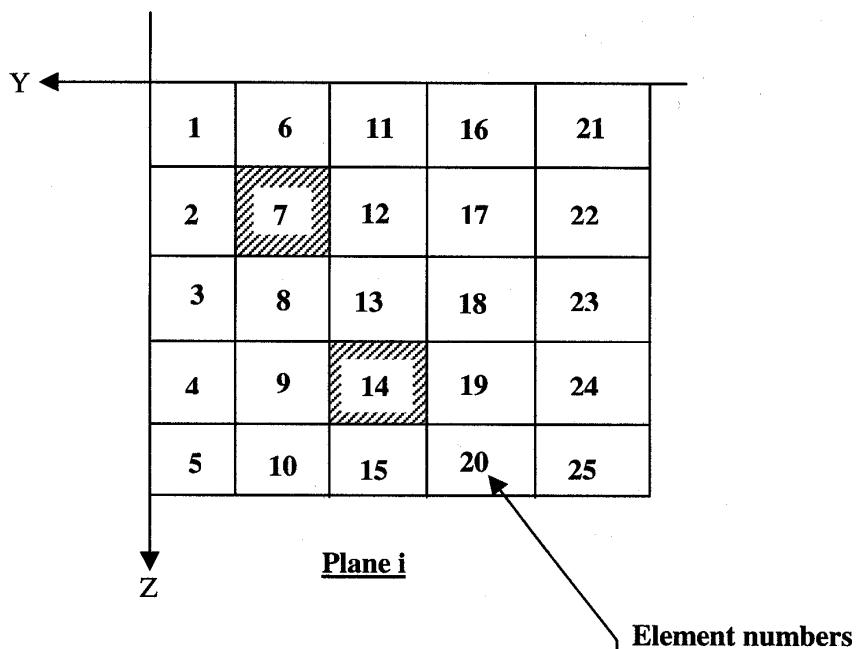
When keyword \*RIGIDNULL is absent, movement is allowed for nodes on the edge common to a null block and a non-null block in the plane in which strain is calculated. Note that, by definition, node movement is not allowed in the direction perpendicular to the strain plane.

### EXPLANATION:

Keyword \*RIGIDNULL allows you to specify null blocks whose surrounding nodes will be completely rigid, that is, will not be allowed to move during calculation of the strain problem. Thus, all nodal points on the attached edges around the specified null block are not moving in any direction.

### Example

In the figures, plane i contains two null blocks while plane i+1 contains 3 null blocks.



If all null blocks are fixed, use the following:

```
*RIGIDNULL *ALL
```

If null blocks only in plane i are fixed, use the following:

```
*RIGIDNULL *ELEMENT  
** plane_number element_number  
    i           7  
    i           14
```

---

## **Fixed Cap Rock**

**\*RIGIDTOP**

### **PURPOSE:**

Constrain vertical movement on the cap rock of a reservoir.

### **FORMAT:**

**\*RIGIDTOP**

### **DEFINITIONS:**

**\*RIGIDTOP**

This single keyword causes the cap rock of a reservoir to be vertically constrained at the top.

### **DEFAULT:**

If keyword **\*RIGIDTOP** is absent, then the cap rock of the reservoir is either not constrained or is constrained by another keyword such as **\*PRESCBC**.

### **EXPLANATION:**

Keyword **\*RIGIDTOP** allows you to assign constrained boundary conditions at the top of the reservoir on the vertical direction. This keyword is an alternative to using keyword **\*PRESCBC** to assign displacement boundary conditions on a reservoir. However, keyword **\*RIGIDTOP** is more convenient because the “essential” boundary conditions are not removed automatically as they are when keyword **\*PRESCBC** is used. Essential boundary conditions are defined as laterally constrained movements on vertical sides of a reservoir and vertically constrained movements at the bottom of a reservoir.

---

## Geomechanics Domain

\*GEODOMAIN

### PURPOSE:

Specify grid sub domains in which geomechanics calculations are performed.

### FORMAT:

\*GEODOMAIN \*ALL

or

\*GEODOMAIN \*IJK

{ i1:i2 j1:j2 k1:k2 }

### DEFINITIONS:

\*GEODOMAIN \*ALL

Geomechanics calculations are performed for all active grid blocks.

\*GEODOMAIN \*IJK

Geomechanics calculations are performed only in selected regions of grid blocks called geomechanics domains.

i1:i2 j1:j2 k1:k2

Range of grid indices in the I, J and K directions. Each index value must fall within 1 to the maximum number of blocks in that direction (see keyword \*GRID). Each lower index must not exceed the corresponding upper index.

Enter a single index as a range, for example, 5:5.

### DEFAULT:

If keyword \*GEODOMAIN is absent, then \*GEODOMAIN \*ALL is assumed.

If keyword \*GEODOMAIN \*IJK is present, then geomechanical calculations are not done for blocks that are not referenced explicitly.

### EXPLANATION:

Keyword \*GEODOMAIN allows you to restrict coupled geomechanics calculations to specified sub domains of a reservoir, resulting in increased efficiency and saving in CPU time.

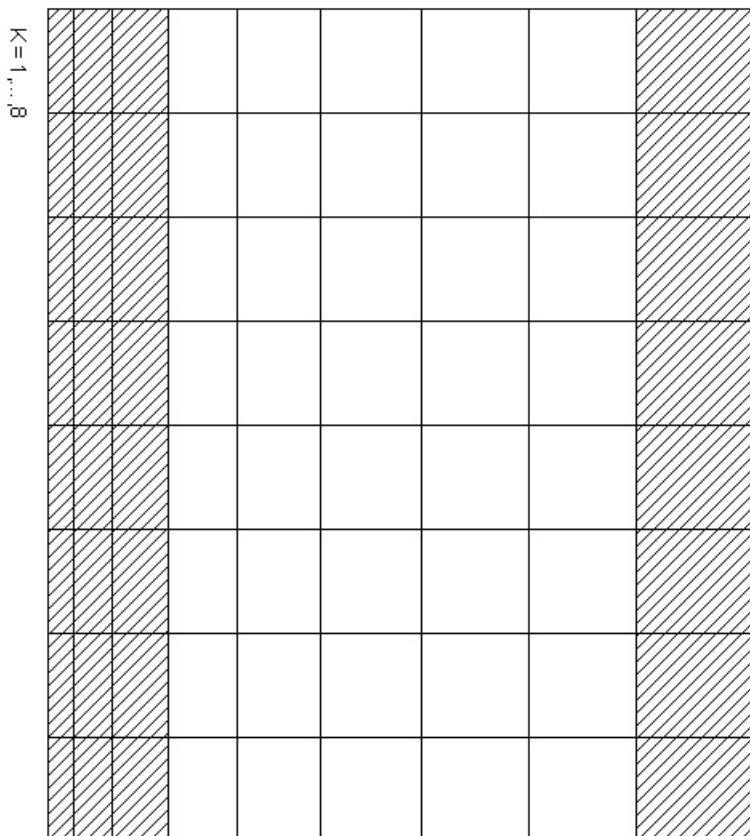
Geomechanical response results (stresses, strains, Young modulus, Poisson ratio, etc.) are calculated only for blocks in the geomechanics domains. However, to satisfy textual and graphical output requirements these quantities are given zero values for blocks that are not in a geomechanics domain.

Example: In an axis-symmetry reservoir with NI = 9, NJ = 1, NK = 8, there are two geomechanics domains defined as follows:

```
*GEODOMAIN *IJK
  ** i1:i2    j1:j2    k1:k2
      1:3      1:1      1:8  ** domain 1
      9:9      1:1      1:8  ** domain 2
```

As shown in the figure, the marked grid blocks belong to geomechanics domains and the blank grid blocks are outside the geomechanics domains.

$$l = 1, \dots, 9$$



---

## Pressure Boundary Domain

\*BCDOMAIN

### PURPOSE:

Specify pressure boundary between geomechanics and non-geomechanics grid domains.

### FORMAT:

\*BCDOMAIN \*ALL

or

\*BCDOMAIN \*IJK

{ i1:i2 j1:j2 k1:k2 }

### DEFINITIONS:

\*BCDOMAIN \*ALL

Apply pressure boundary between all parts of geomechanics and non-geomechanics grid domains.

\*BCDOMAIN \*IJK

Apply pressure boundary between selected parts of geomechanics and non-geomechanics grid domains.

{ i1:i2 j1:j2 k1:k2 }

Table of ranges of grid indices in the I, J and K directions. Each index value must fall within 1 to the maximum number of blocks in that direction (see keyword \*GRID). Each lower index must not exceed the corresponding upper index. Enter a single index value as a range, for example, 5:5.

### DEFAULT:

If keyword \*BCDOMAIN is absent then no pressure boundary is applied.

If keyword \*BCDOMAIN is present, keyword \*GEODOMAIN must appear.

### EXPLANATION:

Keyword \*GEODOMAIN specifies one or more block regions or domains in which geomechanics calculations are performed. Blocks omitted from \*GEODOMAIN make up one or more non-geomechanics domains. The result is one or more surfaces that separate the blocks in geomechanics domains from blocks in non-geomechanics domains.

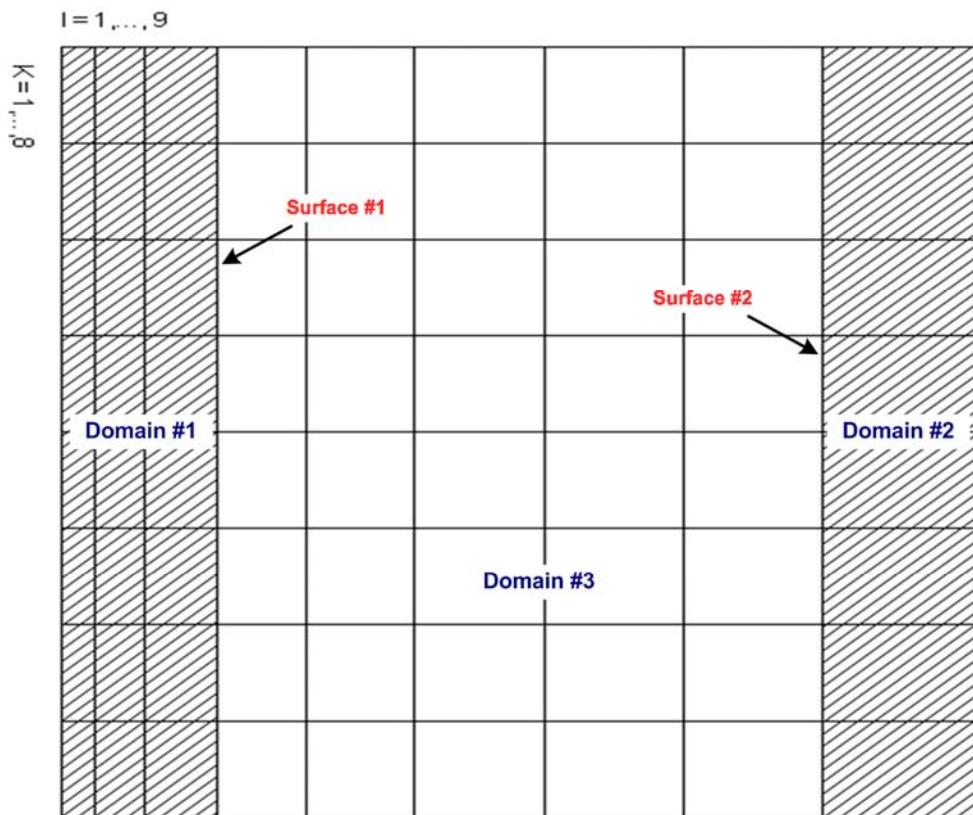
Consider two adjacent blocks, one in a geomechanics domain and the other in a non-geomechanics domain. By default, the block in the non-geomechanics domain has no pressure influence on the block in the geomechanics domain, as though there is a no-flow boundary between them. This is appropriate in some circumstances, for example, when a rigid unperforated casing isolates a wellbore volume from the surround formation.

However, in cases where there is no rigid no-flow boundary between the geomechanics and non-geomechanics domains, use keyword \*BCDOMAIN to apply a pressure-type boundary between them. This lets the fluid pressure of a block in the non-geomechanics domain influence its neighbouring block in the geomechanics domain as a stress constraint. This is appropriate when modelling deforming regions together with open or void volumes, such as non-cased wellbores and open caverns.

Due to the CPU characteristics of the algorithm used to search for boundary locations, the number of cells referenced in keyword \*BCDOMAIN should be as few as possible.

#### **Example: Continuation of \*GEODOMAIN example**

This 9x1x8 grid consists of geomechanics domains “Domain #1” in cells (1:3,1,1:8) and “Domain #2” in cells (9,1,1:8), along with non-geomechanics domain “Domain #3” in cells (4:8,1,1:8). Use keyword \*GEODOMAIN to specify this information. In addition, Domain #3 is an open volume containing fluid but no rock (100% porosity).



Without \*BCDOMAIN, Domains #1 and #2 are not influenced by the fluid pressure in Domain #3, since rigid “no-flow” boundary constraints are applied to the Surfaces #1 and #2. If all the (fluid) driving forces are in Domain #3, then there will be no geomechanics responses in Domains #1 and #2.

However, we do want the pressure of the fluid in Domain #3 to act as a stress boundary condition for Domains #1 and #2. Use keyword \*BCDOMAIN to specify the location of this boundary condition by referencing cells on both sides of the desired boundary surfaces.

The following data specifies that only “Surface #1” in the above diagram has a pressure boundary condition.

```
*BCDOMAIN *IJK  
** i1:i2 j1:j2 k1:k2  
3:4 1:1 1:8 ** Surface #1
```

The following data specifies that both “Surface #1” and “Surface #2” have that boundary condition. This refers to as few blocks as possible and so gives the most efficient boundary searching.

```
*BCDOMAIN *IJK  
** i1:i2 j1:j2 k1:k2  
3:4 1:1 1:8 ** Surface #1  
8:9 1:1 1:8 ** Surface #2
```

The following data specifies both “Surface #1” and “Surface #2” as well but the boundary searching will be more expensive.

```
*BCDOMAIN *ALL
```

---

## Coupling Options

\*GCOUPLING

**PURPOSE:**

Select porosity formula for coupling between flow and deformation.

**FORMAT:**

\*GCOUPLING ( 0 | 1 | 2 | 3 )

**DEFINITIONS:**

\*GCOUPLING 0

Fluid flow porosity does not depend upon deformation.

\*GCOUPLING 1

Porosity is a function of pressure, temperature and volumetric strain.

\*GCOUPLING 2 & 3

Porosity is a function of pressure, temperature and total mean stress.

**DEFAULT:**

If keyword \*GCOUPLING is absent, then \*GCOUPLING 2 is assumed.

**EXPLANATION:****Background**

Fluid flow and formation deformation (geomechanics) are coupled together in a sequential manner, that is, the two calculations alternate while passing information back and forth. The fluid flow calculation updates the pressures and temperatures over an interval specified by \*GCUPDATE. The geomechanics module updates the formation deformation in response to the new pressures and temperatures. To complete the loop, the geomechanics module sends the new deformation information back to the fluid flow calculation for use in the next time interval. It is clear that information flows from fluid flow to geomechanics via pressure and temperature. However, it is not obvious how information flows back the other way.

The fluid flow module calculates porosity as a function of pressure and temperature, in a way that pore volume and hence mass is conserved between timesteps. Here “conserved” means that the porosity at the beginning of a timestep is equal to the porosity at the end of the previous timestep, at that particular pressure and temperature. When the porosity function  $\phi(p,T)$  itself does not change with time, mass conservation across timesteps is ensured.

However, the porosity function can vary between timesteps and still conserve mass. Let  $p^n$  and  $T^n$  be the solution for a grid block for timestep “n” that used porosity function  $\phi^n(p,T)$ . The next timestep “n+1” starts with  $p = p^n$  and  $T = T^n$  but has a different porosity function  $\phi^{n+1}(p,T)$ . Porosity and hence mass will be conserved between these two timesteps if  $\phi^n(p^n, T^n) = \phi^{n+1}(p^n, T^n)$ . However,  $\phi^n$  and  $\phi^{n+1}$  may have different derivatives with respect to dependent variables  $p$  and  $T$  at  $p=p^n$  and  $T=T^n$ .

The geomechanical deformation response is expressed in the fluid flow calculation through changing parameters in the porosity function. These parameters are kept constant during timestep convergence but are updated between timesteps such that porosity and hence mass is conserved. The deformation response is accounted for on a block-by-block basis since each grid block has its own set of porosity function parameters.

Keyword \*GCOUPLING allows you to select the particular form of porosity function for the coupling of reservoir flow equations and geomechanical calculations. In the model descriptions below, superscript “n” denotes the *n*th fluid-flow update, “n-1” denotes the previous update, and “0” denotes initial conditions.

### Porosity Coupling Models

#### \*GCOUPLING 0:

Fluid flow porosity contains no parameters that depend upon deformation from geomechanics. This is “one way” coupling since fluid pressures and temperatures are still used by the geomechanics module to update formation strains and stresses. This option is useful when the fluid-flow calculations use a porosity model that cannot be approximated well by the other “two-way” coupling models. An example of this is the STARS empirical porosity model \*DILATION. See keyword \*PGDILA for a discussion on modelling dilation with the geomechanics module.

If \*PGDILA is not used then Young’s modulus is approximated from rock compressibility as:

$$E = \frac{(1-2v)(1+v)}{\phi_0 C_R (1-v)}$$

where:

E	:	Young’s modulus (kPa   psi)
v	:	Poisson’s ratio
$\phi_0$	:	Initial porosity
$C_R$	:	Rock compressibility

#### \*GCOUPLING 1

Porosity is a function of pressure, temperature and volumetric strain and has the form:

$$\phi^{n+1} = \phi^0 \left[ \frac{V_p^{n-1}}{V_p^0} + \frac{(V_p^n - V_p^{n-1})}{V_p^0 (p^n - p^{n-1})} (p - p^n) \right] - \beta(T - T_0)$$

where:

p	:	Pressure (kPa   psi)
T	:	Temperature (C   F   F)
$V_p$	:	Pore volume ( $m^3$   $ft^3$ )
$\beta$	:	Volumetric thermal expansion coefficient of the formation (1/C   1/F   1/C) given by *CTPOR
$\phi$	:	Porosity

## \*GCOUPLING 2

Porosity is a function of pressure, temperature and total mean stress formula and has the form (Tran et al, SPE/ISRM 78192):

$$\phi^{n+1} = \phi^n + (c_0 + c_2 a_1)(p - p^n) + (c_1 + c_2 a_2)(T - T^n)$$

where:

$$c_0 = \frac{1}{V_b^0} \left( \frac{dV_p}{dp} + V_b \alpha c_b \frac{d\sigma_m}{dp} - V_p \beta \frac{dT}{dp} \right)$$

$$c_1 = \frac{V_p}{V_b^0} \beta$$

$$c_2 = -\frac{V_b}{V_b^0} \alpha c_b$$

$$a_1 = \text{factor} \left\{ \frac{2}{9} \frac{E}{(1-v)} \alpha c_b \right\}$$

$$a_2 = \text{factor} \left\{ \frac{2}{9} \frac{E}{(1-v)} \beta \right\}$$

$c_b$  : Bulk compressibility (1/kPa | 1/psi)

$E$  : Young's modulus (kPa | psi) factor given by \*GCFATOR

$V_b$  : Bulk volume (m<sup>3</sup> | f<sup>3</sup>)

$\alpha$  : Biot number

$v$  : Poisson's ratio

$\sigma_m$  : Mean total stress (kPa | psi)

This porosity model is expected to give similar results to \*GCOUPLING 1 when the stress response is small but may give different results when the stress response is significant.

## \*GCOUPLING 3

Porosity is a function of pressure, temperature and total mean stress formula and has the form

$$\phi^{n+1} = \phi^n + c_0 \Delta p + c_1 \Delta T + c_2 \Delta \sigma_m$$

where  $c_0$ ,  $c_1$  and  $c_2$  are the same as for \*GCOUPLING 2.

### Example

\*GEOMECH

\*GCOUPLING 2 \*\* Porosity depends on p, T, total stress

### References

Tran, D., Settari, A. and Nghiem, L.: "New Iterative Coupling between a Reservoir Simulator and a Geomechanics Module", SPE/ISRM 78192, 2002.

## Geomechanical Coupling Factor

\*GCFATOR

### PURPOSE:

This optional multiplier factor is only used along with the second coupling formula in the geomechanical module, i.e. \*GCOUPLING 2.

### FORMAT:

\*GCFATOR *factor*

### DEFAULT:

If keyword \*GCFATOR is absent, the value of multiplier factor is one.

### DEFINITION:

*factor*

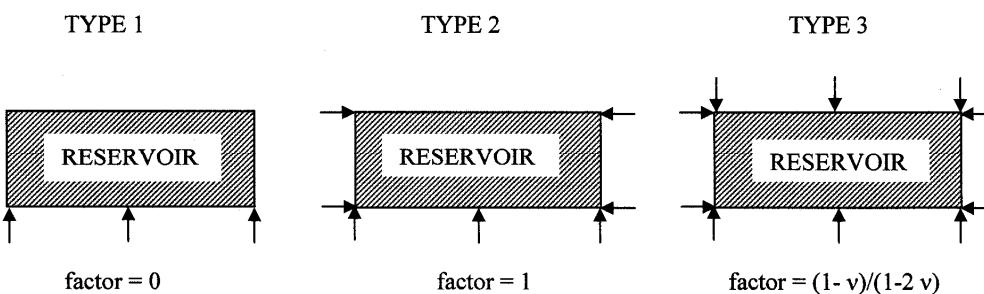
A positive real value which has a range from 0 to 10.

### CONDITION:

If the keyword \*GCFATOR appears, a value of factor must be given.

### EXPLANATION:

When the second formula of coupling (\*GCOUPLING 2) is used, the optional keyword \*GCFATOR is designed along with its value factor to handle different types of constrained boundary conditions which are assigned on a reservoir. For instance, when the value of factor is zero (by default), the reservoir is constrained only at its bottom and freely moves in other directions (Type 1). If the reservoir is constrained laterally and is free to move vertically (Type 2), the factor has a value of one. If the reservoir is constrained in all directions (Type 3), the value of factor equals to  $(1-v)/(1-2v)$ ; where  $v$  is the Poisson's ratio of porous rock. For other types of constrained boundary conditions, the value of factor may be approximated by a real number between 0 and  $(1-v)/(1-2v)$ . The above three types of boundary constraints are illustrated in 2D forms as follows:



## Pressure Tolerance Multiplier

\*GPTOLMUL

### PURPOSE:

Adjust pressure tolerance for computing porosity formulae given in keyword \*GCOUPLING.

### FORMAT

\*GPTOLMUL *multiplier*

### DEFINITIONS:

\*multiplier

A positive real number

### DEFAULT:

If keyword \*GPTOLMUL is absent, then the value of multiplier is one.

### EXPLANATION:

By default, the pressure tolerance used in computing coefficients for porosity formulae in the section of coupling options is 1 psi or 6.89475 kPa. This tolerance can be adjusted by an amount such as: tolerance = tolerance \* multiplier; where multiplier is a positive real number ranging from 1E-6 to 1E+6. As seen in porosity formulae and the coefficients in the coupling options, because there is existence of the term 1/dp, the pressure tolerance must be used to avoid the case of infinitive when dp is zero. Depending on the problem, the pressure tolerance should be adjusted to suit with the pressure change between two consecutive timesteps. For instance, if the pressure change is always less than 1 psi, value of multiplier should be less than 1.

### Example

```
*GPTOLMUL 0.5 ** which mean the pressure tolerance  
is 0.5 psi
```

---

## Coupling Update Times

**\*GCUPDATE**

**PURPOSE:**

Specify at what frequency or times that coupling updates are done.

**FORMAT:**

**\*GCUPDATE ( *freq* | \*TIME | \*TNEXT )**

**DEFINITIONS:**

**\*GCUPDATE**

Specify frequency or time of updating geomechanical conditions and the resulting porosities as specified by keyword **\*GCOUPLING**. **\*GCUPDATE** may appear more than once in a data set, perhaps with different sub-options. The action indicated by the default or specified sub-option stays in effect until the new appearance of this keyword.

*freq*

Non-negative integer *freq* indicates that an update is done for each timestep number evenly divisible by *freq*, in addition to each recurrent data time. Setting *freq* = 0 disables updating. This option may result in geomechanics updating that is unnecessarily frequent when timestep sizes are small.

**\*TIME**

Updating is done for each recurrent data time. This is equivalent to setting a large value for *freq*. This option may result in geomechanics updating that is unnecessarily frequent, especially for recurrent data containing many closely-spaced times.

**\*TNEXT**

Updating is done only for the next recurrent data time after which updating is disabled, resulting in one update per keyword occurrence. This option is useful when specifying updates at infrequent but known times in recurrent data.

**DEFAULTS:**

This keyword is initialized to *freq* = 1.

**CONDITIONS:**

This keyword is valid also in the Well and Recurrent Data section.

**EXPLANATION:**

For larger runs the timespent in performing geomechanics calculations can be a significant fraction of the total simulation run time. If geomechanical changes are small or slow compared to fluid flow changes, geomechanical coupling can be done less frequently thereby resulting in shorter total run times. Frequency sub-options **\*TIME** and **\*TNEXT** allow you to tailor a large run for optimal run times with adequate geomechanical representation.

Keyword \*GCUPDATE affects the frequency at which the deformed geomechanics grid is dumped to the SR2. At the dump times specified by \*WSRF \*GRIDDEFORM in the Input/Output Control data section, the grid is dumped only if it has been changed via geomechanics updating after the last dump time. This prevents dumping of the “same” grid more than once to the SR2. For example, suppose that you wish to dump the geomechanics grid every time that the update is done as specified by \*GCUPDATE. Do this with \*WSRF \*GRIDDEFORM 1, since the grid will not be dumped until it changes. This is much easier to specify in data than manually synchronizing \*WSRF \*GRIDDEFORM with \*GCUPDATE.

Keyword \*GCUPDATE affects the frequency of dumping geomechanics quantities specified by \*OUTSRF \*GRID to the SR2. At dump times specified by \*WSRF \*GRID, these geomechanics quantities are dumped only if the geomechanical response has been updated since the last dump time. This prevents dumping a geomechanics solution that is out-of-date. The dumping of non-geomechanics quantities does not depend upon \*GCUPDATE.

Keyword \*GCUPDATE affects the frequency at which geomechanics quantities are updated for special histories like \*OUTSRF \*SPECIAL \*BLOCKVAR. But special histories are assigned every timestep and interpolation is not done between update times. Therefore, less frequent geomechanics updates may result in a corresponding special history plot that exhibits artificial “stair-step” behavior. In this case it may be more appropriate to use the RESULTS Graph source option “Add Block Property Versus Time” which will not attempt to plot values between the actual dump times.

### Example

```
*GCUPDATE 10      ** Update every 10 timesteps.  
*GCUPDATE *TIME   ** Update each *TIME/*DATE.  
*GCUPDATE *TNEXT  ** Update only next *TIME/*DATE.
```

Example: Update at known but infrequent times.

```
*TIME 100  
*TIME 110  
... many well changes from history match  
*TIME 245  
  *GCUPDATE *TNEXT  
*TIME 250  ** Geomechanics update time  
*TIME 260  
...
```

---

## Porosity Calibration

\*CALIB\_POR

### PURPOSE:

Improve accuracy of porosity coupling option \*GCOUPLING 2.

### FORMAT:

\*CALIB\_POR

### DEFAULTS:

If the keyword does not appear, there is no porosity calibration.

### CONDITIONS:

This keyword applies to all geomechanics rock types.

This keyword is used only with \*GCOUPLING 2.

### EXPLANATION:

The purpose of this option is to improve the accuracy of porosity computed from a second order formula of \*GCOUPLING 2 given in the EXPLANATION for \*GCOUPLING. Since the current reservoir porosity is computed on the basis of porosity coefficients which were estimated in the geomechanics module in a previous timestep, the reservoir porosity may not be as accurate as desired. Because of the explicit calculation of those coefficients and the approximate nature of the porosity function, there is a difference between the porosity calculated in STARS simulator and the actual porosity calculated by GEOMECH. The difference could become substantial, especially for plastic deformation and shear dilation.

After the reservoir porosity is determined, it is compared to the actual porosity which is computed in the geomechanics module. The actual porosity at timestep  $n$  is defined here as a ratio between a pore volume at timestep  $n$  and the initial bulk volume of a grid block. The difference between the reservoir porosity and actual porosity at the end of one timestep will be used to calibrate the next timestep porosity through a weighting function. The main goal of this option is to bring reservoir porosity close to the corrected porosity gradually at the later timesteps.

The porosity formula used in \*GCOUPLING 2 in the Coupling Options Section can be written in an extended form as:

$$\phi_{n+1} = \phi_n + (c_0 + c_2 a_1)_n (p - p_n) + (c_1 + c_2 a_2)_n (T - T_n) + \omega^{(k)} (\phi_n^c - \phi_n)$$

where:

$$\phi_n^c = \frac{V_n^p}{V_0^b} : \text{Actual porosity based on the definition}$$

$V_n^p$  : Pore volume at timestep  $n$

$V_0^b$  : Initial bulk volume

$\omega^{(k)}$  : Weighting function at  $k^{\text{th}}$  iteration is expressed as:

$$\omega^{(k)} = \alpha_p f(\varepsilon_p^k) + \alpha_T f(\varepsilon_T^k)$$

Where:

$$f(\varepsilon_p^k) = 1 - \exp \left( -\frac{1}{1 - \frac{|p_{n+1}^{(k)} - p_{n+1}^{(k-1)}|}{p_n}} \right)$$

$$f(\varepsilon_T^k) = 1 - \exp \left( -\frac{1}{1 - \frac{|T_{n+1}^{(k)} - T_{n+1}^{(k-1)}|}{T_n}} \right)$$

$$\alpha_p = \frac{(c_0 + c_2 a_1)_n (p_n - p_{n-1})}{(c_0 + c_2 a_1)_n (p_n - p_{n-1}) + (c_1 + c_2 a_1)_n (T_n - T_{n-1})}$$

$$\alpha_T = \frac{(c_1 + c_2 a_1)_n (T_n - T_{n-1})}{(c_0 + c_2 a_1)_n (p_n - p_{n-1}) + (c_1 + c_2 a_1)_n (T_n - T_{n-1})}$$

### Example

\*GCOUPLING 2

\*CALIB\_POR

---

## Iterative Coupling to Fluid Flow

**\*STRESSTOL, \*POROSTOL**

**\*NCOUPLING, \*PRESSTOL,**

### PURPOSE:

Specify parameters that control iterative coupling to the fluid-flow solution.

### FORMAT:

<b>*NCOUPLING</b>	<i>ngeo_iter</i>
<b>*PRESSTOL</b>	<i>pres_tol</i>
<b>*STRESSTOL</b>	<i>stress_tol</i>
<b>*POROSTOL</b>	<i>poros_tol</i>

### DEFINITIONS:

#### **\*NCOUPLING** *ngeo\_iter*

Maximum number of geomechanics updates performed per fluid-flow timestep, where *ngeo\_iter* is an integer greater than or equal to 1. If the multi-update option is used then normally *ngeo\_iter* is given a large number (e.g., 20) so that an adaptive convergence criterion like \*PRESSTOL can operate freely.

#### **\*PRESSTOL** *pres\_tol*

Maximum allowed pressure difference allowed for each grid block, between consecutive Newton cycles, before the couple geomechanics-fluid-flow solution is deemed converged. The unit of *pres\_tol* is (kPa | psi) and the allowed range is 0 to 10 kPa (0 to 1.45 psi).

#### **\*STRESSTOL** *stress\_tol*

Maximum allowed stress difference allowed for each grid block, between consecutive Newton cycles, before the couple geomechanics-fluid-flow solution is deemed converged. The unit of *stress\_tol* is (kPa | psi) and the allowed range is 0 to 10 kPa (0 to 1.45 psi).

#### **\*POROSTOL** *poros\_tol*

Maximum allowed porosity difference allowed for each grid block, between consecutive Newton cycles, before the couple geomechanics-fluid-flow solution is deemed converged. The quantity *poros\_tol* is dimensionless and the allowed range is 0 to 1.0.

### DEFAULTS:

If keyword \*NCOUPLING is absent then *ngeo\_iter* = 1 is assumed.

If *ngeo\_iter* > 1 and keywords \*PRESSTOL, \*STRESSTOL and \*POROSTOL are absent, the criterion is pressure difference with *pres\_tol* = 0.01 kPa (0.00145 psi).

**CONDITIONS:**

At most one of \*PRESSTOL, \*STRESSTOL and \*POROSTOL is used. If more than one of these keywords is appears, the last one is used.

**EXPLANATION:**

In most cases only one geomechanics update is performed per timestep, which usually is sufficient for convergence and acceptable results. However, some geomechanics problems such as arching or Mandel's effects may require more than one geomechanics update to obtain a satisfactory result. In a timestep, each geomechanics update occurs before a fluid-flow Newton cycle. The number of such updates depends upon the parameters specified by these keywords as well as the number of fluid-flow Newton iterations (\*NEWTONCYC).

## Boundary Stress Unloading

\*UNLOADSTR

### PURPOSE:

Assign total stress to be unloaded at the well boundary.

### FORMAT:

```
*UNLOADSTR  
    kl      (:k2)      stress  
    :        :         :  
    :
```

### DEFINITIONS:

\*UNLOADSTR

This keyword turns on the single-well boundary stress unloading option.

kl(:k2)

Range of grid layers where boundary stress unloading applies.

stress

Radial boundary stress to be unloaded (kPa | psi).

### DEFAULTS:

Required keyword for the single-well boundary stress unloading option. There are no defaults.

### CONDITIONS:

Boundary stress unloading option works only with a radial grid. Cartesian grid is not allowed.

### EXPLANATION:

This keyword allows the user to specify the amount of external boundary stress to be unloaded at the wellbore boundary. The actual radius where the unloading occurs is specified by the \*WRADIUS keyword. This unloading may occur as a result of fluid and sand production close to the wellbore. A large reduction in external boundary support stress may lead to tension failure for the elements adjacent to the well.

A scheme for removing elements which is shown to fail in tension has NOT been implemented. Therefore, it is advisable to run the model with some cohesion in this case to avoid stability problems. As well, consideration of sand flow with petroleum fluid is NOT considered in this model. These will be considered as a part of CMG's ongoing research in the area.



# Well and Recurrent Data

## Summary of Well and Recurrent Data

The section contains data and specifications which may vary with time. The largest part is well and related data, but there are keywords which define other time-dependent information.

### Required Keywords

The following are the minimum required keywords in this section:

```
*RUN
*TIME or *DATE    ** Starting time
*DTWELL           ** Starting timestep size

*WELL              ** Well definition (at least one set)
*INJECTOR or *PRODUCER
*INCOMP (injector only)
*TINJW (injector, thermal only)
*OPERATE
*PERF or *PERFV

*TIME or *DATE    ** Stopping time
```

### Critical Keyword Ordering

**\*RUN**

This must be at the start of the recurrent data.

**\*TIME, \*DATE**

The keyword after **\*RUN** must be one of these. The times must be increasing. If **\*DATE** is to be used anywhere, it must appear also immediately after **\*RUN**.

**\*WELL**

A well must be defined via **\*WELL** before anything can be done with it.

**\*INJECTOR, \*PRODUCER, \*SHUTIN, \*OPEN**

The current well type list must be defined before any of the keywords that use it.  
**\*INJECTOR** and **\*PRODUCER** can be entered after **\*PERF** but need not be.

\*INCOMP, \*TINJW, \*QUAL

\*OPERATE, etc.

There is a list of keywords specifying well attributes that are applied to the current well type list.

\*MONITOR

This keyword must appear after the corresponding \*OPERATE.

\*GEOMETRY, \*PERF, \*GEO, \*GEOA, \*KH, \*KHA

The radial inflow model parameters must be specified before they are used.

\*PERF may be entered before \*PRODUCER and \*INJECTOR but need not be.

## Well Identifiers

The keywords which control well data can be very flexible and powerful once you understand how to use well names and well lists.

1. Each well has a distinct well name assigned to it by the keyword \*WELL. The well will be referenced by this name later in the data, and is reported by this name in the simulator output. Each well must be defined with \*WELL before anything can be done with it.
2. Some keywords (e.g., \*PERF) refer to a single well via its quoted well name.
3. Some keywords (e.g., \*HEAD-METHOD) may refer to more than well via a list of quoted well names. When a well list is allowed, wildcarding of well names can help specify a series of wells based on similarities in their names. See **Wildcarding Well Names**, below. With one exception, a keyword's well list is "local" to that keyword and is not used for any other purpose.
4. Well type keywords \*INJECTOR, \*PRODUCER, \*SHUTIN and \*OPEN define a well list that is saved and then used by the following keywords:

*PHWELLBORE	*TINJW	*QUAL
*PINJW	*INCOMP	*OPERATE
*MONITOR		

until it is overwritten by another well type keyword's well list. For example, to assign injection temperature 350 deg and quality 70% to two injectors, use

```
*INJECTOR 'Injector 1' 'Injector 3'  
*TINJW 350  
*QUAL .7  
*INCOMP *WATER w(1) ..... w(numx)  
*OPERATE .....
```

## Wildcarding Well Names

There are two wild-card characters, "?" and "\*". Only lists of well names, and not lists of group names, can use the wild-carding. When wild-card characters are used, a list of the wells matched is printed in the output file so that the user may check the list generated.

Any number of ‘?’ can appear in a well name character string. Each ‘?’ matches any character in the same position in a well name, including embedded blanks but not final blanks. For example, ‘WELL??’ matches ‘WELL 1’ but ‘WELL?’ does not match ‘WELL’.

A single ‘\*’ can appear as the last non-blank character in a well name character string. When ‘\*’ is present only the characters preceding ‘\*’ are checked for lack of match. For example, the character string ‘\*’ matches all wells, and ‘WELL\*’ matches ‘WELL’.

Wild characters ‘?’ and ‘\*’ can appear in the same string. For example ‘WELL????PROD\*’ would match ‘WELL\_NW\_PROD\_15’ and ‘WELL\_SE\_PROD\_2’.

### Well Fraction

There are two kinds of well fraction, and each one has a different function.

1. The keyword \*WELL has the subkeyword \*FRAC which directs the simulator to use internally a fraction of the well rates and indices that appear in the input data. The most common usage is with symmetry patterns, where a well on the boundary is a known fraction of a full well. The familiar full-well rates and indices can be entered and \*FRAC specifies the fraction (e.g., 1/2 on the side, 1/4 in a 90 deg corner, 1/8 in a 45 deg corner).

The definition of symmetry elements may involve the use of geometry block modifiers (\*VAMOD) and completion fractions ( \*PERF \*GEO ff ).

2. The keyword \*GEOMETRY defines information used to estimate well index given the configuration of the well and its grid block (Appendix A.1). This is flagged by the \*GEO or \*GEOA options in \*PERF and \*PERFV. If you are not using the \*GEO or \*GEOA option, then you will not use ‘wfrac’.

If \*GEO or \*GEOA is chosen, the most common usage is to define \*GEOMETRY for a full well in the centre of a block in a repeated grid (Figure A.1(a), wfrac = 1), and then to apply the well fraction \*FRAC as described above. This method should be used if the well is on the boundary of a symmetry element (see Figures 6, 7 and 8 in the Reservoir Description section).

Configurations from Figure A.1 other than (a) are meant for non-symmetry element cases, or where the well is not in the center of the grid block. The same comments apply to the radial grid: use Figure A.1a, but set \*FRAC to the fraction of a full circle.

### Perforation Options

There are three options for assigning well perforation locations, and three options for assigning well indices. These options may be mixed.

1. \*PERF is the normal option, requiring the I-J-K locations for a single well.
2. \*PERFV can be used to assign the same K layers to a list of vertical wells whose I-J locations were defined in their \*WELL definitions.
3. The default well index option is to read the value directly for each layer.
4. An option is available for obtaining the well index from assuming a linear pressure drop between the block centre and the block face; this is used for tube-end situations and discretized wellbore completions.

The sub-options \*GEO and \*GEOA indicate that the geometric part of the well index is estimated with information from the current \*GEOMETRY keyword (radial inflow model or linear-pressure-drop). The inflow model uses completion fraction ff and near-well permeability wlprm for each layer, which may vary with layer.

5. Limited Entry Perforations (LEP) can be assigned to wells via \*LEP-WELL.
6. Multiple active wells are allowed in a grid block. It means that different wells may have the same I-J-K location.

### Liquid Level Control of Pumped-Off Wells

A production well that is pumped off to atmosphere often is operated on a liquid level control. This modeled by using keyword \*LAYERGRAD to force the use of gas-like head gradient above the liquid level, and liquid-like value below. In addition, you can control the position of the liquid level via keyword \*BHPDEPTH. These keywords effectively replace the modelling of pumped-off wells that was done with the obsolete \*PUMPOFF sub-option of \*OPERATE \*BHP.

### Specifying Injected Phase

Use \*INCOMP to specify which phase(s) to inject. You may inject either one phase or co-inject steam with solvent which could be in oil or gas phase or both at STC per well.

For example,

```
*INCOMP *GAS 0 0 0 .79 .21      ** (79% component 4 and 21%
                                         ** component 5)
*OPERATE *STG 10000             ** rate in m^3/day
```

causes gas phase to be injected at 10000 m<sup>3</sup>/day, whereas

```
*INCOMP *GAS 0 0 0 .79 .21      ** (79% component 4 and 21%
                                         component 5)
*OPERATE *BHP 2000              ** compressor rated at 2000 psi
```

cause gas phase to be injected at 2000 psi.

\*INCOMP must be specified for each injector. When steam injection is not specified as cold water equivalent (CWE) but as a specified mixture of water and gas then use two wells, one injecting hot water with zero quality and the other one injecting water component in gas phase. For example,

```
*WELL 1 'INJWATER'
*INCOMP WATER 1 0 0    ** Liquid water (component #1)
*QUAL 0.0
*TINJW 250.0
*PINJW 3973
*OPERATE STW 100.0    ** water injection rate

*WELL 2 'INJGAS'
*INCOMP GAS 1 0 0     ** Gaseous water (component #1)
*TINJW 250.0
*PINJW 3973
*OPERATE STG 1.19E6   ** Represents 90% quality if water density
                                         ** is 5.55e4 and gas density 41.9
                                         ** gmole/m^3 at surface conditions
```

## **Well Group Control**

The definition of a well group hierarchy (gathering centres) and group control is optional. If a group control hierarchy is used, not all wells must be attached explicitly to groups. For more information about well groups, see “Well Management and Group Control” in the Tutorial section, as well as the EXPLANATION for keyword \*GROUP.

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 the apportionment method keyword \*APPOR-METHOD. The most offending well can be shut by using the \*SHUTIN action under the \*GCONM keyword.

Wells can be drilled automatically to maintain production or injection targets by specifying their initial status as \*AUTODRILL and by specifying \*GAPPOR 'group' \*AUTODRILL \*ON.

## **Grouping Well Statistics**

Arbitrary groups of wells can be defined with weighting factors for output and SR2 with the keywords \*REPORTING-GROUP. A well may appear in more than one group, and weighting factors may be applied. This type of well group is not related to the well group hierarchy defined by \*GROUP.

## **Heater Options**

There are six different models for specifying heater or heat loss control:

1. Constant Model

Keyword \*HEATR allows assignment of a heat gain (+) or loss (-) rate on a per block basis.

2. Convective Model

Keyword \*UHTR allows assignment of a heat gain (+) or loss (-) proportional rate coefficient on a per block basis; heat transfer rate will depend also on a temperature setpoint defined via \*TMPSET. Keywords \*UHTRAREA1-, etc., assign the rate coefficient on a per area basis.

3. Automatic Switching

Keywords \*AUTOHEATER and \*AUTOCOOLER allow combination of \*HEATR and \*UHTR to create a heater/cooler control that operates on a constant rate below/above a given temperature.

4. Adiabatic Model

Keyword \*ADHEAT allows assignment of data to model a heat gain rate which depends on the difference in temperature between a heater block and a reference block.

5. Slaved Model

Keyword \*HEATSLAVE allows assignment of data to model a heat gain rate in a “slave” block which depends on the rate in another “master” block.

## 6. Heater Well

Keyword \*HTWELL allows you to specify heat-only gain or loss in the same locations as a fluid well defined by keyword \*WELL.

Note that these heater options are not related to the Electrical Heating option (see keyword \*ELECHEAT in the Other Reservoir Properties chapter as well as Appendix G). It is possible, but not recommended, to use \*ELECHEAT together with one or more of the above heater options.

### Keywords from Other Sections

The following is a list of keywords from other sections which may appear in recurrent data.

Input/output Control:	*MAXERROR, *PRINT_REF, *WRST, *REWIND, *WSRF, *OUTSOLVR, *OUTPRN, *WPRN, *OUTSRF *GRID
Reservoir Description:	*THTYPE, *TRANSI, *TRANSJ, *TRANSK, *TRANLI, *TRANLJ, *TRANLK, *TRANSIENT
Other Reservoir Properties:	*TRANSIJ+ *TRANSIJ-, *TRANSIK+, *TRANSIK+
Component Properties:	*VSTYPE
Rock-Fluid Properties:	*KRTYPE, *KRTYPE_VERT, *BSWR, *BSWCRT, *BSORW, *BSOIRW, *BSGR, *BSGCON, *BSORG, *BSOIRG, *BSWRG, *BSWIRG, *BKRWIRO, *BKROCW, *BKRCGW, *BPCWMAX, *BPCGMAX
Numerical Methods:	*MAXSTEPS, *DTMAX, *DTMIN, *NUMSET, *NORM, *CONVERGE, *MATBALTOL, *NEWTONCYC, *UNRELAX, *UPSTREAM, *PRECC, *NORTH, *PIVOT, *ITERMAX, *AIM, *PVTOSCMAX, *NCUTS

Remember, when using the array-reading option \*IJK to assign grid arrays (such as \*KRTYPE), that you are allowed to refer to only select grid blocks if you wish, instead of the entire grid (as in the other data sections). See the description for array-reading option \*IJK.

### Fine-Grid Inheritance

In the initialization data sections, inheritance is performed after all the initialization data is read. Data assigned explicitly to refined blocks via array qualifier \*RG will override values that could have been inherited from the parent block, independent of the order of appearance of the parent and child grid keywords. For example, assume block (1,2,1) is refined. The keyword data in the Initial Conditions section

```
*TEMP *CON 70  
*TEMP *RG 1 2 1 *CON 100
```

and the data (in reversed order)

```
*TEMP *RG 1 2 1 *CON 100  
*TEMP *CON 70
```

have the same result, that is, a value of 100 in all the fine blocks in (1,2,1) and 70 everywhere else. Here, data assigned via \*RG takes precedence.

On the other hand, inheritance in the recurrent data section is slightly different. All ARRAY format data have valid initial values either carried from initialization sections (such as \*KRTYPE) or indicating a disabled option (such as \*HEATR). Inheritance is performed immediately upon reading ARRAY format data, and the result may depend upon the order of appearance of keywords. Using the previous example, the keyword data

```
*UHTR *CON -2  
*UHTR *RG 1 2 1 *CON -5
```

will result in the value -5 for the fine blocks in (1,2,1) and the value -2 everywhere else. On the other hand, reversing the order

```
*UHTR *RG 1 2 1 *CON -5  
*UHTR *CON -2
```

will result in a value of -2 everywhere since the second line causes an immediate inheritance of -2 to the fine blocks, overwriting the result of the first line. Here, the order of keywords takes precedence.

A good strategy is to specify such keyword data in the order of fundamental grid first and successively finer grids after if the values differ from the fundamental grid.

Immediate inheritance is done also for transmissibility multipliers \*TRANSI, etc.

### Inheritance of Connection-Based Quantities

There is no inheritance of connection-based quantities \*PTRANSx. Therefore, you must assign values explicitly to refined blocks instead of relying on inheritance.

### Compatibility with IMEX Recurrent Data

STARS can use the following keyword data from IMEX without change:

*DATE	*TIME	*DTMAX	*SHUTIN
*WELL	*NULL-PERF	*PRODUCER	*LAYERIJK
*OPEN	*AUTODRILL	*BHPGRAD	*LAYERGRAD
*ALTER	*BHPDEPTH	*DRILLQ	*LAYERXYZ
*MRC-RESET	*GCONN	*STOP	*GAPPOR
*GCPOFF	*GCIOFF	*GROUP	*DTMIN

STARS can use the following keyword data from IMEX with the noted differences. IMEX keywords that do not appear in these tables are not supported in STARS.

*DTWELL	The default is different. IMEX value persists for later times but STARS value applies only to the next time.
*AIMSET	STARS default is fully implicit; IMEX is IMPES.
*HEAD-METHOD	*ZERO-HEAD not supported; use *BHPGRAD to assign zero head.
*INJECTOR	STARS default is *UNWEIGHT. For *MOBWEIGHT STARS uses explicit mobility by default but can use implicit; IMEX uses implicit.
*IWELLBORE	Use *PHWELLBORE *SAMODEL or *TABLE.

*PWELLBORE	Use *PHWELLBORE *SAMODEL or *TABLE.
*INCOMP	The syntax for referring to components is different.
*OPERATE	Not supported: constraint types *STS and *BHS; *PENALTY; subkeyword *INITIALIZE of *WHP.
*MONITOR	Not supported: constraint types *SOR, *GSOR and *STS; action *RECOMPLETE.
*GLIFT	*RATE only. Specify gas composition *INCOMPGL.
*GEOMETRY	Use geofac values from Appendix A.1, or divide IMEX geofac by $\sqrt{[(d1*d1+d2*d2)/(d1*d2)]}$ where d1 and d2 are block sizes normal to well direction (factor = $\sqrt{2}$ when d1=d2).
*PERF & *PERFV	Only options *WI, *GEO, *GEOA, *KH, *KHA, *FLOW-TO and *FLOW-FROM are supported. Only parameters wi, ff and kh are supported. STARS *GEO and *GEOA both correspond to IMEX *GEOA; the same comments apply to *KH and *KHA.
*GCONP, *GCONI	*STS and *SOLVENT not supported.
*APPOR-METHOD	*SOLI not supported
*PRIOR-FORM	*SOLI not supported
*GUIDEP, *GUIDEI	*STS not supported.

### Converting Data from Versions Before 2002

Between versions 2001 and version 2002 the well definition data underwent significant change caused by STARS adopting the Well Management module used by the other CMG simulators. This measure makes available to STARS an impressive list of new well control options. The data change involved is done automatically by the version 2002 data pre-processor.

The following instructions show how to do the changes manually. These instructions assume that you wish to reproduce the behavior given by versions before 2002, and are not using newly available options. Of course, after the data is converted you may try the new options (e.g., well group control).

**Note:** Even if data is converted simply to the new syntax, the well behavior may not be the same as given by pre-2002 versions. This is because some details of well control in Well Management are different from the previous behavior in STARS.

1. \*INCOMP is mandatory for injectors, and must be added before \*OPERATE if it is absent.
2. Optional keywords \*TINJW and \*QUAL must appear before \*OPERATE, not after.
3. After \*GEOMETRY the direction indicator ( \*I | \*J | \*K ) is mandatory, and must be added if it is absent.
4. When using the \*GEO option of \*PERF and \*PERFV, the quantity "ff" is mandatory. If it was absent use default value of "1".

5. Indicate the linear pressure drop option with the \*TUBE-END sub-keyword of \*PERF and \*PERFV. Also, change “rad” after \*GEOMETRY to some positive value to satisfy keyword syntax. If the perforation is a discretized wellbore block, the linear pressure drop option is used internally even if \*GEO is indicated.
6. When using the \*GEO option of \*PERF and \*PERFV to specify well permeability “wlprm”, use the \*KH sub-option instead with “kh” = “wlprm” times the block thickness in the well direction. This may require splitting up a common definition for a list of wells if the block thicknesses vary between wells.
7. The well list defined by keywords \*INJECTOR and \*PRODUCER is not retained across \*TIME and \*DATE lines. Therefore, keywords \*INCOMP, \*TINJW, \*PINJW, \*QUAL and \*OPERATE must come after the corresponding keywords \*INJECTOR and \*PRODUCER within a time segment.
8. For \*ALTER the new values must fall on a new line, and may not be on the same line as the well names.
9. \*PERF or \*PERFV must not occur without the corresponding \*OPERATE.
10. The wellbore heatloss model enabled by \*HEATLOSS is no longer available. Use \*PHWELLBORE instead.
11. For \*PERF and \*PERFV the UBA qualifiers “WB” and “TU” are not supported. For a non-circulating wellbore change “WB” to “/ 1 1”. For a circulating wellbore change “WB” to “/ 2 1 1” and “TU” to “/ 1 1 1”.
12. \*SAMINFO is no longer a sub-keyword of \*PHWELLBORE, but one occurrence controls output for all wells using \*PHWELLBORE.
13. Keyword \*WIOLD is no longer available.
14. Specify non-default depth of the BHP position via primary keyword \*BHPDEPTH instead of the \*PERF (and \*PERFV) sub-keyword \*BHPDEPTH.
15. Specify non-default head gradient in a well via primary keyword \*LAYERGRAD instead of the \*PERF (and \*PERFV) sub-keyword \*HEADGRAD.
16. Specify well constraint choosing option via \*MRC-RESET instead of \*CONSTRNCHK.
17. The \*OPERATE remedial action \*CONT in previous versions actually corresponds to \*CONT \*REPEAT in v2002. Therefore, to reproduce the previous behavior, add \*REPEAT after \*CONT.
18. Keyword \*INCOMPGL syntax is changed.
19. If old keyword \*GROUPWT was used, merge the \*GROUP and \*GROUPWT definitions into \*REPORTING-GROUP.
20. Sub-keyword \*KRCYC of cyclic group keywords \*INJ\_C\_SWT and \*PROD\_C\_SWT is no longer available.

---

## **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.

---

## Simulation Reference Times

\*TIME, \*DATE, \*STOP

### PURPOSE:

Control start, stop, well change and printout times.

### FORMAT:

\*TIME *time*  
\*DATE *yyyy mm dd*  
\*STOP

### DEFINITIONS:

*time*

Simulation reference time (days | days | mins).

*yyyy mm dd*

Year, month and day of the reference time. The time-of-day is given by the decimal fraction of dd from midnight. For example, a well change at noon of August 19, 1988, is entered as: \*DATE 1988 08 19.5. Only years from 1901 to 5000000 are allowed. In the conversion between years and days, leap years are accounted for correctly only up to 2099.

### DEFAULTS:

If the first reference time is specified with \*DATE, then the corresponding time is 0.

If \*STOP is absent from the data file, then the run will terminate after the last reference time.

### CONDITIONS:

A reference time must appear immediately after the \*RUN keyword.

Recurrent data must be organized into segments, where each segment has a beginning reference time, some recurrent data and an ending reference time, in that order. A good practice is to left-justify the reference time keywords while indenting recurrent data that falls between.

If the first reference time was specified with \*TIME then \*DATE is not accepted in the remaining recurrent data.

Reference times must increase, going in sequence from \*RUN to the end of the data.

\*STOP, if present, should be placed immediately after the corresponding reference time (\*TIME or \*DATE).

\*STOP must not appear immediately after the first reference time, since at least two time points are required to specify one recurrent data segment.

Time on 2 consecutive \*TIME cards must be  $> 10^{-5}$  days for SI and Field units and  $10^{-10}$  days for LAB units.

## **EXPLANATION:**

To indicate a reference time you must use \*TIME or \*DATE.

A change to data corresponding to a recurrent data keyword is deemed to happen immediately after the reference time preceding the keyword. For example,

```
*TIME 100
  *OPERATE *STO 100
*TIME 200
```

causes the rate to be 100 in the interval from 100 to 200 days. Be careful of keywords whose data apply to points in time instead of time intervals.

For example,

```
*WRST 0          ** I/O Control section
.
.
*TIME 100
  *WRST *TIME
*TIME 200
```

causes the first restart to be written at 200 days since \*WRST \*TIME is applied after 100 days. \*STOP is the only keyword which does not strictly follow this rule.

\*TIME or \*DATE may be defined at a time with no recurrent data change, causing the simulator to obtain a solution at this time, perform any output and then proceed.

Example:

Start producing oil at 750 bbl/day, get results at 150 days and stop at 365 days.

```
*TIME 0
  *PRODUCER 2
  *OPERATE *MAX *STO 750
*TIME 150
*TIME 365 *STOP
*TIME 900
```

Restart from 150 days with 500 bbl/day, obtain solution at 365 days; and stop at 900 days.

```
*TIME 0
  *PRODUCER 2
  *OPERATE *MAX *STO 750
*TIME 150
  *ALTER 2 500.
*TIME 365 **STOP      ** *STOP commented out
*TIME 900 *STOP        ** *STOP is optional
```

\*STOP causes the simulation to terminate after the solution is obtained, at the corresponding reference time.

\*STOP may appear any number of times in your date, but only the first one encountered is used and all data after the reference associate with that \*STOP is ignored. For a restart run, any \*STOP associated with a time before the restart time will be ignored.

If other keywords appear between reference time and \*STOP, then the simulation will stop at the next reference time.

**Example**

```
*TIME 2555.0  
*STOP  
*TIME 3500
```

---

## Simulation Pause

\*PAUSE

### PURPOSE:

Pause simulator execution temporarily to allow another application to run.

### FORMAT:

\*PAUSE *nsec pause\_file*

### DEFINITIONS:

*nsec*

Length of the pause cycle in seconds.

*pause\_file*

Quoted name of the pause file.

### DEFAULTS:

If \*PAUSE is absent then no pausing is done.

### CONDITIONS:

If \*PAUSE is present then it is assumed that the file named *pause\_file* is deleted by the other application when it is finished each of its run segments.

This keyword should be placed immediately after \*TIME/\*DATE or \*STOP if present.

At most one \*PAUSE keyword is allowed per segment of recurrent data.

### EXPLANATION:

\*PAUSE is particularly useful before an \*INCLUDE keyword that specifies an auxiliary data file that is to be created or updated by another application. For example, the other application may obtain information from STARS output files up to the pause time and generate new data for STARS via an \*INCLUDE statement.

When \*PAUSE is encountered STARS takes the following steps in this order:

1. A file named *pause\_file* is created.
2. STARS pauses for *nsec* seconds.
3. If file *pause\_file* still exists, control passes to the previous step. If file *pause\_file* is absent, control passes to the next step.
4. STARS continues execution of the simulation.

Steps 2 and 3 together effectively put STARS into pause mode until the other application is finished. The pause is an OS-level “sleep” command that causes STARS to use almost no CPU while it is paused. The purpose of *pause\_file* is to signal that the other application may execute. The other application may be in pause mode itself while STARS runs, periodically testing for the existence of *pause\_file*. The other application must delete *pause\_file* when it is finished, signaling that STARS may continue the simulation.

The pause option can support more than one other application in the same run, merely by using different pause file names. Also, the signaling file name is flexible. For example, if the other application has name “xxx” then *pause\_file* could be named ‘xxx.run’.

## Dimension Scanning and Run-Time Data Generation

Some options cause additional storage to be allocated after the “dimension scan” phase of data reading. The entire data set is scanned and some storage is allocated only if certain keywords are present. For example, internal arrays for the heater options are allocated storage only if the corresponding keywords like \*HEATR are present.

When using the \*PAUSE option to generate data at run time, make sure that keywords that will be generated later in the run are present in some form during the dimension scan phase. For example, if \*HEATR data is generated later in the run then put \*HEATR \*CON 0 in the first segment of recurrent data before any other \*HEATR usage. In the example below, the run should start with file “heater.dat” containing \*HEATR \*CON 0 instead of the file being empty.

### EXAMPLE:

Assume we want another application to update an auxiliary file called “heater.dat”, containing \*HEATR keywords, several times a month. The other application looks for file ‘stars.pause’, then reads STARS output files, generates some new heater data, writes it in STARS keyword format into ‘heater.dat’, and finally deletes file ‘stars.pause’. The pause time of 60 seconds reflects the fact that each update could take several minutes. A segment of recurrent data might look like:

```
*TIME 0
    ... well specifications
    *INCLUDE 'heater.dat' ** Original heater data
*TIME 10
    *PAUSE 60 'stars.pause'
    *INCLUDE 'heater.dat' ** Data updated automatically
*TIME 20
    *PAUSE 60 'stars.pause'
    *INCLUDE 'heater.dat' ** Data updated automatically
    ... well changes
*TIME 31
    *PAUSE 60 'stars.pause'
    *INCLUDE 'heater.dat' ** Data updated automatically
*TIME 46
    *PAUSE 60 'stars.pause'
    *INCLUDE 'heater.dat' ** Data updated automatically
*TIME 61
```

Note that the schedule of updates is pre-determined so that the recurrent data can be constructed. However, the content of ‘heater.dat’ is entirely dynamic. For example, different blocks could be referenced at different times, corresponding to heaters being added or removed as conditions in the reservoir evolve. Also, different auxiliary data file names could be used at different times after the \*INCLUDE keyword, or multiple \*INCLUDE keywords could be specified at some or all the pause times.

---

## Simulation Reference Times

\*DTWELL

### PURPOSE:

Provide starting timestep size.

### FORMAT:

\*DTWELL *time\_size*

### DEFINITIONS:

*time\_size*

Size of the timestep immediately following the reference time (days | days | min).

### DEFAULTS:

If \*DTWELL is absent, then the timestep size after \*TIME or \*DATE is carried from the previous timestep.

### CONDITIONS:

\*DTWELL is required between the first and second reference times, after which it is optional.

### EXPLANATION:

Use \*DTWELL to specify the size of the first timestep after the previous reference time. For example,

```
*TIME 100  
*DTWELL 0.01  
...  
*TIME 200
```

## **Group Identification (Optional)**

**\*GROUP**

### **PURPOSE:**

\*GROUP is used to identify gathering centres, groups and platforms. The information entered with this keyword is used to build a tree structure of groups.

\*GROUPWT is obsolete. Use \*REPORTING-GROUP instead.

### **FORMAT:**

\*GROUP 'child\_1' ... 'child\_n' \*ATTACHTO 'parent'

### **DEFINITIONS:**

'child\_1', 'child\_2' .. , 'child\_n'

Names of child groups that are attached to the 'parent' group. Each group is identified by a unique name up to 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:**

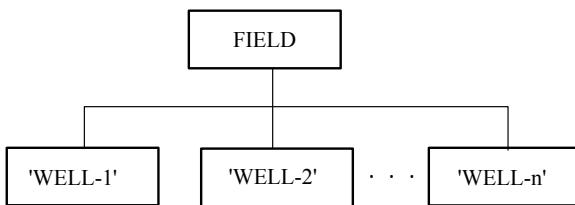
Optional keyword. If no \*GROUP line appears in the data, no group structure exists in the simulation and well rates and cumulatives are summed directly into a field cumulative. This is reported as the FIELD cumulative in output, but no group called 'FIELD' actually exists in this case and no group control 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:

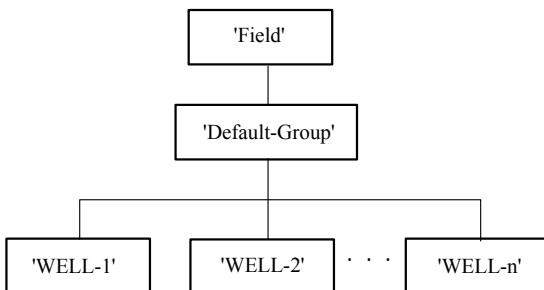


Quotes were purposely omitted around FIELD above to emphasize that no group with the name 'FIELD' actually exists in this case. No group control is possible in this case.

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. When the simulator encounters a \*GROUP keyword, it continues to look for \*GROUP keywords until a different keyword is found. It then constructs a set of hierarchical group pointers. When these pointers are constructed, all group controls (e.g. those entered using the \*GCONN, \*GCONP, or \*GCONI keywords) for the pre-existing group hierarchy cease to be in force. Thus if the user intends to add a group to the hierarchy after the beginning of simulation, all group controls must be re-established (if the intention is for them to remain in force) by re-entering them on the appropriate data lines. Otherwise, no group controls will be in effect at the resumption of simulation.

Group definition must not result in a circular relationship between any two groups.

### EXPLANATION:

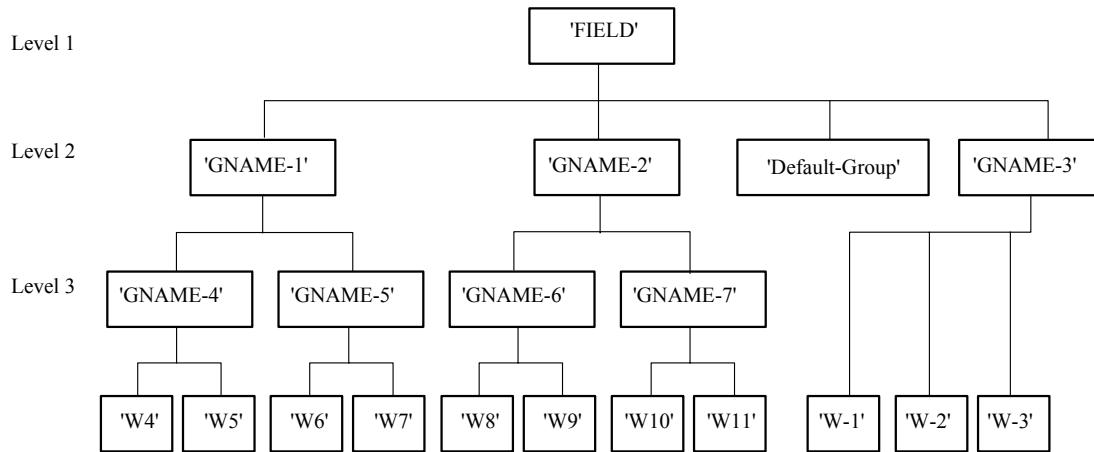
This keyword identifies the group, by name, and assigns it to a parent group. There is no limit to the number of levels of groups allowed in the group hierarchy:

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 not explicitly attached to a parent group. Level 2 groups can have either wells or groups attached to them, but not a mixture of the two. That is, if a level 2 group is named after the \*ATTACHTO subkeyword in a \*WELL line, then that group must not appear after the \*ATTACHTO subkeyword in any \*GROUP line, and vice versa.
3. Level *n*. These groups have level *n*-1 groups as their parents. A level *n* group may consist entirely of attached wells or entirely of attached groups but not a mixture of the wells and groups.

Examples of valid and invalid well control trees are given below.

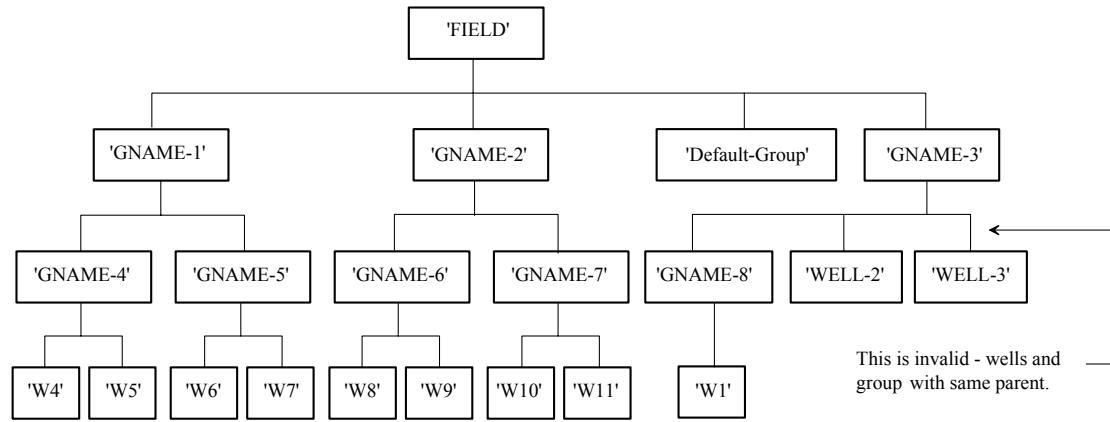
Valid example of a well control hierarchy:



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:



---

## Well Identification (Required)

\*WELL

### PURPOSE:

\*WELL is used to identify wells.

### FORMAT:

\*WELL (*wnum*) *well\_name* (\*VERT *ibl jbl*) (\*FRAC *frac*)  
(\*ATTACHTO *group\_name*)

### DEFINITIONS:

*wnum*

Well number is obsolete. Use *well\_name* instead. Well number is still read by both STARS and Builder for compatibility with existing data files, but Builder does not write well number. This applies to all keywords that require a well identifier.

*well\_name*

Well name in quotes (maximum 40 characters). See **Well Identifiers** and **Wildcarding Well Names** at the beginning of this chapter.

\*VERT *ibl jbl*

This optional subkeyword indicates that the well is vertical and all completion layers have the same I and J grid block indices. *ibl* and *jbl* are the I and J direction grid block index for the vertical well, respectively. There are no default values. If present, \*FRAC must follow immediately after *well\_name*.

\*FRAC *frac*

This optional subkeyword indicates that the rates and indices used internally will be the fraction *frac* of those specified directly in the well data. The allowed range for *frac* is 0 to 1. All rates and indices can be entered for a full well, and they will be multiplied by *frac* for internal use.

\*ATTACHTO *group\_name*

This optional subkeyword attaches this well to the parent group named *group\_name*, where *group\_name* is a quoted string of up to 16 characters. See the manual page for keyword \*GROUP. If \*ATTACHTO is absent then the well is connected to an internally-generated group named 'Default-Group' by default.

### DEFAULTS:

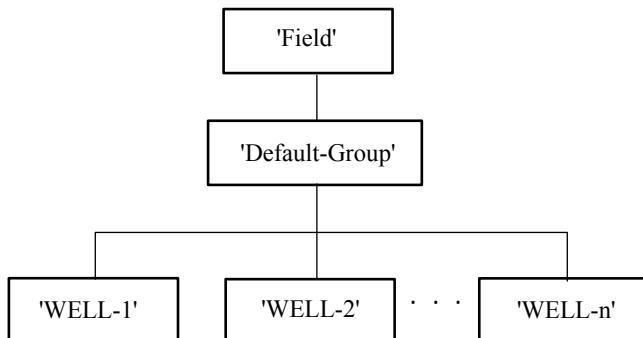
Required keyword. No defaults. Minimum required is:

\*WELL *well\_name*

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.



If \*ATTACHTO is absent, then the well is not attached to a non-default group. The well may be attached to one or more groups via \*GROUP.

#### CONDITIONS:

Well names must be unique, since they are the only way to refer to wells.

#### EXPLANATION:

This keyword identifies the well name, and optionally assigns it to a well group.

\*FRAC is useful for defining fractions of wells when simulating parts of field patterns, e.g., symmetry elements. Rates and well indices for a full well may be entered as data, and \*FRAC will specify what fraction of these values to use. \*FRAC affects the following quantities:

1. Rates specified by \*OPERATE.
2. Well indices entered directly via \*PERF or \*PERFV, that is, without \*GEO. The index used will be  $wifrac$ .
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

```

See the section entitled "Well Fraction" at the beginning of this chapter.

For a well which uses *frac* different from 1, all the reported well performance statistics correspond to the fractional well. Note: In addition to \*FRAC, it may be necessary to account for partial blocks in \*PERF as well. See **Well Completion in a Partial Block** in the \*VAMOD entry.

### Well Groups

\*ATTACHTO allows you to associate the well with a group for reporting purposes. Alternatively, a well can be attached to one or more groups via the \*GROUP keyword. See \*GROUP for a description of well groups.

You may change later in the run the group to which a well is attached by using \*WELL with a different group\_name. This will remove the well from all group lists and then attach the well to group\_name if \*ATTACHTO is present.

### Discretized Wellbores

Discretized wellbores are defined via keyword \*WELLBORE in the Reservoir Description chapter. Each discretized wellbore has one stream for each non-circulating wellbore and two streams for each circulating wellbore (tubing and annulus). Each wellbore stream will be associated with a source/sink well via \*PERF.

Example: There are 2 non-circulating discretized wellbores, 1 circulating discretized wellbore, and 2 normal source/sink wells. Keyword data for this situation might be

```

*WELLBORE rw          ** First horizontal producer
*RANGE . . .

*WELLBORE rw          ** Circulating steam injector
*CIRC WELL ra i j k nwbwt
*RANGE . . .

*WELLBORE rw          ** Second horizontal producer
*RANGE . . .

*WELL 'HorzProd 1'    ** First horizontal producer
*WELL 'Inj Tubing'    ** Circulating steam injector
*WELL 'Inj Vent'      ** Circulating steam injector
*WELL 'HorzProd 2'    ** Second horizontal producer
*WELL 'VertWell 1'     ** Non-discretized well
*WELL 'VertWell 2'     ** Non-discretized well

```

See the EXPLANATION for keyword \*WELLBORE in the Reservoir Description chapter.

## 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 quoted well names assigned via \*WELL. The reporting group will include all wells in the list. See **Wildcarding Well Names** at the beginning of this chapter.

#### 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,  $w(\text{well})$  is the weight value for the particular well, and  $Q(\text{well})$  is the quantity's value for the well. Note that  $w(\text{well})$  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.

A well's contribution to a group is based on its fraction defined via \*WELL \*FRAC. For example, a well with \*FRAC 0.5 and \*OPERATE \*STW 100 will contribute 50 to the group if its group weighting is not changed from 1. To contribute a full-well value to the group, use a group weighting factor equal to the inverse of the well fraction.

---

## Well Head Method (Optional)

\*HEAD-METHOD

### PURPOSE:

\*HEAD-METHOD is used to identify the head method to be used.

### FORMAT:

\*HEAD-METHOD *well\_list*  
    *method*

where *method* is

\*GRAVITY

or

\*GRAV-FRIC (\*HEADRROUGH *rrough*) (\*DUKLER-BANKOFF)

or

\*GRAV-FRIC-HLOS (\*HEADRROUGH *rrough*) (\*DUKLER-BANKOFF)

### DEFINITIONS:

*well\_list*

One or more quoted well names to specify the wells to which this definition applies. See **Wildcarding Well Names** at the beginning of this chapter.

\*GRAVITY

The head between well layers is calculated based on mobility weighted densities.

\*GRAV-FRIC

The head between well layers is calculated using a correlation which accounts for fluid densities, frictional effects, and kinetic energy effects. See **Closed Perforations** below.

\*GRAV-FRIC-HLOS

\*GRAV-FRIC with additional conductive heat transfer calculation between a wellbore and a reservoir. Rock and phase heat conductivities are taken from rock type 1 and must be the individual values, not some average value for water saturated rock. See the explanation about frictional pressure drop and heatloss calculation under \*PHWELLBORE. Heat transfer will occur only for grid cells specified in \*PERF. See **Closed Perforations** below.

\*HEADRROUGH *rrough*

Specify relative roughness (dimensionless) to be used in the frictional head calculation for the listed wells.

\*DUKLER-BANKOFF

Specify another method of friction calculation. See EXPLANATION below.

## **DEFAULTS:**

If \*HEAD-METHOD is absent for a well, that well uses head method \*GRAVITY.

If \*HEADROUGH is absent for a well that uses a frictional head method, that well uses  $rrough = 10^{-4}$ .

If \*DUKLER-BANKOFF is absent for a well that uses a frictional head method, that well uses the method of Xiao et al.

## **CONDITIONS:**

When \*GRAV-FRIC or \*GRAV-FRIC-HLOS is used then the first layer on the \*PERF card must be leading to the surface.

## **EXPLANATION:**

This keyword identifies the well head method to be used for a given list of wells.

Example:

```
*WELL 'Producer1'  
*WELL 'Injector1' *VERT 12 14  
*WELL 'Prod. 54-233a'  
*HEAD-METHOD 'Producer1' 'Prod. 54-233a' *GRAV-FRIC  
*HEAD-METHOD 'Injector1' *GRAVITY
```

Two different methods are used to calculate the friction pressure drop and liquid holdup in the wellbore. The default method calculates friction pressure drop and liquid holdup according to a flow regime existing in the wellbore. These correlations are valid only for co-current flow. This method is based on "A Comprehensive Mechanistic Model for Two-Phase Flow in Pipelines", J.J. Xiao, O. Shoham, J.P. Brill, Proceedings from 65th Annual Technical Conference of SPE, September 23-26, 1990, New Orleans, USA, SPE 20631.

The method invoked with \*DUKLER-BANKOFF uses Bankoff's correlation to evaluate liquid holdup and Dukler's correlation to calculate friction pressure drop. These correlations are valid only for co-current vertical upward or horizontal flow. A more detailed description can be found in "Aspects of Discretized Wellbore Modelling Coupled to Compositional/Thermal Simulation", V. Oballa, D.A. Coombe, W.L. Buchanan, JCPT, April 1997, Volume 36, No. 4, page 45.

## **Closed Perforations**

Each grid cell specified by a well's \*PERF data becomes a node in that well's tree structure. Between each pair of adjacent nodes, a straight-line assumption is used for geometrical aspects of the fluid head and friction calculations. A wellbore may be unperforated along part of its length, either in intermediate non-pay zones, blank sections along a wellbore or in a top section between pay zone and the to-surface location.

When a grid cell is skipped in the \*PERF data, no head-method node is created at that location. The geometrical aspects of that section are still accounted for to a certain extent by the straight line used between the wellbore nodes on either side of it. This approximation is acceptable for fluid head which uses only elevation. It may not be as accurate for friction calculations which depend upon length along the wellbore. For example the flow regime may change in that interval. Also, the wellbore trajectory may be curved, and including unperforated cells with \*CLOSED status may approximate the wellbore trajectory more

closely. On the other hand, skipping unperforated grid cells in \*PERF data likely will introduce no loss of accuracy for simple vertical wells.

When an unperforated cell is specified in \*PERF data and given \*CLOSED status, a node is created at that location and the head-method calculations are done there. In general this is a more accurate method for unperforated cells, especially for a well with horizontal, curved or arbitrary trajectory.

Heat transfer triggered by \*GRAV-FRIC-HLOS will occur only for grid cells with a well node specified via \*PERF. Therefore an unperforated cell must be specified via \*PERF with status \*CLOSED in order for it to experience \*GRAV-FRIC-HLOS heat transfer but no fluid transfer to and from the grid.

The most complete accounting of heatloss for a well is to do both of the following.

1. Specify length of wellbore from point of entry in the grid to toe location via \*PERF, with status \*CLOSED for unperforated cells, plus \*HEAD-METHOD \*GRAV-FRIC-HLOS. This accounts for heat transfer between wellbore and gridded volume, which is usually the pay zone but sometimes includes a region where the well turns from vertical to horizontal.
2. Specify length of wellbore from surface to point of entry in the grid via \*PHWELLBORE. This accounts for heat loss outside the gridded volume.

---

## **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 quoted well names to specify the wells to which this definition is to be applied. See **Wildcarding Well Names** at the beginning of this chapter.

#### **\*STOP-SIM**

If the simulator detects an attempt to perforate a 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 well in an inactive grid block, a warning message is printed and the perforation is given CLOSED status. With CLOSED status, no fluids flow from or to the reservoir in the layer, but the layer is retained in the well and enters the well's head calculation.

### **DEFAULTS:**

The default is \*CLOSED.

### **CONDITIONS:**

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

### **EXPLANATION:**

This keyword specifies the treatment of attempts to perforate a well in an inactive (null or pinched out) grid block. The attempt may either be considered an error, resulting in the termination of the simulation, or may result in the layer's being perforated with a status of \*CLOSED. A layer with closed status is allowed to be the reference layer for the well (the layer in which the well's bottomhole pressure is defined).

Example:

```
*WELL 1 'Producer1'  
*WELL 2 'Injector1'  
*WELL 3 'Prod. 54-233a'  
*NULL-PERF 'Producer1' 'Injector1' *CLOSED  
*NULL-PERF 'Prod. 54-233a' *STOP-SIM
```

---

## Well Backflow Model (Optional)

\*XFLOW-MODEL

### PURPOSE:

\*XFLOW-MODEL is used to identify the method used to model well backflow and crossflow in a specified set of wells.

### FORMAT:

\*XFLOW-MODEL *well\_list* (\*FULLY-MIXED | \*ZERO-FLOW)

### DEFINITIONS:

*well\_list*

One or more quoted well names assigned with the \*WELL keyword. See **Wildcarding Well Names** at the beginning of this chapter. The wells may be producers or injectors.

### \*FULLY-MIXED

Backflow or crossflow in the wellbore is modelled using an assumption of complete mixing in a wellbore of zero volume (Coats et al, SPE 29111, 1995).

### \*ZERO-FLOW

Backflow or crossflow is handled by setting layer flows to zero. The layer is not explicitly closed, and layer flow resumes if pressures changes result in forward flow.

### DEFAULTS:

For each well not referenced by keyword \*XFLOW-MODEL, \*ZERO-FLOW is assumed.

If neither \*FULLY-MIXED nor \*ZERO-FLOW appears after \*XFLOW-MODEL *well\_list*, then \*FULLY-MIXED is assumed for each well in *well\_list*.

\*FULLY-MIXED and \*ZERO-FLOW are mutually exclusive options and cannot be applied simultaneously to the same well.

### CONDITIONS:

The \*FULLY-MIXED option is not available for (1) unweighted injectors and (2) limited entry perforation (LEP) wells.

### EXPLANATION:

This keyword specifies the backflow model to be used for the listed wells.

Example: Wells 1,2, and 4 are to use the fully-mixed crossflow, and well 3 is to use the zero-flow treatment.

```
*WELL 1 'Producer1'  
*WELL 2 'Injector1' *VERT 12 14  
*WELL 3 'Prod. 54-233a'  
*WELL 4 'Pozo 4'  
*XFLOW-MODEL 1:2 4 *FULLY-MIXED  
*XFLOW-MODEL 'Prod. 54-233a' *ZERO-FLOW
```

---

## **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:**

1. To set the global well initialization frequency, i.e. to apply to all wells, enter  
    \*WELLINIT       (\*CHANGE)  
                      (\*Timestep)
2. To set wells in a user-chosen list to a particular well initialization frequency, enter  
    \*WELLINIT well\_list       (\*CHANGE)  
                      (\*Timestep)

### **DEFINITIONS:**

#### **CHANGE**

Indicates that bottom hole pressure of wells with a constraint different from BHP are to be reinitialized only after significant changes in well operating conditions.

#### **Timestep**

Indicates that bottom hole pressure of wells with a constraint different from BHP are to be reinitialized at the beginning of each timestep, after new wellbore head values have been calculated for the coming timestep.

#### **well\_list**

A set of 'well\_names' or well\_numbers; see below. The presence or absence of the well\_list identifies to the simulator whether the global initialization frequency is being set or particular frequencies are being set.

#### **well\_names**

Any number of well names (in quotes) to specify the wells to which this alteration of initialization frequency applies. Limited wildcarding is available for the list of well names; please see the manual page for the \*SHUTIN keyword for an explanation of the wildcard facility.

## **well\_numbers**

Any number of integers, or ranges of integers to specify the well numbers to which this alteration of initialization frequency applies.

### **DEFAULTS:**

Optional keyword. If \*WELLINIT does not appear in the data set, \*WELLINIT \*CHANGE is the default global well initialization frequency in STARS.

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.

### **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 may facilitate convergence of the Newtonian iterations. Sometimes only a few wells in a large field require the \*WELLINIT \*Timestep treatment; in such cases the default treatment may be set to \*CHANGE using the first format above and the problem wells may be flagged for special treatment using the second format above, including a well list.

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', the following sequence is appropriate:

```
*WELLINIT    *Timestep  
*WELLINIT    'WELL1'    *CHANGE
```

in the WELL AND RECURRENT DATA section.

---

## Shut in Wells above Formation (Optional)

\*MODELHUT,

\*EQUILIBRATE

### PURPOSE:

Allow fluid equilibration in shut-in wells via crossflow between layers.

### FORMAT:

\*MODELHUT *well\_list* (\*ON | \*OFF)  
\*EQUILIBRATE *epsmds*

### DEFINITIONS:

*well\_list*

One or more quoted well names assigned with the \*WELL keyword. See **Wildcarding Well Names** at the beginning of this chapter.

*epsmds*

Fluid equilibration criterion *epsmds* ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ) must be a non-negative, real number.

### DEFAULTS:

For each well not referenced by keyword \*MODELHUT, \*OFF is assumed.

If neither \*ON nor \*OFF appears after \*MODELHUT *well\_list* then \*ON is assumed for each well in *well\_list*.

If keyword \*EQUILIBRATE is absent then *epsmds* is assumed to be 0.1.

### CONDITIONS:

This keyword must be located after a well is defined via \*WELL but before the well is shut in via \*SHUTIN or another operation control.

\*MODELHUT cannot be applied to a discretized wellbore; use \*TRANSIENT \*ON to model transient behavior in a discretized wellbore.

The \*MODELHUT option cannot be applied to a well whose type does not support the \*FULLY-MIXED back/cross flow model; these types are (1) unweighted injectors and (2) limited entry perforation (LEP) wells.

### 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 \*MODELSHUT itself does not cause a well to shut in. Instead, it gives a well the permission to go through an equilibration stage when the well is shut in for the reasons listed above. During this equilibration stage the well is operated on a total rate constraint of zero, which allows cross-flow in the wellbore to redistribute fluid between layers in the reservoir (\*FULLY-MIXED cross flow model). This occurs even if a different \*XFLOW-MODEL option was specified for that well. In the text output file, a well operating in an equilibration stage is identified by the well status MSHT (as opposed to SHUT).

As the equilibration stage continues, pressure drops and layer flow rates may gradually decrease. Depending on the reservoir, well completion and operation conditions, such a fluid equilibration process may last months or even much longer in cases with persistent back flow. When each layer's rate falls below the criterion defined by \*EQUILIBRATE the well leaves the equilibration stage to become fully shut, that is, all the well's active layers are closed immediately. A message is issued when the equilibration criterion has been satisfied.

Once a well has switched from the equilibration stage to fully shut, fluid equilibration is no longer checked unless it is activated by another \*MODELSHUT action. When it is fully shut, a \*MODELSHUT well will not be involved in any group target apportionment. The keyword \*AUTODRILL will not put an autodrillable well in an equilibration stage unless such a well has been drilled previously.

\*MODELSHUT is not effective for a single-perforation well, which will equilibrate instantly.

Various actions can cause a shut-in well to open: \*OPEN, effective \*WLISTOPEN, \*ALTER or \*OPERATE with a non-zero rate value. If an open action is encountered while the well is in an equilibration stage, the well is immediately put back on its usual (most restrictive) operating constraint with its assigned or defaulted backflow model. However, the well retains its \*MODELSHUT status, that is, its ability to use an equilibration stage upon the next shut-in action. If you want to remove this ability, use \*MODELSHUT \*OFF explicitly.

Example:

```
** Wells #1 and #2 will experience fluid equilibration
** when they encounter a shut-in action. Well #3 will
** be fully shut in immediately.
*TIME ...
    *MODELSHUT 1:2 *ON
    *EQUILIBRATE 1d-2    ** Larger equilibration criterion
...
*TIME ...
    *SHUTIN 1:3
```

---

## **Well Type Definition (Required)**

**\*OPEN, \*AUTODRILL**

**\*PRODUCER, \*INJECTOR, \*SHUTIN,**

### **PURPOSE:**

Specify a well's type (injector or producer) or its operating state.

### **FORMAT:**

<b>*PRODUCER</b>	<i>well_list</i>
<b>*INJECTOR</b>	(*MOBWEIGHT (*IMPLICIT  *EXPLICIT)   *UNWEIGHT ) <i>well_list</i>
<b>*SHUTIN</b>	<i>well_list</i>
<b>*OPEN</b>	<i>well_list</i>
<b>*AUTODRILL</b>	<i>well_name</i>

### **DEFINITIONS:**

#### *well\_list*

One or more quoted well names assigned with the \*WELL keyword. See **Wildcarding Well Names** at the beginning of this chapter.

#### **\*PRODUCER *well\_list***

The specified wells are production wells.

#### **\*INJECTOR *well\_list***

The specified wells are injection wells.

#### **\*MOBWEIGHT ( \*IMPLICIT | \*EXPLICIT )**

This subkeyword defines a total mobility weighted injector. The total mobility of a grid block which contains the well is used in rate calculations.

For \*IMPLICIT the total mobility is treated implicitly, that is, the most up-to-date value is used. This option may be necessary when there are large changes in fluid total mobility between timesteps. For \*EXPLICIT (the default) the total mobility is updated only at the beginning of the timestep.

#### **\*UNWEIGHT**

This subkeyword defines an unweighted injector. Injected fluid mobility should be part of a well index.

#### **\*SHUTIN *well\_list***

The specified wells are shut in. SHUTIN can be entered for a well as soon as it has been defined with \*WELL. Another way to shut in a well is to specify a zero rate for it via \*OPERATE or \*ALTER.

#### **\*OPEN *well\_list***

The specified wells are reopened after having been fully defined and then shut-in. Another way to open a well is to specify a non-zero rate for it via \*OPERATE or \*ALTER.

### **\*AUTODRILL *well\_name***

The specified well is currently not drilled (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. See the manual entries for \*DRILLQ and \*GAPPOR for more information.

### **DEFAULTS:**

If both \*UNWEIGHT and \*MOBWEIGHT are absent after \*INJECTOR, then \*UNWEIGHT is assumed.

If \*IMPLICIT and \*EXPLICIT are absent after \*MOBWEIGHT, \*EXPLICIT is assumed.

### **CONDITIONS:**

A well is either an injector or a producer type, and may not be switched from one type to another. This means that after a well has appeared in an \*INJECTOR *well\_list* it may not appear in a \*PRODUCER *well\_list* and vice versa.

A well may appear also any number of times in a \*SHUTIN *well\_list*, or in an \*OPEN well list after it has been typed with \*INJECTOR or \*PRODUCER.

A given well may be declared \*SHUTIN only after it has been defined with \*WELL.

The \*MOBWEIGHT option is required for single-well co-injection. See \*INCOMP.

### **EXPLANATION:**

Keywords \*PRODUCER, \*INJECTOR, \*SHUTIN and \*OPEN are used for well type definition. They also define the current well type list used by the following keywords:

*OPERATE	*TINJW	*PINJW	*INCOMP
*MONITOR	*QUAL	*PHWELLBORE	

Each time a keyword from this list appears, the data associated with that keyword is assigned to all the wells in the current well type list.

A well may be defined at one time using \*WELL, have its completions specified at a later time with \*PERF, and at a still later time have its type defined with \*PRODUCER or \*INJECTOR and go into operation at this time.

If a well has been fully defined and then shut in, \*OPEN will open the well at the starting operating condition. The value of that operating condition can be modified using \*ALTER.

\*OPEN, \*ALTER, and \*AUTODRILL cannot be applied until the well's type has been specified and all its perforations and operating constraints have been specified.

### **\*MOBWEIGHT and \*PHWELLBORE**

If an injector uses \*PHWELLBORE it is recommended to use \*MOBWEIGHT. If the specified operating rate is the maximum rate for the specified wellbore dimensions, then \*IMPLICIT must be used. If \*EXPLICIT were to be used, the calculated rate at the beginning of a new timestep may be much higher than the specified rate due to the explicit treatment of total block mobility, in which case the wellbore size can not handle the higher rates and friction pressure drop will be excessive.

## Shut and Reopen a List of Wells (Optional)

\*WLISTSHUT,

\*WLISSTOPEN

### PURPOSE:

\*WLISTSHUT and \*WLISSTOPEN provide a means to shut temporarily a large list of wells and later re-open them without disturbing the pattern of shut and auto-drillable wells.

### FORMAT:

\*WLISTSHUT    *well\_list*  
\*WLISSTOPEN    (*well\_list*)

### DEFINITIONS:

\*WLISTSHUT *well\_list*

Shut in, and if applicable temporarily remove auto-drillable status from, each well in *well\_list*. Remember *well\_list* for possible use with \*WLISSTOPEN. If multiple \*WLISTSHUT keywords are not separated by \*WLISSTOPEN then their well lists are combined. See EXPLANATION, below.

\*WLISSTOPEN (*well\_list*)

Reopen, or restore the auto-drillable status to, each well in *well\_list* if it is present or in the effective well list defined by the immediately previous \*WLISTSHUT. See EXPLANATION, below.

*well\_list*

One or more quoted well names assigned with the \*WELL keyword. See **Wildcarding Well Names** at the beginning of this chapter. Numbers and names cannot be mixed in the same list.

### DEFAULTS:

These are optional keywords, and there are no defaults.

If no well list follows \*WLISSTOPEN then all of the \*WLISTSHUT list will be opened.

### CONDITIONS:

These keywords must be located in, but may appear anywhere in, the WELL AND RECURRENT DATA keyword group.

### EXPLANATION:

\*WLISTSHUT and its paired keyword \*WLISSTOPEN are used to pause the operations of a list of wells within the time period defined by their appearance in recurrent data. Wells that have already been shut-in, or whose operational types not yet defined prior to the entry of the keyword, are automatically excluded from the list. If there is an explicit action that could potentially modify the status for a particular well during this period (such as \*OPEN, \*SHUTIN, \*AUTODRILL, \*PRODUCER, \*INJECTOR, \*CYCLPROD, \*ALTER, \*TARGET, \*ALTERCP), the change is made and that well is removed from the well list.

Wells being reopened or having their auto-drillable status re-acquired by \*WLISOPEN are taken out of the remaining well list.

Keywords \*WLISHUT and \*SHUTIN are different in that \*WLISHUT has a dynamic well list that paired keyword \*WLISOPEN can operate upon to reopen the listed wells without disturbing the pattern of shut and open wells that existed when the \*WLISHUT keyword was applied.

Example:

```
*TIME 0.
*PRODUCER 'W1' 'W2' 'W3'
    *OPERATE *MAX *STO 500.
*INJECTOR 'W4' 'W5'
    *INCOMP *WATER
    *OPERATE *MAX *STW 150.

*SHUTIN 'W1'
*AUTODRILL 'W3'

*TIME 100.
*WLISHUT 'W1' 'W2' 'W3' 'W4' 'W5'
** Since Well W1 has already been shut in, it will be
** automatically excluded from the list.
** Well W3 is temporarily disallowed to be auto-drilled.

*TIME 200.
*ALTER 'W2'
400.    ** Well W2 is explicitly altered and is thus
        ** removed from the *WLISHUT list.

*TIME 300.
*WLISOPEN **Reopening wells on the remaining list
** issued by *WLISHUT, i.e. Well W4 and W5. Meanwhile,
** Well W3 re-acquires the auto-drillable candidacy.
```

## Wellbore Pressure Drop and Heatloss (Optional)

\*PHWELLBORE

### PURPOSE:

\*PHWELLBORE specifies that pressure drop and heat loss will be calculated from the surface to the first perforation specified via \*PERF card.

### FORMAT:

```
*PHWELLBORE *TABLE wdepth table_number
or
*PHWELLBORE *SAMODEL (*REGIME | *DUKLER-BANKOFF)
{ *RTUBIN x           | *RTUBOUT x           | *RINSUL x
| *RHOLE x           | *RCASOUT x           | *RCASIN x
| *EMTUB x           | *EMINS x             | *EMCAS x
| *EMFORM x          | *COND TUB x         | *CONDINS x
| *COND CAS x        | *COND CEM x         | *COND FORM x
| *HCAFORM x         | *GEOGRAD x          | *DEPTH x
| *CASLENGTH x       | *WLENGTH x           | *RELROUGH x
| *SURFACE_TEMP x   | *INSLLENGTH x       | *KICKOFF_DEPTH x
| *NUMBER-OF-DIVISIONS x | *PUMP_DEPTH x       | *PUMP-POWER x
| *PUMP-MAX-PRESSURE-INCREASE x }
```

### DEFINITIONS:

#### \*TABLE

Specifies that tubing pressure drop between surface and downhole is obtained from the table specified via keyword \*PTUBE1 or \*ITUBE1.

**Note:** This option is currently available only for isothermal runs.

#### *wdepth*

Well depth *wdepth* (m | ft | cm) is the difference in depth between the surface and the reference well completion (center of the grid block in which the well's reference layer is perforated).

#### *table\_number*

Positive integer *table\_number* specifies which of the user-supplied tables entered under \*PTUBE1 (for producers) or \*ITUBE1 (for injectors) is to be used for the current well.

#### \*SAMODEL

This keyword specifies that a semi-analytical model will be used to calculate a wellbore pressure drop and a heatloss.

#### \*REGIME

This keyword indicates a method to be used to calculate a wellbore friction. See EXPALNATIONS below.

**\*DUKLER-BANKOFF**

This keyword indicates another method for wellbore friction calculation. See EXPALNATIONS below.

**\*RTUBIN x**

A real number specifying the inner tubing radius (m | ft | cm).

**\*RTUBOUT x**

A real number specifying the outer tubing radius (m | ft | cm). Difference between \*RTUBOUT and \*RTUBIN is the tubing wall thickness.

**\*RINSUL x**

A real number specifying the insulation radius (m | ft | cm). Difference between \*RINSUL and \*RTUBOUT is the insulation thickness.

**\*RCASIN x**

A real number specifying the casing inner radius (m | ft | cm). Difference between \*RCASIN and \*RINSUL is the annular space.

**\*RCASOUT x**

A real number specifying the casing outer radius (m | ft | cm). Difference between \*RCASOUT and \*RCASIN is the casing thickness.

**\*RHOLE x**

A real number specifying hole radius (m | ft | cm). Difference between \*RHOLE and RCASOUT is the cement thickness.

**\*EMTUB x**

A real number specifying emissivity of the tubing (dimensionless). It is used to calculate the radiation heat transfer coefficient in the annular space when insulation is not present.

**\*EMINS x**

A real number specifying emissivity of insulation (dimensionless). It is used to calculate the radiation heat transfer coefficient in the annular space.

**\*EMCAS x**

A real number specifying emissivity of the casing (dimensionless). It is used to calculate the radiation heat transfer coefficient in the annular space.

**\*EMFORM x**

A real number specifying emissivity of formation around the wellbore (dimensionless). It is used to calculate the radiation heat transfer coefficient in the annular space when casing is not present.

- \*CONDTUB x**  
Thermal conductivity of a tubing wall (J/m-day-C | Btu/ft-day-F | J/cm-min-C).
- \*CONDINS x**  
Thermal conductivity of insulation (J/m-day-C | Btu/ft-day-F | J/cm-min-C).
- \*CONDCAS x**  
Thermal conductivity of a casing wall (J/m-day-C | Btu/ft-day-F | J/cm-min-C).
- \*CONDCEM x**  
Thermal conductivity of cement (J/m-day-C | Btu/ft-day-F | J/cm-min-C).
- \*CONDFORM x**  
Thermal conductivity of formation around the wellbore (J/m-day-C | Btu/ft-day-F | J/cm-min-C).
- \*HCAPFORM x**  
Volumetric heat capacity of the formation around the wellbore (J/m<sup>3</sup>-C | Btu/ft<sup>3</sup>-F | J/cm<sup>3</sup>-C).
- \*GEOGRAD x**  
Average geothermal gradient (C/m | F/ft).
- \*DEPTH x**  
Wellbore depth (m | ft | cm). It is the vertical distance from a surface to the center of a reference layer.
- \*CASLENGTH x**  
Actual casing length (m | ft | cm). Wellbore without a casing is assumed to be open hole.
- \*WLENGTH x**  
Actual wellbore length (m | ft | cm). It is a distance from a surface to the center of a reference layer.
- \*RELROUGH x**  
Relative tubing roughness (dimensionless).
- \*SURFACE\_TEMP x**  
Surface formation temperature (C | F).
- \*INSLENGTH x**  
Insulation length (m | ft | cm). This keyword is used when the tubing insulation is shorter than the wellbore length.

**\*KICKOFF\_DEPTH x**

Kickoff depth (m | ft | cm). It is a depth at which the wellbore deviates from vertical. When a horizontal well is drilled from a surface at an angle different from 90 degrees then the kickoff depth is equal to zero. The angle entry will be calculated from a wellbore length and a wellbore depth.

**\*NUMBER-OF-DIVISIONS x**

Dimensionless number specifying number of sections along a wellbore. All wells located at the same position MUST have the same number of divisions.

**\*PUMP\_DEPTH x**

Pump depth (m | ft | cm) is a depth at which a pump is located when fluid can not be lifted naturally. When the same location is used for a producer and an injector e.g. Tubing and Annulus or injector/producer, then the injector must also have this keyword. Pump depth keyword together with other “locations” keywords is used to calculate the wellbore sections. The same “wellbore” must have the same number of sections due to heatloss calculation.

**\*PUMP-POWER x**

Pump power (W | Horse Power | W) value will be used to calculate pressure increase at the pump location. The obtain pressure together with other existing parameters is used to calculate the producer’s friction and heatloss from the pump location upwards. Do not use this keyword for injectors even when the injector has the \*PUMP-DEPTH keyword.

**\*PUMP-MAX-PRESSURE-INCREASE x**

Maximum pressure increase for a pump with a specified power (kPa | psi | kPa).

**DEFUALTS:**

If \*PHWELLBORE is absent then wellbore pressure drop and heatloss is not calculated. If \*PHWELLBORE is present, the following parameter defaults are available:

```
*REGIME  
*RHOLE = *RCASOUT = *RCASIN = *RINSUL = *RTUBOUT  
*EMTUB, *EMCAS = 0.8; *EMINS = 0.9; *EMFORM = 0.94  
*CONDTUB, *CONDTCAS = 3.738e6 J/m-day-C; 576.85 Btu/ft-day-F;  
*CONDINS = 7776 J/m-day-C; 1.2 Btu/ft-day-F;  
*CONDCEM = 3.024e4 J/m-day-C; 4.8 Btu/ft-day-F;  
*CONDFORM = 1.496e5 J/m-day-C; 24 Btu/ft-day-F;  
*HCAPFORM = 2.347e6 J/m3-C; 35 Btu/ft3-F;  
*CASLENGTH = *WLENGTH = *DEPTH  
*GEOGRAD = 0;
```

```
*RELROUGH = 0.0001;  
*SURFACE_TEMP = *TSURF  
*INSLLENGTH = 0 m; 0 ft;  
*KICKOFF_DEPTH = 0 m; 0 ft;  
*NUMBER-OF-DIVISIONS - 30 or 25m sections whichever is less  
*PUMP-DEPTH = 0 m; 0 ft;  
*PUMP-MAX-PRESSURE-INCREASE = 1.e4 kPa
```

### CONDITIONS:

\*PHWELLBORE is applied to the current well list defined by the last \*PRODUCER or \*INJECTOR keywords, and so must immediately follow them. \*PHWELLBORE is required if one of the operating constraints is \*WHP or when the wellbore model is desired. Water, oil and gas heat conductivities are also needed in the calculation. These values must be the actual heat conductivities because they are used to calculate dimensionless quantities such as Prandtl number. Do not use the average values. Properties from ROCKTYPE 1 are used.

A warning message is issued if a SAM \*DEPTH does not match the depth of the block containing the attached well. A common cause is that depth is not specified for the reservoir.

The first perforation on the \*PERF card must be the one that connects directly to this model.

The \*PHWELLBORE data must be repeated every time \*INJECTOR or \*PRODUCER is specified at later times. When this data is not repeated then friction pressure drop and heatloss are no longer calculated. STARS issues a message when this happens. When \*ALTER is used then the \*PHWELLBORE data is retained and does not need to be repeated.

### EXPLANATION:

Pressure drop along the wellbore and radial heatloss from a wellbore is calculated semi-analytically. The basic idea is from a paper: J. P. Fontanilla, K. Aziz, 'Prediction of bottom-hole conditions for wet steam injection wells.', JCPT, March-April 1982, p.82-88. This idea was generalized to both types of wells – injectors, producers, injection or production of any fluid and the use of any STARS operating conditions. The Semi-analytical Model (SAM) may be used with Sink/Source wells and also Discretized wells. A well can enter the formation under any angle. The angle is calculated from a depth and wellbore length data. When a horizontal well is drilled and the angle changes from 90 degrees to another value then the keyword \*KICKOFF\_DEPTH should be used to indicate the depth at which the angle changes.

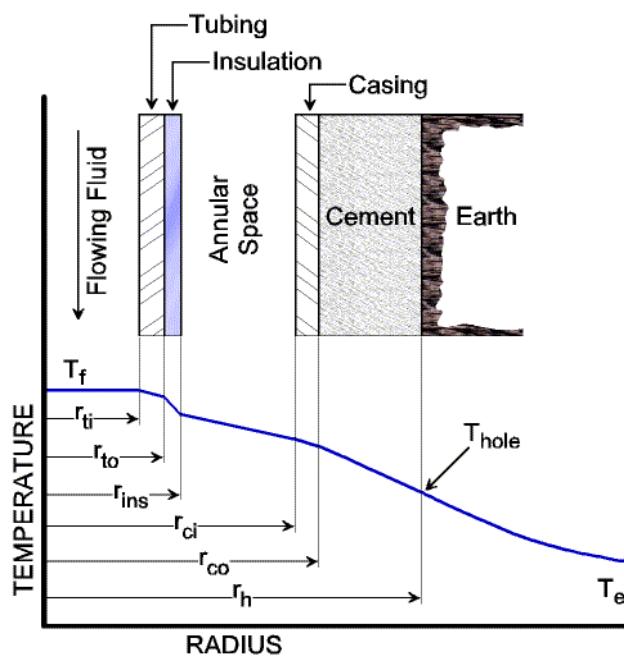
Momentum and energy equations must be solved simultaneously to evaluate pressure and enthalpy changes along the wellbore. Pressure drop depends on friction, gravity and kinetic energy. Two methods are used to calculate friction pressure drop and gravity. The default method which is invoked with \*REGIME calculates friction from correlations which depend on a flow regime and type of a fluid (two phase or homogeneous). These correlations are valid only for co-current flow. This method is based on "A Comprehensive Mechanistic Model for Two-Phase Flow in Pipelines", J.J. Xiao, O. Shoham, J.P. Brill, Proceedings from 65th Annual Technical Conference of SPE, September 23-26, 1990, New Orleans, USA, SPE 20631.

The second method which is invoked with \*DUKLER-BANKOFF uses Bankoff's correlation to calculate liquid holdup and friction pressure drop according to Dukler. These correlations are valid only for co-current vertical upward or horizontal flow. A more detailed description can be found in "Aspects of Discretized Wellbore Modelling Coupled to Compositional/Thermal Simulation", V. Oballa, D.A. Coombe, W.L. Buchanan, JCPT, April 1997, Volume 36, No. 4, page 45.

Kinetic energy is calculated from velocity and fluid properties. Changes in thermal properties along the wellbore are accounted for by evaluating the enthalpy at various depths. Enthalpy, instead of temperature must be used because of steam injection. Enthalpy changes depend on gravity, kinetic energy, radial wellbore heatloss and the amount of fluid flowing in the wellbore. Radial wellbore heatloss is a product of an overall heat transfer coefficient and a difference between fluid and formation temperature. The overall heat transfer coefficient is calculated from the input data. It depends on resistivity in the fluid film, tubing wall, insulation, annular space, casing wall and cement. Radial heatloss will decrease with time because the formation temperature will increase. It is necessary to know the fluid rate to evaluate the total enthalpy. It is sometimes difficult to calculate the correct rate after a well change when a well operates on a constraint different from a rate constraint. The problem is more complex for Discretized wellbore with dual stream (tubing, annulus) and pseudo-steady state initialization

(\*TRANSIENT OFF). In this case the tubing and annulus must be initialized to such conditions that are very close to pseudo-steady state without prior knowledge of a wellbore pressure or fluid composition at the bottom hole conditions. The estimated rates may be too high or too low just with a small error in wellbore pressure estimation (difference less than 0.1 kPa) due to high wellbore permeability (well index). When the estimated rate is too high then high friction is calculated and unphysical pressure values are encountered. When the rates are very low then the available heat is not enough to service the heatlosses and a spontaneous condensation occurs in the wellbore. Because of this complexity only certain combinations of starting operating conditions are allowed for tubing and annulus. Currently, wellhead pressure (WHP) may not be used as starting operating condition for a producer. However, it may be used as a second or subsequent operating constraint. WHP may be used as a starting operating constraint for an injector only when the other stream (tubing or annulus) operates at constant rate. WHP may be used with other combinations as a subsequent operating constraint. STARS checks all combinations and will stop with an error when it can not initialize a dual stream wellbore.

SAM model is fully coupled with the simulator except the formation temperature. Temperature gradient between Thole and Te (see Figure below) is calculated in the wellbore model and has an effect on the radial heatloss. However, the heatloss/gain will not affect a grid block temperature when specified wellbore depth is in the pay (for example the heel of a horizontal well). If the grid block temperature change is important then the SAM well should end at the top of the pay when Discretized wells or Sink/Source wells with heatloss calculation are used. It does not matter for regular Sink/Source wells because conductive heat transfer is not accounted for anyway.



#### *Wellbore schematic*

A simple option for pumping a producer is also installed in STARS. The pump depth and pump output power is a user specified value. The pump output power is the product of the power at the pump inlet and its efficiency. The total dynamic head (pressure increase) is calculated as a ratio of pump power and existing volumetric flow rate. The calculated pressure increase is added to the existing wellbore pressure value at the pump location and wellbore friction and heatlosses are evaluated at each wellbore section towards the well-head location. When the pump option is not specified and wellbore pressure decreases to atmospheric before it reaches the well-head location STARS issues a message that it can not lift the producing fluid. The option assumes that the pump power is high enough to satisfy the minimum required Net Positive Suction Head.

Example:

```

*WELL 2 'INJECTOR'
*INJECTOR 2
** For injector use SAM to calculate bottom-hole pressure and quality. It
is a horizontal well that enters the formation under and angle different
from 90 degrees. A sink/Source well model is used.
*PHWELBORE *SAMODEL
RTUBIN 0.15
DEPTH 460.0
WLENGTH 600.0
:
*OPERATE .....
:
PERF 2
1:19 1 24 ** horizontal section

```

## **Printing and Plotting Wellbore Results**

Use keyword \*SAMINFO in the Recurrent Data section to enable printout of wellbore heatloss results to the “.out” file.

Use special history \*OUTSRF \*SPECIAL \*PHWELL to plot specific wellbore heatloss results versus time.

## **Converting Data from the \*HEATLOSS Model**

The obsolete wellbore heatloss model that was enabled via keyword \*HEATLOSS in the recurrent data can be converted to \*PHWELBORE \*SAMODEL. Most individual data items are converted merely by changing the keyword. The following steps will help in the conversion. Note that, even for a simple case that uses no features not found in the \*HEATLOSS model, the result may be different due to differences in details in the calculations (e.g., water properties).

1. Identify the first \*HEATLOSS (\*ON) keyword in recurrent data, and replace it with \*PHWELBORE \*SAMODEL. This new keyword works on the same well list as \*HEATLOSS did.
2. Move the wellbore heatloss definition data from the “Other Reservoir Properties” chapter to a point immediately after \*PHWELBORE \*SAMODEL.
3. Change the keywords according to the following table.

<u>*HEATLOSS</u>	<u>*PHWELBORE *SAMODEL</u>
*RTI rti	*RTUBIN rti
*RTO rto	*RTUBOUT rto
*RIN rin	*RINSUL rin
*RCI rci	*RCASIN rci
*RCO rco	*RCASOUT rco
*RH rh	*RHOLE rh
*ETO eto	*EMTUB eto
*ECI eci	*EMCAS eci
*EIN ein	*EMINS ein
*EE ee	*EMFORM ee
*XKE xke	*CONDFORM xke
*XKIN xkin	*CONDINS xkin
*XKCM xkcm	*CONDCEM xkcm
*XAE xae	*HCAPFORM xke/xae
*DEPTH depth	*WLENGTH wlength
*CDEPTH cdepth	*CASLENGTH cdepth
*ANG ang	*DEPTH wlength*sin(ang)
*AGD agd	*GEOGRAD agd
*WBHLOUT	*SAMINFO

---

## Injection Stream Attributes

\*TINJW, \*QUAL, \*PINJW

### PURPOSE:

Assign and override defaults of attributes of the injected stream.

### FORMAT:

\*TINJW *tinjw*  
\*QUAL *qual*  
\*PINJW *pinjw*

### DEFINITIONS:

*tinjw*

Temperature of the fluid injected into the well (C | F). It must lie within the allowed temperature range (see \*MINTEMP and \*MAXTEMP).

If the water phase injection rate includes steam expressed in its equivalent of cold water, then *tinjw* is the temperature of the water-steam mixture coming out of the boiler.

If the wellbore heatloss model is used, then *tinjw* is the temperature of water entering the wellbore. Otherwise, it is the temperature of water entering the reservoir.

*qual*

Steam quality, expressed as mass or mole ratio of vapour to vapour plus liquid, of water (component #1). The allowed values are 0 to 1. Injection rate of the steam-water mixture is expressed as the rate of equivalent condensed water (component #1) at the steam boiler intake (cold water equivalent). This equivalent rate is entered in the water phase, and the water phase injectivity index will be used.

*pinjw*

Pressure of the injected single-phase fluid (kPa | psi). It is used in enthalpy and density calculation, even when \*QUAL is not used. It must lie within the allowed pressure range (see \*MNPRES and \*MAXPRES). It is recommended that \*PINJW be defined when \*QUAL 1 is used.

### DEFAULTS:

If \*QUAL is absent then *qual* = 0 is assumed. When *qual* = 0 (via default or keyword) the enthalpy of injected water is based on liquid phase alone and is a function of \*TINJW and possibly \*PINJW.

If \*PINJW is absent then  $p_{injw}$  depends on the value of  $qual$ :

1. If \*QUAL is present and  $qual$  lies between 0 and 1, default  $p_{injw}$  is the steam saturation pressure at temperature  $t_{injw}$ .
2. If \*QUAL is absent or  $qual$  is equal to 0 or 1, default  $p_{injw}$  is equal to the pressure of the grid block containing the well. It is recommended that \*PINJW be defined when \*QUAL 1 is used.

### CONDITIONS:

\*TINJW is required for thermal runs.

\*TINJW must appear after \*INJECTOR and before \*OPERATE.

\*PINJW is used when  $qual$  is 0 or 1, or when \*QUAL is absent.

### EXPLANATION:

To specify injection of saturated water (liquid and vapour) at  $t_{injw}$ , enter \*TINJW  $t_{injw}$  and mandatory \*QUAL  $qual$  where  $0 < qual < 1$ . At saturated conditions,  $p_{injw}$  is the saturation pressure of water at  $t_{injw}$ , and  $t_{injw}$  is less than critical temperature.

To specify injection of undersaturated water (liquid) at  $t_{injw}$  and  $p_{injw}$ , enter \*TINJW  $t_{injw}$ , \*PINJW  $p_{injw}$  and optional \*QUAL 0. At undersaturated conditions,  $p_{injw}$  is greater than the saturation pressure of water at  $t_{injw}$ , and  $t_{injw}$  is less than critical temperature.

To specify injection of superheated or supercritical water (vapour) at  $t_{injw}$  and  $p_{injw}$ , enter \*TINJW  $t_{injw}$ , \*PINJW  $p_{injw}$  and mandatory \*QUAL 1. At superheated conditions,  $p_{injw}$  is less than the saturation pressure of water at  $t_{injw}$ , and  $t_{injw}$  is less than critical temperature.

At supercritical conditions,  $t_{injw}$  is greater than critical temperature.

Example: Inject 100 m<sup>3</sup>/d CWE of 70% quality steam with 2 m<sup>3</sup>/d of aqueous tracer. If  $m_L$  and  $m_G$  are the masses of liquid and vapour water component, respectively, then steam quality  $m_G/(m_G+m_L)$  is independent of the tracer. Water and tracer surface condition densities are  $5.5 \cdot 10^4$  gmole/m<sup>3</sup> and  $5 \cdot 10^3$  gmole/m<sup>3</sup>, respectively.

```
*INJECTOR *MOBWEIGHT 'INJECTOR'  
*TINJW 400  
*QUAL 0.70  
*INCOMP *WATER 0.998185 0.001815 0 ** water, tracer  
*OPERATE *STW 102 ** 100 water + 2 tracer
```

## Composition of Injected Phases

\*INCOMP

### PURPOSE:

Specifies which phases are to be injected, along with their compositions.

### FORMAT:

*INCOMP *WATER	w(1) ... w(numx)
*INCOMP *OIL	x(1) ... x(numx)
*INCOMP *GAS	y(1) ... y(numy)
*INCOMP *WATER-GAS	v(1)....v(numy)
*INCOMP *WATER-OIL	v(1)....v(numx)
*INCOMP *WATER-GAS-OIL	v(1)....v(numy)
*INCOMP *CYCLING	

### DEFINITIONS:

\*WATER w(1) ... w(numx)

Mole fractions of injected water phase. The allowed range for each is 0 to 1. They should sum to one, but will be normalized if not. See keyword \*MODEL.

\*OIL x(1) ... x(numx)

Mole fractions of injected oil phase. The allowed range for each is 0 to 1. They should sum to one, but will be normalized if not. See keyword \*MODEL.

\*GAS y(1) ... y(numy)

Mole fractions of injected gas phase. The allowed range for each is 0 to 1. They should sum to one, but will be normalized if not. See keyword \*MODEL.

\*WATER-GAS v(1) ....v(numy)

Inject water or steam together with a component that exists in the gas phase at surface conditions, for example, gaseous solvent with steam.

\*WATER-OIL v(1) ....v(numx)

Inject water or steam together with a component that exists in the oil phase at surface conditions, for example, liquid solvent with steam.

\*WATER-GAS-OIL v(1) ....v(numy)

Inject water or steam together with multiple components, each of which exists in either the oil or gas phase at surface conditions. This is a generalization of \*WATER-GAS and \*WATER-OIL and can mimic them with suitable surface K values.

\*CYCLING

This subkeyword specifies the water cycling injector whose phase composition is to be obtained from the water phase produced by assigned producers.

$v(i)$

**Volume** fraction of component injected at surface conditions. The allowed range is 0 to 1. The  $v(i)$  should sum to one, but will be normalized if not. The value for water  $v(1)$  must be non-zero. See **Multi-phase Co-injection**, below.

## DEFUALTS:

There are no defaults.

The composition of a water cycling injector is initialized to the lumped water phase composition produced by the producers under the same group, or the entire field if no group hierarchy exists. This composition will then be reset accordingly if the group water recycling option is in effect (see \*GCONI \*RECYCLE).

## CONDITIONS:

\*INCOMP is a required keyword for injectors, located before its corresponding list of \*OPERATE keywords.

Single-well co-injection requires \*INJECTOR \*MOBWEIGHT and non-zero  $v(1)$ .

## EXPLANATION:

See **Specifying Injection Phase** at the beginning of the Recurrent and Well Data chapter.

### Multi-phase Co-injection

Gas and/or oil phases can be co-injected with water or steam in the same well at the same time using \*INCOMP subkeywords \*WATER-GAS, \*WATER-OIL and \*WATER-GAS-OIL. An example is steam injected with gaseous or liquid solvent. A total phase rate constraint must be specified with \*OPERATE \*STF, but pressure-type constraints can be used as well. The \*MOBWEIGHT option of \*INJECTOR must be used, and  $v(1)$  must be non-zero.

For the purpose of specifying the composition of the injection stream, each component volume is referenced to a single phase at surface conditions with an associated density. Fraction  $v(i)$  is the volume of each injected component divided by the total injected volume. Water component (number 1) is referenced to the water phase, and non-condensable gas components (number numx+1 to numy) are referenced to the gas phase. Oleic components (number numw+1 to numx) are referenced to the gas phase for \*WATER-GAS and to the oil phase for \*WATER-OIL. For \*WATER-GAS-OIL an oleic component is referenced to the gas phase if its surface-condition gas-liquid K value is greater than one, and is referenced to the oil phase otherwise. In any case, a component must exist in its reference phase at surface conditions as specified by \*MODEL, K value data and possibly \*SURFLASH.

For consistency of the  $v(i)$ , the injected volumes of all the components must have the same unit. Specifically, in field units the volume of a component referenced to the gas phase must be calculated in units of bbl/day.

In order to relate the injection surface and downhole rates, water uses the steam quality option but the remaining injected components use a flash calculation to determine the downhole gas-oil phase split and ratio between surface and downhole densities. This gas-oil split is calculated only once per timestep (explicit), but the density ratio is updated continuously (implicit) to account for pressure changes in the wellbore.

Rates and accumulations of co-injected components are reported in their respective surface reference phase. The columns of the diary (log) report are adjusted to accommodate oil phase injection, since by default the injection columns are only for water and gas phases.

**Example**

```
** Co-injection of water with air
** Component    PhaseRate   Fraction   CompRate   VolFrac
** 'WATER'      88.3       w         1.00       88.3      0.002877318
** 'INRT GAS'   30600      g         0.79       24174.0   0.787726919
** 'OXYGEN'     30600      g         0.21       06426.0   0.209395763
** Total          30688.3
*INJECTOR *MOBWEIGHT 'INJECTOR'
*INCOMP *WATER-GAS 0.002877318 0 0 0 0.787726919 0.209395763
*OPERATE *STF 30688.3 ** water rate + air rate in m3/day
```

## Well Operating Constraints (Required)

\*OPERATE

### PURPOSE:

\*OPERATE defines a well operating constraints, and the remedial action to be taken when it is violated.

### FORMAT:

\*OPERATE ( \*MAX | \*MIN ) *type value ( action )*

or

\*OPERATE ( \*MIN ) \*STEAMTRAP *value ( uba ) ( action ) (\*FLUID-TEMP)*

where *type* is one of:

- \*STO, \*BHO,
- \*STG, \*BHG,
- \*STW, \*BHW,
- \*STL, \*BHL,
- \*STF, \*BHF,
- \*WHP (\*IMPLICIT), \*BHP,
- \*DWN, \*DWA, \*DWB,
- \*STEAM
- \*OXYGEN
- \*STO\_COMP *comp\_name*
- \*STG\_COMP *comp\_name*

and *action* is one of:

- \*CONT (\*REPEAT)
- \*SHUTIN (\*REPEAT)
- \*STOP
- \*NEXTSEG
- \*ONETIME

### DEFINITIONS:

#### \*MAX, \*MIN

Specifies a maximum or minimum constraint type. If neither are present, the constraint uses the default which is \*MAX unless explicitly noted below.

#### \*STO, \*BHO

Constraint type is oil phase rate, and *value* has units (m<sup>3</sup>/day | bbl/day | cm<sup>3</sup>/min). Reference condition is surface for \*STO, reservoir for \*BHO. A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed.

#### \*STG, \*BHG

Constraint type is gas phase rate, and *value* has units (m<sup>3</sup>/day | ft<sup>3</sup>/day | cm<sup>3</sup>/min). Reference condition is surface for \*STG, reservoir for \*BHG. A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed.

**\*STW, \*BHW**

Constraint type is water phase rate, and *value* has units ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ). Reference condition is surface for \*STW, reservoir for \*BHW. A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed.

**\*STL, \*BHL**

Constraint type is total liquid (water phase plus oil phase) rate, and *value* has units ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ). Reference condition is surface for \*STL, reservoir for \*BHL (producer only). A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed.

**\*STF, \*BHF**

Constraint type is total fluid (water phase plus oil phase plus gas phase) rate, and *value* has units ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ). Reference condition is surface for \*STF (injectors only), reservoir for \*BHF (producer only). A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed. Use \*STF for co-injection of steam and gas or steam and solvent.

**\*WHP (\*IMPLICIT), \*BHP**

Constraint type is pressure, and *value* has units (kPa | psi | kPa). The location is bottomhole for \*BHP and wellhead for \*WHP. \*WHP requires that pressure-drop data be entered via keywords \*PHWELLBORE.

Each producer should have a minimum pressure type constraint \*BHP or \*WHP; the recommended value is 100 kPa, but whatever is used must lie within the allowed pressure range (see \*MINPRES and \*MAXPRES).

If neither \*MIN nor \*MAX is present, then \*MAX is assumed for an injector and \*MIN is assumed for a producer.

Maximum WHP or BHP constraints are not allowed for producers under \*OPERATE, but may be imposed using \*MONITOR. Similarly, minimum WHP or BHP constraints for injectors are not allowed under \*OPERATE, but may be imposed using \*MONITOR.

Wellhead pressure option \*WHP has only one implicitness setting, corresponding to subkeyword \*IMPLICIT in the other CMG simulators.

**\*DWN, \*DWA, \*DWB**

Constraint type is drawdown pressure (difference between the wellbore pressure and the grid block pressure), and *value* has units (kPa | psi | kPa). A zero value will shut in an open well, and a non-zero value will open a shut-in well. Only \*MAX is allowed for both producers and injectors. There are three different drawdown options to choose from. In the following, sign  $\pm$  indicates (+) for a producer and (-) for an injector.

\*DWN specifies drawdown in the well's **reference** layer

$$\Delta P_d = \pm (P_{block_{ref}} - P_{well_{ref}})$$

When the reference layer is perforated in a null or pinched-out block, the simulation will either (1) be terminated if \*DWN is the primary operating constraint, or (2) continue with \*DWN constraint taking no effect if it is not the primary one.

\*DWA specifies the **maximum** drawdown within all open (sink-source) layers. This constraint type is useful when avoiding formation damage.

$$\Delta P_d = \max_{l, open} \pm (P_{block_l} - P_{well_l})$$

\*DWB specifies the **average** drawdown for all open (sink-source) layers, weighted by the total productivity or injectivity (PI) at reservoir conditions.

$$\Delta P_d = \pm \sum_{l, open} PI_l (P_{block_l} - P_{well_l}) / \sum_{l, open} PI_l$$

This has 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.

#### \*STEAM

Constraint type is steam rate expressed in cold water equivalent (CWE) and *value* has units (m<sup>3</sup>/day | bbl/day | cm<sup>3</sup>/min). Steam is all aqueous components produced in the vapour phase at downhole conditions. A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. This constraint type may not be used with an injector and only \*MAX is allowed.

Use special history \*OUTSRF \*SPECIAL PRODSTEAMR to plot the actual value of steam rate CWE. When a well is operating under the \*STEAM constraint, its actual steam rate PRODSTEAMR will be close to the constraint *value* times the well fraction.

#### \*OXYGEN

Obsolete. Use \*STG\_COMP instead.

#### \*STEAMTRAP *value* ( *uba* ) ... (\*FLUID-TEMP)

Constraint type is production in steam trapping mode (see EXPLANATION, below). The *value* is by how much the steam saturation temperature (corresponding to well bottomhole pressure) exceeds the temperature of the produced water. The unit is temperature difference (C deg | F deg | C deg).

The associated \*PRODUCER well list must contain a single well. This constraint type may be applied to multiple wells simultaneously, but there must be a separate set of \*PRODUCER and \*OPERATE keywords for each well. \*MAX may not be used with this constraint type. This constraint type may not be used with an injector or the ZT formulation (\*TFORM \*ZT).

User block address *uba* is the optional steam trap location and is needed only when it is different from the default location. For a source/sink well *uba* must appear in the well's \*PERF or \*PERFV definition. *uba* may not be specified for a discretized wellbore. See section **User Block Address** in Keyword Data Entry System chapter.

Optional fluid temperature \*FLUID-TEMP may be specified only for a source/sink well when \*HEADMETHOD \*GRAV-FRIC-HLOS is used. In this case the wellbore fluid temperature instead of a grid block temperature is used in the steam trap calculation.

Use special history \*OUTSRF \*SPECIAL PRODSTEAMR to plot the actual value of steam rate CWE.

**\*STO\_COMP** *comp\_name*

Constraint type is rate of the specified component in oil phase at surface conditions, and *value* has units (m<sup>3</sup>/day | bbl/day | cm<sup>3</sup>/min). This constraint is available for producers only. A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed.

**\*STG\_COMP** *comp\_name*

Constraint type is rate of the specified component in gas phase at surface conditions, and *value* has units (m<sup>3</sup>/day | ft<sup>3</sup>/day | cm<sup>3</sup>/min). This constraint is available for producers only. A zero rate will shut in an open well, and a non-zero rate will open a shut-in well. Only \*MAX is allowed.

*comp\_name*

Component name in quotes. The component must be found in the phase indicated by the keyword, as determined by \*MODEL and the K value data entered.

**\*CONT**

The action on violation is to switch to operating on this constraint, and continue the simulation. When \*REPEAT is used then a timestep will reconverge with the new operating constraint.

**\*SHUTIN**

The action on violation is to shut in the well.

**\*REPEAT**

The timestep will be reconverged (repeated) after the action is taken. Every well with this option is entitled to repeat once within one timestep. However, the maximum number of total repeats for this timestep is limited by user input (see keyword \*MXCNRPT).

**\*STOP**

The action on violation is to stop the simulation.

**\*NEXTSEG**

The action on violation is to immediately read and apply the well changes for the next \*TIME or \*DATE.

**\*ONETIME**

This subkeyword specifies that the constraint will not be checked for violation when another constraint is operating. For this option to be effective, this must be the first operating constraint chosen; this can be forced by defining this constraint first in the constraint list and using \*MRC-RESET ... \*NO-RESET.

This option makes it possible to reproduce the behavior of versions 96 and earlier, in which operation always started on the first constraint and it was never checked for violation once control left it.

**DEFUALTS:**

At least one operating constraint must appear for each active well in the data set. When a well does not have BHP constraint then the simulator will assign \*OPERATE \*MIN 101.3 kPa or 14.7 psia for producers and \*OPERATE \*MAX 1.0e+6 kPa or 147,000 psia for injectors.

Except for \*BHP, \*WHP and \*STEAMTRAP noted above, if \*MAX and \*MIN are absent, then \*MAX is assumed.

If *uba* is absent after \*STEAMTRAP then the steam trap location is: a) Sink/Source wells - block/layer where steam first appears; b) Discretized Wellbore - block where Sink/Source end is attached. For a horizontal-only section it would be the heel; when both vertical and horizontal leg is simulated then it would be the surface.

If \*FLUID-TEMP is absent in \*STEAMTRAP then a grid block temperature is used in calculation.

The default remedial action for a constraint violation is \*CONT \*REPEAT.

If \*ONETIME is absent, then the first constraint is checked for violation just like any other.

**CONDITIONS:**

Each occurrence of \*PRODUCER or \*INJECTOR must be followed by at least one \*OPERATE.

When automatic constraint choosing is defeated, the first \*OPERATE after \*PRODUCER or \*INJECTOR is the starting operating constraint for the recurrent time segment defined by the two bracketing \*TIME or \*DATE keywords. Other operating constraints are added by consecutive occurrences of \*OPERATE.

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 *REPEAT
*CONT
least serious

Each constraint type (e.g., \*STO, \*BHP) may appear only once in a constraint list. For example, both \*MIN \*BHP and \*MAX \*BHP constraints may not appear in the same list. The same rule applies to constraint types that differ only in their reference condition, e.g., \*STO and \*BHO.

In general, any combination of \*MIN and \*MAX for injector or producer and operating constraint type is allowed. Exceptions are explicitly noted above.

All constraint types may be used for production. The constraint types that may not be used for injection are \*BHL, \*STEAM, \*OXYGEN and \*STEAMTRAP.

The keyword \*ALTER can be used to change the value of the first operating constraint. However, all the \*OPERATE and \*MONITOR must be repeated if any of the other constraint values are to be changed from a previous definition.

See "Specifying Injection Phase" at the beginning of the Recurrent and Well Data chapter.

#### **EXPLANATION:**

\*OPERATE defines a well's operating constraints and the remedial action to take when a constraint is violated. (Use \*MONITOR for constraints which cannot be used for operating the well, such as WOR). See Appendix F.4 for a discussion of well equation treatment.

A starting operating constraint may be chosen automatically from the list of \*OPERATE (see \*MRC-RESET). When automatic constraint choosing is defeated, then the simulator will initially attempt to operate the well on the first operating constraint. It will check the other constraints for violation only when the timestep converges. If the first \*OPERATE is not realistic, the simulation may stop due to numerical problems.

The list of operating constraints is checked for violation at the end of each timestep. If one of the other signified operating constraints is violated and \*CONT or \*CONT \*REPEAT has been used, then this constraint becomes the operating constraint.

If more than one constraint, regardless of type, is violated at the same time, then the most drastic assigned action is taken, in this order of priority: \*STOP, \*SHUTIN, \*CONT \*REPEAT and \*CONT.

The value of the first operating constraint is changed easily at a subsequent simulation time for a well list with the \*ALTER keyword.

A value for each constraint, regardless of the type of constraint, is required.

For example, to start producing a well at a rate of 500 m<sup>3</sup>/D, with a minimum bottomhole pressure at 2500 kPa, and with monitoring constraints of GOR at 2000 m<sup>3</sup>/m<sup>3</sup> and water cut at 98%, the input is:

```
*OPERATE *MAX *STO 500  
*OPERATE *MIN *BHP 2500 *SHUTIN  
*MONITOR *MAX *GOR 2000 *STOP  
*MONITOR *MAX *WOR 0.98 *STOP
```

While the primary constraint is active and the bottomhole pressure falls below 2500 kPa, then the bottomhole pressure constraint becomes the operating constraint and the well is shut in.

### **Steam Trap**

The steam trap constraint is used to prevent the production of live steam. It does this by keeping the well's flowing bottomhole pressure (and hence the pressure in the grid block containing the well) high enough that live steam does not appear in the well block.

Effectively, with \*STEAMTRAP the well is constrained to a specified BHP as it is with the \*BHP option, but the BHP is a function of well block temperature instead of being constant.

The well equation that is solved is

$$T_{sat}(BHP) - T(block) = \text{value}$$

where  $T_{sat}(BHP)$  is the steam saturation temperature corresponding to pressure BHP,  $T(block)$  is the temperature in the well block, and val is the desired difference between these temperatures. Another way to think of this constraint is

$$BHP = Psat(T(block) + \text{value})$$

When \*FLUID-TEMP is specified the location at which the constraint is applied may be user specified. It is important for:

- a) Slanted Sink/Source wells in heterogeneous reservoirs

- When the location is specified then a complete user block address must be used. For example: a Sink/Source well is perforated in a block 10 1 5 that is refined into 3 1 1. The actual well location is in the refined block 1 1 1. The user block address will be 10 1 5 / 1 1 1.

When \*FLUID-TEMP is specified then the wellbore fluid temperature is used in the above equations instead of  $T(block)$ . Numerically these calculations are more difficult especially for backflowing wells because of wellbore fluid temperature discontinuity. As a consequence the CPU time may increase.

---

## Maximum Number of Continue-Repeats (Optional)

\*MXCNRPT

### PURPOSE:

Specify maximum number of continue-repeats allowed in each timestep.

### FORMAT:

\*MXCNRPT *nmxrpt*

### DEFINITIONS:

*nmxrpt*

A non-negative integer defining the maximum allowed number of timestep repeats (reconvergences) due to well constraint switches.

### DEFAULTS:

If \*MXCNRPT is absent then *nmxrpt* = 1 is assumed, that is, at most one timestep repeat (reconvergence) due to well constraint switch is allowed per timestep, no matter which well(s) experience the constraint switch.

If \*OPERATE ... \*CONT \*REPEAT is absent for all wells, this keyword is ignored.

### CONDITIONS:

This keyword effects only wells that specify \*OPERATE ... \*CONT \*REPEAT.

### EXPLANATION:

Every well having \*CONT \*REPEAT (continue-repeat) specified through \*OPERATE is entitled to one repeat (reconvergence) of the current timestep if one of its operating constraints is violated. During convergence of a timestep, several wells may experience their single constraint switch at different Newton iterations and hence cause multiple convergence repeats in that timestep. Although the possible number of such repeats per timestep cannot exceed the number of wells, it may lead to a significant increase in 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.

## Well Monitoring Constraints (Optional)

\*MONITOR

### PURPOSE:

\*MONITOR defines the monitored well constraints, and the remedial action to be taken when a monitored constraint is violated. Monitored constraints differ from operating constraints in that a monitored constraint can never be imposed directly as a well target. The constraints are checked and the actions are applied only after a timestep has been completed; actions are not applied part way through a timestep.

### FORMAT:

\*MONITOR monitored\_constraint value action

Where:

monitored\_constraint for producers is one of:

*MAX	*GOR
*MAX	*WCUT
*MAX	*WOR
*MAX	*WGR
*BACKFLOW	
*WHYSTAB	
*MIN	*STO
*MIN	*STODWN
*MIN	*STG
*MAX	*STG
*MAX	*STW
*MAX	*STL
*MAX	*WHP
*MAX	*BHP
*MAX	*O2CONC
*MAX	*TEMP

monitored\_constraint for injectors is one of:

*BACKFLOW	
*WHYSTAB	
*MIN	*STG
*MIN	*STW
*MIN	*WHP
*MIN	*BHP

action for injectors is one of:

*STOP	
*SHUTIN ( *REPEAT )	
*NEXTSEG	

action for producers is one of:

- \*STOP
- \*SHUTIN (\*REPEAT)
- \*NEXTSEG
- \*AUTOWELL frequency
- \*SHUTLAYER
- \*AUTOLAYER frequency

#### **DEFINITIONS:**

**\*MAX**

Indicates that the monitored constraint is to be a maximum type.

**\*MIN**

Indicates that the monitored constraint is to be a minimum type.

**\*WCUT**

This subkeyword identifies a maximum water-cut (fraction) monitor for a producer. The water-cut is defined as (water\_rate)/(total liquid rate); total liquid rate = water rate + oil rate;

**\*WOR**

This subkeyword identifies a water-oil ratio ( $\text{m}^3/\text{m}^3$  | bbl/bbl |  $\text{cm}^3/\text{cm}^3$ ) constraint.

**\*GOR**

This subkeyword identifies a gas-oil ratio ( $\text{m}^3/\text{m}^3$  | ft<sup>3</sup>/bbl |  $\text{cm}^3/\text{cm}^3$ ) constraint.

**\*WGR**

This subkeyword identifies a water-gas ratio ( $\text{m}^3/\text{m}^3$  | bbl/ft<sup>3</sup> |  $\text{cm}^3/\text{cm}^3$ ) constraint.

**\*BACKFLOW**

This subkeyword identifies backflow monitoring. 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.

**\*WHYSTAB**

This subkeyword identifies wellbore hydraulics stability monitoring. A real well can operate only above a certain rate at which the bottom-hole pressure remains in the stable region of the well-head pressure curve (a region in which the curve of WHP vs. BHP has a positive slope). It can only be used for wells for which a method of computing WHP has been introduced with the \*PHWELLCORE 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.

**\*STO**

This subkeyword identifies a surface oil rate ( $\text{m}^3/\text{day}$  | STB/day) constraint for a producer.

**\*STODWN**

This subkeyword identifies a surface oil rate ( $\text{m}^3/\text{day}$  | STB/day) constraint for a producer. This constraint will be violated only after the produced oil rate has exceeded the specified minimum oil rate.

**\*STG**

This subkeyword identifies a surface gas rate ( $\text{m}^3/\text{day}$  | scf/day) constraint.

**\*STW**

This subkeyword identifies a surface water rate ( $\text{m}^3/\text{day}$  | STB/day) constraint.

**\*STL**

This subkeyword identifies a total surface liquid (oil + water ) rate ( $\text{m}^3/\text{day}$  | STB/day) constraint for a producer.

**\*WHP**

This subkeyword identifies a maximum (producers) or minimum (injectors) well-head pressure (kPa | psi) monitor. It can only be used for wells for which a method for computing WHP has been introduced with the \*PHWELLBORE keywords.

**\*BHP**

This subkeyword identifies a maximum (producers) or minimum (injectors) bottom hole pressure (kPa | psi) monitor.

**\*O2CONC**

Maximum oxygen (component #numy) gas mole fraction of all the blocks completed in this well.

**\*TEMP**

Maximum temperature (C | F) of all blocks completed in this well. It must lie within the allowed temperature range (see \*MINTEMP and \*MAXTEMP).

**value**

A real number specifying the constraint 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.

**\*REPEAT**

The timestep will be reconverged (repeated) after the action is taken.

**\*NEXTSEG**

This subkeyword specifies that if the constraint is violated, the next \*TIME or \*DATE keyword and well changes should be immediately read and applied.

**\*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 only for the following monitors used by producers: \*MIN \*STO, \*MIN \*STG, \*GOR, \*WCUT, \*WGR, \*MAX \*WHP, \*MAX \*BHP and \*WHYSTAB.

**\*SHUTLAYER**

Plug the most offending layer and continue the simulation.

**\*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 can not have the \*AUTOLAYER action specified.

**DEFUALTS:**

Optional keywords. No defaults.

If an action subkeyword is absent, then \*SHUTIN is assumed.

**CONDITIONS:**

If it appears at all, this keyword must follow the list of operating conditions specified via \*OPERATE.

In general any combination of \*MIN and \*MAX for injector or producer and monitor type is allowed; the following combinations are not:

\*WOR with injector  
\*GOR with injector  
\*WGR with injector  
\*O2CONC with injector

**EXPLANATION:**

\*MONITOR defines which performance variables to monitor, as well as the remedial action to take when a constraint is violated. (Use \*OPERATE for constraints which can be used for operating the well, such as oil rate or pressure).

See the EXPLANATION for \*OPERATE.

---

## Well Element Geometry (Conditional)

\*GEOMETRY

### PURPOSE:

\*GEOMETRY specifies well's geometric characteristics to be used by the simulator to calculate the well index internally. See Appendix A.1.

### FORMAT:

\*GEOMETRY ( \*I | \*J | \*K ) rad geofac wfrac skin

### DEFINITIONS:

\*I, \*J, \*K

These subkeywords associate a direction with the well index calculation. For a normal wellbore, it is the grid axis which is parallel to the wellbore; for the linear-pressure-drop option, it is the direction of flow. For purposes of well index calculations, layer thickness refers to the distance across the block in the specified grid direction, that is, well length, and well permeability refers to that in the plane perpendicular to the specified grid direction.

rad

A real number specifying the well radius. (m | ft | cm).

geofac

A real number specifying the geometric factor for the well element. See Appendix A, Figure A.1. This quantity is not used when \*GEOA or \*KHA is used after keyword \*PERF or \*PERFV (see Appendix A.3).

wfrac

A real number between 0 and 1, specifying the fraction of a circle that the well models. See Appendix A, Figure A.1.

skin

A real number specifying the well skin factor.

### DEFAULTS:

If \*GEOMETRY is absent then rad = 8.6 cm and skin = 0, and geofac and wfrac correspond to a full well in the middle of a block (Figure A.1(a), in Appendix A).

### CONDITIONS:

\*PERF and \*PERFV followed by \*GEO, \*GEOA, \*KH, \*KHA or \*TUBE-END will use the \*GEOMETRY values preceding them to calculate well index, and will use the default values if \*GEOMETRY is absent. Therefore, each well with different \*GEOMETRY values should have its own \*GEOMETRY keyword.

Completions in discretized wellbore blocks will ignore \*GEOMETRY data since that information is obtained from keyword \*WELLBORE.

## **EXPLANATION:**

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  
**      rad      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  
2:4      1.  
  
** Well geometries are input for well 3  
**      rad      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  
2:4      1.
```

## **Symmetry Elements**

A well on the boundary of a repeated symmetry element modelled with a Cartesian grid almost always corresponds to case (a) in Figure A.1. This is because the grid is defined so that wells fall on block centres. The resulting element boundaries then determine which portions of the raw grid to eliminate via block modifier keyword \*VAMOD. Therefore, the index is calculated for the full well with wfrac = 1 and geofac = .249. Finally, the appropriate fraction is applied via \*WELL \*FRAC.

If a radial grid is used to model a symmetry element, then case (f) in Figure A.1 is used for the well at the centre of the grid. In this case, you can apply the desired well fraction to wfrac or to \*WELL \*FRAC but not both. Use case (a) for a well not at the centre of the grid.

## Location of Well Completions (Conditional)

\*PERF

### PURPOSE:

Specify the well completion locations and indices.

### FORMAT:

```
*PERF *WI well_name
  { location wi (status) (connection) }
-or-
*PERF (*GEO | *GEOA | *TUBE-END) well_name
  { location ff (status) (connection) }
-or-
*PERF (*KH | *KHA) well_name
  { location kh (status) (connection) }
```

### DEFINITIONS:

#### \*WI

Enter well indices *wi* directly.

#### \*GEO, \*GEOA

Well index of each layer is calculated from the geometric information on the preceding \*GEOMETRY keyword, the dimensions and permeability of the grid block in which the completion occurs and the partial completion factor *ff*. For option \*GEOA, geofac = 0.249 is assumed (see \*GEOMETRY keyword and Appendix A.3). A discretized wellbore completion will use \*TUBE-END if \*GEO or \*GEOA is specified.

#### \*TUBE-END

Well indices of layers are calculated from a linear flow model instead of the usual radial flow model and the partial completion factor *ff*. See **Tube-End Option**, below. A discretized wellbore completion will use this option if \*GEO or \*GEOA is specified.

#### \*KH, \*KHA

Well index of each layer is calculated from *kh* (permeability times completion length) and the geometric information on the preceding \*GEOMETRY keyword. For option \*KHA, geofac = 0.249 is assumed (see \*GEOMETRY keyword and Appendix A.3).

#### *well\_name*

Well name in quotes.

#### *location*

*if ff kf(/ ...(/ irn jrn krn ...) (\*SS)*

*if*, *jf* and *kf* are integer or integer ranges specifying the grid block index of the fundamental grid in the I, J or K direction, respectively, through which the well is completed. (See explanation). At most one direction may have a range.

Optional *irn*, *jrn* and *krn* are similar to *if*, *jf* and *kf*, but at the n'th level of refinement (if applicable) for the perforated grid block. At most one direction may have a range. See “User Block Address” in the manual chapter “Keyword Data Entry System”. UBA qualifiers are not allowed.

Optional \*SS indicates that this is an additional Sink/Source well attached to a discretized wellbore. See EXPLANATION.

#### *wi*

Well index for the layer. See Appendix A.2 for typical well index calculations. See Appendix A.4 for a discussion of backflow in multi-layer wells. Use *wi* = 0 to indicate a shut-in layer.

**Producer:** The layer rate for each phase at reservoir conditions is

$$q = wi * (\text{Phase mobility}) * (P_{\text{block}} - P_{\text{well}})$$

Quantity *wi* is the constant geometric part of the well index, has unit (md-m | md-ft | md-cm) and does not contain a mobility factor.

**\*MOBWEIGHT Injector:** The layer rate for the injected phase at reservoir conditions is

$$q = wi * (\text{Total mobility}) * (P_{\text{block}} - P_{\text{well}})$$

Quantity *wi* is the constant geometric part of the well index, has unit (md-m | md-ft | md-cm) and does not contain a mobility factor. The total mobility is that of the fluid phases in the grid block into which the well is injecting, so this is a type of downstream mobility weighting.

**\*UNWEIGHT Injector:** The layer rate for the injected phase at reservoir conditions is

$$q = wi * (P_{\text{block}} - P_{\text{well}})$$

Quantity *wi* contains both the constant geometric part of the well index and the downhole mobility of the injected phase, and has unit ( $\text{m}^3/\text{kPa-day}$  |  $\text{bbl}/\text{psi-day}$  or  $\text{ft}^3/\text{psi-day}$  |  $\text{cm}^3/\text{kPa-day}$ ). The volume unit corresponds to the phase (liquid or gas) being injected.

#### *ff*

The well index calculated in the \*GEO, \*GEOA or \*TUBE-END option is multiplied by this factor. It can be used to account for partial completions through a grid block. Values of *ff* must be greater than zero.

You may need to over-ride the default of *ff* for a layer completed in a partial block. See **Well Completion in a Partial Block** in the \*VAMOD entry.

*kh*

Permeability times completion length for the layer (md\*m | md\*ft). Layer production and injection rates are calculated as:

q =  $2\pi * kh * wfrac / (\ln(re/rad) + \text{skin}) * (mobility) * (Pblock - Pwell)$   
re =  $\text{geofac} * \sqrt{(\text{areap}) / (3.14159 * wfrac)}$   
see Appendices A.1 and A.3  
rad = wellbore radius  
areap = area perpendicular to the wellbore

For phase production rates, the mobility in a particular phase is used. Phase rates are summed to give total layer rates. Injection mobility is the sum of the water, gas and oil mobilities.

*status*

\*OPEN | \*AUTO | \*CLOSED

\*OPEN

This subkeyword specifies that the well layer is open (perforated). If no *status* keyword is present, then \*OPEN is assumed.

\*AUTO

This subkeyword specifies that the well layer is currently plugged but is a candidate for automatic recompletion. This automatic recompletion is specified by using the \*RECOMPLETE action of the \*MONITOR keyword.

\*CLOSED

This subkeyword specifies that the well layer is a geometrical node point for the purpose of defining the well trajectory. The layer will remain plugged unless overridden by a new \*PERF card selecting the layer status at that location as \*OPEN or \*AUTO. A closed layer may be the reference layer for a well. The pressure drop through a well segment with closed status is accounted for in computing the wellbore pressure differences between completions. No fluid flows from wellbore to reservoir or from reservoir to wellbore in a closed layer.

*connection*

(\*FLOW-TO *ily* | \*FLOW-FROM *ily*) (\*REFLAYER)

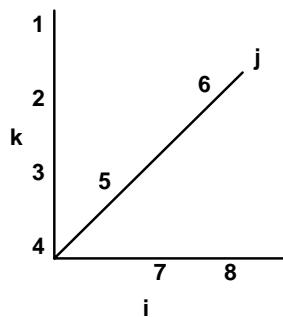
\*FLOW-TO

This keyword 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      connection
    1   1   1:3  1.0   FLOW-TO 'SURFACE' ** 1-3
    1   1       4   1.0   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. In this case by default the reference layer for BHP calculation is the first layer. However, any of the layers 1-4 could have been designated as reference layer with the keyword \*REFLAYER. Layers 5-8 are not eligible as the reference layer since branching has occurred prior to those layers being defined.

*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 keyword 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 the \*FLOW-TO keyword 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 (pwf). 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:**

Once the well's type has been defined with \*PRODUCER or \*INJECTOR, each well must have at least one open perforation defined.

If \*WI, \*GEO, \*GEOA, \*KH, \*KHA and \*TUBE-END are absent, then \*WI is assumed.

A discretized wellbore completion will use \*TUBE-END if \*GEO or \*GEOA is specified.

### **CONDITIONS:**

One of \*PERF or \*PERFV is required.

Each line of I-J-K addresses is allowed to have a range in at most one direction.

\*SS may be used only for attaching a source/sink well to a discretized wellbore block.

### **EXPLANATION:**

This keyword specifies the grid blocks in which a well is completed.

\*PERF is used for horizontal or deviated wells, where the completions are not in a single vertical column of grid blocks. For vertical wells you may use the \*PERFV keyword.

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.

```
*WELL 1 'Producer 1'  
*WELL 2 'Producer 2'  
*WELL 3 'Tubing'      ** Controlling end for tubing  
*WELL 4 'Annulus'    ** Controlling end for annulus  
*WELL 5 'Pump'       ** Location of a pump inside an  
                      annulus that is modelled as  
                      a 'Discretized Wellbore'  
...  
**           rad   geofac wfrac skin  
*GEOMETRY *K 0.375 0.2488 1.0 0.0  
*PERF *GEO 1  
** i j k ff  
12 6 2:4 1.  
13 6 5 .5  
*PERF 2  
** i j k wi  
16 8 4 1.56  
17 8 5 2.34  
18:20 8 6 12.4  
*PERF GEO 3  
** i j k  
1 1 15 / 1 1 1  
*PERF GEO 4  
** i j k  
1 1 15 / 2 1 1    **surface  
*PERF GEO 5  
** i j k  
1 1 12 / 2 1 1 *SS
```

Note for Radial Grid: The innermost radial block is never discretized in the angular direction. This means that a well at the centre ( $i = 1$ ) is completed only for  $j = 1$ ; any  $j > 1$  are called null blocks.

This keyword specifies the grid blocks in which a well is completed within refined grids

```
*WELL 1 'Producer 1'  
*PERF 1  
** if jf kf / ir jr kr wi  
12 6 2:4 / 2 2 1:3 1.78  
13 6 5 / 2 2 1:3 1.5  
  
*WELL 2 'Injector 2'    ** well in hybrid grid  
*PERF 2  
** if jf kf / ir jr kr wi  
12 6 2:4 / 1 1 1 1.78
```

Use \*PERF for a well located in both refined and unrefined blocks. For example, if block (2,3,4) is refined to 3 x 3 x 3 but block (2,3,5) is not refined, use the following.

```
*PERF *GEO 1
** if jf kf / ir jr kr
    2   3   4   /  2   2   1:3
    2   3   5
```

To attach a source/sink well to one end of a discretized wellbore embedded in a hybrid grid, use \*PERF to refer to the innermost hybrid grid block:

```
*PERF *GEO iw          ** attach s/s well to block
** i j k / ir jr kr
    3 4 5 / 1 1 1 / 1 1 1 **regular wellbore
```

## Well Index Units

The injection or production downhole flow rate of a phase in a layer can be written as

$$q = C_g * M_p * (P_{well} - P_{block})$$

where  $C_g$  is a constant that depends only on geometry and absolute permeability, and  $M_p$  is a phase mobility. In all cases except one (see below), the input well index  $wi$  corresponds to the geometric constant  $C_g$ . Appendix A.1 describes how to calculate  $C_g$  from a few wellbore parameters. In fact, the \*GEO-like options merely cause that same  $C_g$  calculation to be done internally from wellbore parameters defined via the \*GEOMETRY keyword. The usual unit for  $C_g$  is md-m for \*SI units and md-ft for \*FIELD units, the same as transmissibility, and a conversion factor is applied to get normal rate units (see Table 7). The  $C_g$  unit is displayed in the well data echo in the output file, and is the unit used for any input  $wi$  values.

The one exception to the definition of quantity  $wi$  is the \*UNWEIGHT injector. Here  $wi$  is  $C_g * M_p$ , where  $M_p$  is the mobility of the phase being injected (see Appendix A.2). The unit of  $wi$  is the converted one but without the viscosity, e.g., bbl/psi-day. In field units the volume unit depends on which phase (liquid or gas) is being injected. The \*QUAL option will require mixing the water and gas phase indices to get the cold-water-equivalent value.

## Well Index Reference Conditions

STARS assumes that  $wi$  does not contain the density ratio, and applies an appropriate density ratio to the downhole rate when constrained by or reporting a surface condition rate. This density ratio is calculated for each Newton iteration, so the ratio is correct for each converged timestep.

## Unweighted Injector with \*GEO-Like Options

All \*GEO-like options are available with an \*UNWEIGHT injector. The density ratio is calculated and applied as described above. A mobility factor is estimated from wellbore temperature \*TINJW and composition \*INCOMP, flashed to reservoir conditions at the time the well is opened. This constant mobility factor is used for subsequent timesteps.

Flashing and density assumptions in this model may not be applicable to the user's particular situation. The use of \*GEO-like options with \*UNWEIGHT is intended to be used only in the absence of actual index data, perhaps to get an initial feel for the well performance. The user can always specify desired well index and mobility via the \*WI option where  $wi$  contains mobility.

## Defining the "Bottom Hole" Location

The very first completion layer defined by the list of block addresses is the "bottom hole" layer; the exact location is the center of this block. This is the location at which the \*BHP pressures will be applied to the wellbore calculations. This block should be at one of the ends of a multi-layer well.

The short form k1(:k2) to refer to grid K layers is convenient if K = k1 for the block that you want as "bottom hole". However, if you want K = k2 instead as "bottom hole" then use the long form, that is, one line for each layer with K = k2 the first one.

## Changing the Number of Layers

The number of completion layers of a well may be increased or decreased in subsequent invocations of \*PERF or \*PERFV. It is not necessary to shut in unused layers by using  $wi = 0$  or  $ff = 0$ . Accumulations for shut-in layers will be printed along with open layers since they contribute to the well's accumulation.

## Direction Default of \*GEO-Like Options

When using a \*GEO-like option be aware of what "layer direction" is being used to calculate well index. The following are the sources for this direction, listed in decreasing order.

1. Layer direction specified via keyword \*LAYERIJK.
2. Layer direction implied by the relationship between the I-J-K indices. If two layers that are adjacent in the data entry order differ in I-J-K index in exactly one direction, that direction is used. This is an expansion of the previous method that used the index range syntax.
3. When attaching a source/sink well to a discretized wellbore block, the wellbore direction is derived from its definition entered in the Reservoir Description data section. Since only one block will be referenced in this case, this will be the determining rule.
4. Layer direction appearing explicitly in the previous appearance of \*GEOMETRY.

A direction is obtained for each perf layer independent of whether or not an index range was used. This may give unexpected results when specifying deviated wells, as shown in the following example.

```
*GEOMETRY *K 0.375 0.2488 1.0 0.0
*PERF *GEO 1
** i j k ff
 1:8 6 2 1. ** Range is I direction
 9 6:8 2 1. ** Range is J direction, but perf layer in
                ** (9,6,2) has direction I since it is
                ** compared to (8,6,2)
```

In this case, if you want to force direction J for (9,6,2), use the \*LAYERIJK keyword.

## Invalid Completions

Well completions are not allowed in null blocks or zero-porosity blocks. When such a block is referenced in \*PERF a warning message is issued and that block is skipped.

## **Discretized Wellbore Option**

A discretized wellbore is merely a collection of grid blocks in a certain configuration, and it needs a source/sink well to specify its operation. This is achieved by attaching a source/sink well to one end, that is, a grid block that is part of the discretized wellbore.

A discretized wellbore is an open pipe or annulus so its connection to a source/sink well corresponds to the tube-end geometry. A discretized wellbore completion will use \*TUBE-END if \*GEO or \*GEOA is specified, since the radial inflow geometry does not apply. Perforation options \*WI, \*KH and \*KHA can be used to specify the corresponding quantities directly.

When the horizontal part of a well is discretized but the vertical leg not, the source/sink well should be attached at the heel of the well. When the vertical leg is discretized then the source/sink well should be attached to the wellbore block closest to the surface.

Sometimes fluid is drawn off by a pump located part way down the discretized wellbore (e.g., to control the liquid level). In this case, the pump should be modelled with an additional source/sink well. In the example specified above, a pump (well No. 5) is located three blocks below the surface (well No. 4). Due to gravity effects, fluid will segregate in the wellbore. Therefore, the pump may draw only liquid to the surface while the other well may produce only gas phase.

Discretized wellbore qualifiers \*TU and \*WB are not allowed. The addressed block (UBA) must be at the heel or surface location of the discretized wellbore. An additional Sink/Source well may be attached to a Discretized wellbore anywhere in the vertical section. This additional well mimics the withdrawal of fluid by a pump. In this case the UBA must be followed by \*SS.

See Appendices A.7 and A.8 for a detailed discussion of the discretized wellbore option.

## **Tube-End Option**

A linear flow model may be used instead of the usual radial well model to estimate well index. The geometrical part of the index is the same as the linear transmissibility calculation for blocks, but the separation distance is from the block centre to the block face (i.e., half the block length). If grid parameters are uniform, the index is twice the transmissibility of the face in the corresponding direction.

This model is appropriate for the end of a core face, such as found in lab-scale core experiments. However, it can be used in any situation in which each well completion block satisfies the following conditions:

1. the block is a fundamental grid block, that is, the block is not the result of local grid refinement;
2. the block lies on the external surface of grid, i.e., its (I,J,K) address satisfies at least one of the following conditions: I = 1, I = NI, J = 1, J = NJ, K = 1 and K = NK.

A candidate direction for transmissibility calculations is one for which all the completion blocks satisfy condition (2) above in that direction. If there is more than one candidate direction for the chosen blocks (e.g., they are all on the edge of the grid), then the following additional criteria are used:

3. If the direction given by the current \*GEOMETRY keyword is a candidate direction, it is chosen, so you may use \*GEOMETRY to manually resolve multiple candidate directions. However, since the \*GEOMETRY direction is the last one defined or defaulted, the result may be unexpected unless \*GEOMETRY is provided explicitly for this well.
4. A candidate direction is eliminated if the grid is not resolved in that direction. For example, candidate direction J would be dropped if NJ = 1 from grid definition keyword \*GRID.

If these additional criteria are insufficient then the data is in error, and \*GEOMETRY must be used to resolve it manually (use the desired direction keyword and zero for each of the remaining numbers). This option is evoked with \*PERF \*TUBE-END.

A discretized wellbore completion using \*TUBE-END does not need \*GEOMETRY since that information is obtained from the corresponding \*WELLBORE data.

### **Limited Entry Perforations**

An injector may be assigned Limited Entry Perforations (LEP) using keyword \*LEP-WELL which replaces the \*MOBWEIGHT or \*UNWEIGHT injection flow calculations described above with steam-specific critical or sub-critical flow rates.

---

## Location of Vertical Well Completions (Conditional)

\*PERFV

### PURPOSE:

Specify the well completion locations and indices for vertical wells. Use \*PERF for a well in a hybrid grid.

### FORMAT:

```
*PERFV *WI well_name
      { kf wi (status) }
-OR-
*PERFV ( *GEO | *GEOA | *TUBE-END ) well_name
      { kf ff (status) }
-OR-
*PERFV ( *KH | *KHA ) well_name
      { kf kh (status) }
```

### DEFINITIONS:

See the manual entry for \*PERF.

### DEFAULTS:

See the manual entry for \*PERF.

### CONDITIONS:

See the manual entry for \*PERF.

Each well in the *well\_list* must have had its block address I-J indices defined via the \*VERT option of keyword \*WELL.

### EXPLANATION:

This keyword specifies the grid blocks in which a vertical well is completed. If several vertical wells are all completed in the same layers, they may all be defined with one \*PERFV keyword.

```
*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
...
**           rad   geofac   wfrac   skin
*GEOMETRY *K 0.375  0.2488  1.0    0.0
*PERFV *GEO 2:3
**     kf     ff
      2:4     1.
      5       .5
*PERFV     1 4
**     kf     ff
      2:4   1.56
      5     1.1
```

See also EXPLANATION for \*PERF.

---

## Geometric Data for Deviated Well Completions (Conditional)

\*LAYERXYZ

### 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.

### FORMAT:

\*LAYERXYZ *well\_name*

{locat.} {deviated layer information}  
:  
:

### DEFINITIONS:

*well\_name*

Single quoted well name specifying the well to which the following deviated layer specifications apply. No wildcarding is allowed.

{location}

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn knr} }

These block addresses specify the layers of the well 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 keyword.

{deviated layer information}

(x1 y1 z1 x2 y2 z2 plength | \*UNDEVIATED)

x1 y1 z1

Coordinates (m|ft) of the “entry point” of the wellbore into the perforated grid block. See explanation.

x2 y2 z2

Coordinates (m|ft) of the “exit point” of the wellbore from the perforated grid block. See explanation.

plength

Perforated length (m|ft) of the deviated wellbore within the named grid block. 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.

### **DEFAULTS:**

Layers named under \*LAYERXYZ with geometric information (as opposed to the \*UNDEVIATED subkeyword) are flagged as deviated within the simulator; undeviated is the default status assigned at the beginning of the run. The undeviated status can be re-imposed through the \*UNDEVIATED subkeyword.

### **CONDITIONS:**

The named layers must all have been previously created for the specified well with \*PERF or \*PERFV lines using the \*GEO or \*GEOA option for computation of well indices. If another well index calculation was specified, the layer is rejected as invalid for deviated perforation. If geometric information is supplied, all seven of the real numbers x1, y1, z1, x2, y2, z2, plength 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.

A well completion in a hybrid grid (see \*REFINE \*HYBRID) cannot be deviated. Such a completion should not appear in the \*LAYERXYZ list. If it does appear, it should be \*UNDEVIATED. If it appears with deviation data, it will be set internally to undeviated.

### **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 x1, y1, z1 and x2, y2, z2 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 gives the actual perforation length through the named grid block. It is valid for plength to exceed the distance between (x1, y1, z1) and (x2, y2, z2) in order to allow more freedom in matching well indices. However the distance between (x1, y1, z1) and (x2, y2, z2) must be positive; if (x1, y1, z1) and (x2, y2, z2) 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 applies to the deviated layer. If the user intends that plength 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

$$WI = 2\pi * wfrac * K * plength * ff / (\ln(re/rw) + ss)$$

The well angular fraction wfrac, the completion factor ff, the well radius rw, and the skin term ss are entered on the relevant \*GEOMETRY or \*PERF.. lines. plength is read directly in the \*LAYERXYZ data.

The drainage radius re is computed from the information entered under \*LAYERXYZ as follows. When the wellbore is parallel to the D axis (D is I, J, or K), re(D) is calculated as

$$Re(D) = geofac * \sqrt{V / (\pi * xh(D) * wfrac)}$$

Where,  $V$  is the bulk volume of the perforated grid block,  $xh(D)$  is the grid block thickness in the direction  $D$ , and  $geofac$  is the geometric factor entered with the \*GEOMETRY keyword.

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  $\mathbf{u}$  be a unit vector in the wellbore direction ( $x2 - x1, y2 - y1, z2 - z1$ ). It makes no difference whether  $\mathbf{u}$  points with or against the fluid flow in the wellbore. Let  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$  be unit vectors pointing in the local I, J, and K directions 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  $x1$ ,  $y1$ , etc. are defined) nor are the vectors  $\mathbf{i}$ ,  $\mathbf{j}$ , and  $\mathbf{k}$  necessarily mutually orthogonal.

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  $\sin^{**2}(\theta_I)$  as  $1 - \cos^{**2}(\theta_I)$  and similarly for J and K. Then the interpolated value of  $re$  to apply in the direction of the wellbore is

$$re(u) = (re(I)*\cos^{**2}(\theta_I)*\sin^{**2}(\theta_J)*\sin^{**2}(\theta_K) + \\ re(J)*\cos^{**2}(\theta_J)*\sin^{**2}(\theta_K)*\sin^{**2}(\theta_I) + \\ re(K)*\cos^{**2}(\theta_K)*\sin^{**2}(\theta_I)*\sin^{**2}(\theta_J))/S,$$

where  $S$  is the sum of the three trigonometric weighting factors in the numerator.

The completion planar averaged permeability  $K$  is computed similarly except that  $re(I)$ ,  $re(J)$ , and  $re(K)$  are replaced with

$$\begin{aligned} K(I) &= \sqrt{K_y * K_z} \\ K(J) &= \sqrt{K_z * K_x} \\ K(K) &= \sqrt{K_x * K_y} \end{aligned}$$

## EXAMPLES:

Example 1:

```
*LAYERXYZ  'WELL-NNE17'
 65 23 5
** x1      y1      z1      x2      y2      z2      plength
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      plength
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 x1, y1, z1, x2, y2, z2 and plength.

---

## 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 *well\_name*

{locat.} {layer direction}  
:  
:

### DEFINITIONS:

*well\_name*

Single quoted well name specifying the well to which the following deviated layer specifications apply. No wildcarding is allowed.

{location}

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn krn} }

These block addresses specify the layers of the well 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. Layer range is not allowed. Any layer named under \*LAYERIJK must already have been defined for the well using a \*PERF or \*PERFV statement.

{layer direction}

(\*I | \*J | \*K | \*UNDEVIATED)

\*I

Signifies the layer is perforated in the local I direction.

\*J

Signifies the layer is perforated in the local J direction.

\*K

Signifies the layer is perforated in the local K direction.

\*UNDEVIATED

Sub-keyword indicating that the layer is henceforth to be treated as an un-deviated 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; un-deviated is the default status assigned at the beginning of the run. The un-deviated status can be re-imposed through the \*UNDEVIATED sub-keyword.

## **CONDITIONS:**

The named layers must all have been previously created for well wn with \*PERF or \*PERFV lines using the \*GEO or \*GEOA option for computation of well indices. If another well index calculation was specified (example \*KH), the layer is rejected as invalid for deviated perforation. Not all of a well's layers need to be named under \*LAYERIJK; those omitted are treated as un-deviated according to the original \*PERF specification.

A well completion in a hybrid grid (see \*REFINE \*HYBRID) cannot be deviated. Such a completion should not appear in the \*LAYERIJK list. If it does appear, it should be \*UNDEVIATED. If it appears with deviation data, it will be set internally to undeviated.

## **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.

---

## **Limited Entry Perforations (Optional)**

\*LEP-WELL, \*LEP-DIAMETER,  
\*LEP-DISCHARGE-COEFF, \*LEP-DISCHARGE-COEFF-CNST

### **PURPOSE:**

Use limited entry perforations (LEP) for specified injectors.

### **FORMAT:**

```
*LEP-WELL well_list
*LEP-DIAMETER well_list
    diam_list
*LEP-DISCHARGE -COEFF well_list
    dcoef_list
*LEP-DISCHARGE -COEFF-CNST well_list
    dcoef_list
```

### **DEFINITIONS:**

#### *well\_list*

One or more quoted well names to specify the wells to which this definition applies. These names must be on the same line as the keyword. If more than one line is required for the well list, the keyword must be repeated. See **Wildcarding Well Names** at the beginning of this chapter.

#### *diam\_list*

List of perforation diameters (m|ft), one for each well in order in *well\_list*. All values must appear on a new line immediately after the \*LEP-DIAMETER line.

#### \*LEP-DISCHARGE -COEFF

Discharge coefficient will be altered with gas content.

#### \*LEP-DISCHARGE -COEFF-CNST

Discharge coefficient is constant.

#### *dcoef\_list*

List of discharge coefficients (dimensionless), one for each well in order in *well\_list*. All values must appear on a new line immediately after the \*LEP-DISCHARGE -COEFF line.

### **DEFAULTS:**

When \*LEP-WELL is absent for a well, that well's flow rates are calculated in a standard way as described in APPENDIX A.

When \*LEP-DIAMETER is absent for a well, that well's perforation diameter is assumed to be 0.0125 m or 0.041 ft (12.5 mm).

When \*LEP-DISCHARGE-COEFF and \*LEP-DISCHARGE-COEFF-CNST are absent for a well, that well's discharge coefficient is assumed to be 0.87 and constant.

## CONDITIONS:

These keywords must be located in the WELL AND RECURRENT DATA keyword group.

## EXPLANATION:

These optional keywords are used to describe a special way of perforating wells – Limited Entry Perforations (LEP). LEP are used in the field to achieve a desired steam distribution in the reservoir by designing the correct number, size and placement of these perforations. LEP are designed in such a way that a choked (critical) flow occurs when fluid flows through the perforation. The implementation of LEP in STARS is based on published papers: “Critical and Subcritical Flow of Multiphase Mixtures Through Chokes”, T.K. Perkins; SPE Drilling & Completion, December 1993, “Critical Flow of Wet Steam Through Chokes,” Sze-Foo Chien; JPT, March 1990 and “Targeted Steam Injection Using Horizontal Wells with Limited Entry Perforations”, T.J. Boone, D.G. Youck, S. Sun; JCPT, January 2001, Vol. 40, No. 1.

The LEP option replaces the perforation flow calculations described in \*PERF (e.g., for \*MOBWEIGHT or \*UNWEIGHT injector) with the following critical or sub-critical flow rates. When a Discretized well is specified as an LEP well then the flow between a reservoir block and a wellbore block is substituted with the following calculations. Critical flow occurs when the ratio of downstream to upstream pressure is less than the critical pressure ratio  $F_p^*$ . Quantity  $F_p^*$  is evaluated from a maximum possible mass flow rate [ $\text{kg/s-m}^2$ ] as  $dq_m/dF_p=0$ . Mass flow rate is calculated as follows:

$$q_m = \left[ \frac{2P_{up}\lambda\rho_g(l - F_p^{k-1/k}) + \alpha(1 - F_p)}{(fgF_p^{-1/k} + \alpha)^2 - A_{dw}^2/A_{up}^2(fg + \alpha)^2} \right]^{0.5}$$

$P_{up}$  = upstream (stagnation) pressure [Pa]

$\rho_g, \rho_o, \rho_w$  = upstream mass density of gas oil and water [ $\text{kg/m}^3$ ]

$fg, fo, fw$  = mass fraction of gas, oil and water

$A_{dw}, A_{up}$  = downstream and upstream area [m]

$k$  = ratio of mixture heat capacities at constant pressure to constant volume

$\alpha = \rho_g(fo/\rho_o + fw/\rho_w)$

$$\lambda = fg + \frac{(fgc_{vg} + foc_{vo} + fwc_{vw})M}{ZR}$$

$c_{vg}, c_{vo}, c_{vw}$  = heat capacities of gas, oil and water at constant volume [J/kg-C]

$M$  = molecular mass [kg/mole]

$Z$  = compressibility factor

$R$  = gas constant

Critical volumetric flow [ $\text{m}^3/\text{s}$ ] is calculated as:

$$q^* = A C_d q_m / \rho$$

$q^*$  - critical flow [ $\text{m}^3/\text{s}$ ]

A - LEP area [ $\text{m}^2$ ]

$\rho$  - mixture mass density [ $\text{kg}/\text{m}^3$ ]

$C_d$  - discharge coefficient – dimensionless

When \*LEP-DISCHARGE-COEFF is used then the input value of discharge coefficient is corrected with gas content as:

$$C_d = \max(C_{di} f_g^{0.031}, 0.61 C_{di})$$

$C_{di}$  – discharge coefficient read in as data

When pressure ratio is greater than the critical pressure ratio then the flow becomes sub-critical and the injection/production rate is calculated as:

$$q = q^* \left[ 1 - \left[ \left( F_p - F_p^* \right) / \left( 1 - F_p^* \right) \right] \right]^{0.5}$$

$q$  - sub-critical flow rate [ $\text{m}^3/\text{s}$ ]

Sometimes the number of LEP in a block (perforation) may be different from 1. When LEP is used with a Sink/Source well then a value of FF on the perforation card \*PERF should be set to the required number. When no LEP is in a block then that perforation should be specified as CLOSED (see \*PERF). When LEP is used with a Discretized well then \*TRANSWB should be used instead of FF.

---

## 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 well_name
  {location}    (pressure_gradient_value | *DEFAULT)
  :
```

### DEFINITIONS:

*well\_name*

Single quoted well name specifying the well to which the layer pressure gradient specifications apply. No wildcarding is allowed.

{location}

if jf kf / ir1 jr1 kr1 { / ... { / irn jrn krn} }

These block addresses specify the layers of the well to be assigned user-specified head pressure gradients. It is valid to name some of the well's layers and not others in the \*LAYERGRAD lines, but layers not named do not acquire a special gradient value and have heads calculated in the normal way. Any layer named under \*LAYERGRAD must already have been defined for the well using a \*PERF or \*PERFV statement. The gradient specified will be used for the head calculation between the named layer and the next heel-ward completion (the completion to which the named completion flows – see the discussion in the manual page for the \*PERF keyword).

pressure\_gradient\_value

Non-negative real value ( kPa/m | psi/ft ) specifying the gradient to be used for the pressure difference calculation.

\*DEFAULT

Specifies that the pressure difference between the named layer and the next heel-ward layer should be calculated normally, not using a specified pressure gradient value.

### 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 the specified well 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 the 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 statically, using a pressure gradient depending on local fluid densities. 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

$$\Delta P = (P_{\text{next}} - P_{\text{named}}) = hgrad * (\text{depth}_{\text{next}} - \text{depth}_{\text{named}})$$

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, effectively maintaining a liquid level control. For example, a pump is placed in a wellbore 6 ft below the desired liquid level, and the pump is set to flow when the pressure at the pump exceeds atmospheric pressure plus the liquid head at a 6 ft depth. This situation would be modelled by assigning head gradients of air (~0) above the liquid level and the expected liquid gradient (water is 9.8 kPa/m or 0.43 psi/ft) below the liquid level.

In addition, the \*BHPDEPTH option allows you to specify the BHP reference depth at a pump location that does not fall at a grid block centre. The user must ensure that the resulting wellbore pressures are not below the physical minimum (e.g., 1 atm). In addition, keyword \*BHPGRAD allows you to specify the pressure gradient value which will be used to compute the difference between the wellbore pressure in the reference completion and the bottom-hole reference depth.

## 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.

---

## 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

### DEFINITIONS:

#### well\_list

One or more quoted well names to specify the wells to which this specification of reference depth applies. See **Wildcarding Well Names** at the beginning of this chapter.

#### depth\_values

A list consisting of either non-negative real numbers ( m | ft) or the subkeyword \*DEFAULT. If the depth\_values list contains only a single entry then this entry will be applied to all wells in well\_list; if there are more than a single entry in the depth\_values list, the number of entries must match the number of wells in well\_list and the first depth value is applied to the first well, etc. The numbers representing depths must contain a decimal point to identify them as real numbers and distinguish them from well numbers.

#### \*DEFAULT

When \*DEFAULT occurs in the depth\_value list, it restores the well to the default state in which the 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 upon the distribution of the well list or depth value list over different lines in the data set; the depth values may begin on the same line as the last well identifier in the list and continue over as many lines as necessary. The well list too may be spread over more than a single line.

## **EXPLANATION:**

When a BHP reference depth is entered for a well using \*BHPDEPTH, the BHP differs from the wellbore pressure in the well's reference completion. The pressure difference is equal to

$$\text{delp} = g * \rho * (\text{ref\_depth} - \text{completion\_depth})$$

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

### DEFINITIONS:

#### well\_list

One or more quoted well names to specify the wells to which this specification of pressure gradient applies. See **Wildcarding Well Names** at the beginning of this chapter.

#### gradient\_values

A list consisting of either non-negative real numbers (kPa/m | psi/ft) or the subkeyword \*DEFAULT. If the gradient\_values list contains only a single entry then this entry will be applied to all wells in well\_list. If there are more than a single entry in the 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

When \*DEFAULT occurs in the gradient\_value list, it restores the well to the default state in which a mobility-weighted fluid density in the reference completion is used to determine the pressure difference between reference depth and the reference completion.

### DEFUALTS:

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

$$\text{delp} = \text{p\_grad} * (\text{ref\_depth} - \text{completion\_depth})$$

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.

---

## **Alter Primary Well Operating Constraint Value (Optional)**

**\*ALTER**

### **PURPOSE:**

\*ALTER allows modification of only the primary operating constraint for the specified wells. 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 quoted well names to specify the wells to which this alteration of the primary operating constraint applies. These names must be on the same line as the \*ALTER keyword. If more than one line is required for the well list, then the \*ALTER keyword must be repeated. See **Wildcarding Well Names** at the beginning of this chapter.

**values**

One number for each well identified in well\_list specifying the new value of the primary operating constraint. Values must appear on a new line immediately after the \*ALTER line.

### **DEFAULTS:**

Optional keywords. No defaults.

### **CONDITIONS:**

\*ALTER must be located in the WELL AND RECURRENT DATA keyword group, and may appear anywhere in this keyword group following the initial \*OPERATE declarations for all of the wells in the \*ALTER list.

All of the wells read in the list must have already had the well type defined. If a listed well has not yet had its type defined, an error message is issued and the run is terminated.

### **EXPLANATION:**

This optional keyword is used to alter the primary operating constraint for a well or set of wells without having to redefine all of the additional operating constraints. It is an effective method of altering values when performing a history match.

\*ALTER, followed by a non-zero value, also opens a well if the well has been shut in by a previous action or if the well has been initially defined as a shut in well.

When \*ALTER is encountered in a data set, the simulator checks if the primary constraint with the new value becomes the most restrictive well constraint. If so, the well is switched to the primary constraint. If not, the new value is entered for the primary constraint but the well is switched to (or continues to run on) the currently most restrictive constraint.

If a primary constraint value of zero is specified using \*ALTER for an open well then that well is shut-in. If a non-zero value is specified for a shut-in well then that well is opened.

Examples:

```
*PRODUCER 'Prod'  
*OPERATE *MAX *STO 500.00  
. . .  
*ALTER 'Prod'  
    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 'Producer 1'  
*OPERATE *MAX *STO 500.00  
*PRODUCER 'Producer 2'  
*OPERATE *MAX *STO 750.00  
*PRODUCER 'Producer 3'  
*OPERATE *MAX *BHP 2500.0  
*INJECTOR 'Injector 1'  
. . .  
*TIME 1200.  
** At a later date, want to adjust the operating  
** constraint values.  
*ALTER 'Producer 1' 'Producer 2' 'Producer 3'  
** values  
    2*1000.0 800.0
```

---

## Alter Well Constraint Value (Optional)

\*TARGET

### PURPOSE:

\*TARGET allows modification of any previously specified well constraint value or the specification of a new constraint type and value for well(s).

### FORMAT:

```
*TARGET  constraint_type well_list  
        value_list
```

### DEFINITIONS:

#### *constraint\_type*

A constraint type that is valid for \*OPERATE for wells in *well\_list*. See \*OPERATE for the full list and explanations of constraint types.

#### *well\_list*

One or more 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 **Wildcarding Well Names** at the beginning of this chapter.

#### *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*.

### DEFAULTS:

Optional keyword. No defaults.

### CONDITIONS:

For each well in *well\_list*, this keyword may appear only after that well's initial \*OPERATE specification. In addition, the well's type (injector/producer) must have already been specified; otherwise, an error message is issued and the run is terminated.

### EXPLANATION:

This optional keyword is used to alter a constraint value for a well without having to respecify all the other operating constraints. It is an effective method of altering constraints when performing a history match.

\*TARGET opens a well if the well was shut in by a previous action or if the well was initially specified as \*SHUTIN. When \*TARGET is encountered in a data set, the simulator checks if the altered constraint with the new value becomes the most restrictive well constraint. If so, the well is switched to this new constraint type. If not, the new value is entered for the constraint but the well is switched to (or continues to run on) the currently most restrictive constraint.

Do not use \*TARGET to add new constraints for types \*STO\_COMP, \*STG\_COMP, or \*STEAMTRAP. Those constraint types may be specified only through \*OPERATE since they require more information than just the constraint values.

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 (a) the re-specification of any operating constraint type and (b) the specification of any new constraint type except those noted above.

**Examples:**

```
*WELL 1 'Producer'  
*PRODUCER 'Producer'  
*OPERATE *MAX *STO 500.0  
*MONITOR *MIN *STO 10.00 *SHUTIN  
:  
*TIME 100  
*TARGET *STO      ** Alter *STO constraint value  
    'Producer'  
    750
```

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 'Producer 1'  
*OPERATE *MAX *STO 500.0  
*MONITOR *MIN *STO 10.00 *SHUTIN  
*PRODUCER 'Producer 2'  
*OPERATE *MAX *STO 750.0  
*MONITOR *MIN *STO 10.00 *SHUTIN  
*PRODUCER 'Producer 3'  
*OPERATE *MIN *BHP 2500.0  
*MONITOR *MIN *STO 10.00 *SHUTIN  
*INJECTOR 'Injector 1'  
*OPERATE *MAX *STW 100.0  
*MONITOR *MIN *STW 10.00 *SHUTIN  
:  
*TIME 1200.  
** Adjust the constraint values  
*TARGET *STO  
    'Producer 1' 'Producer 2'  
    2*1000.0  
*TARGET *BHP 'Producer 3'  
    800.0  
*TARGET *STW 'Injector 1'  
    50.0
```

---

## **Resetting Well Operating Constraint after Value Change (Optional)**

**\*MRC-RESET**

### **PURPOSE:**

\*MRC-RESET allows the user to specify on a well-by-well basis that a well should or should not be set to its Most Restrictive Constraint after a change in the value of an operating constraint (through data read under the \*OPERATE or \*ALTER keywords).

### **FORMAT:**

\*MRC-RESET well\_list      (\*RESET)  
                                  (\*NO-RESET)

### **DEFINITIONS:**

#### **well\_list**

One or more quoted well names to specify the wells to which this alteration of initialization frequency applies. See **Wildcarding Well Names** at the beginning of this chapter.

#### **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. 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. The first operating constraint in the list is used.

### **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. If a well list is included in the \*MRC-RESET line, then the \*MRC-RESET line must follow all of the \*WELL lines which define the wells in the list.

## **EXPLANATION:**

When operating constraint values are changed the type of the most restrictive constraint may also change; for example, when a rate constraint value is increased, the 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, to converge 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 or \*ALTER 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  
*ALTER *STO  500.0
```

would have the oil rate constraint value of well SPECIAL-PRODUCER set to 500 units per day and the operating constraint would stay as it is (presumably on the oil rate constraint), regardless of whether the oil rate constraint is the most restrictive constraint at the beginning of a timestep.

Choosing the optimal operating constraint for a discretized circulating wellbore is very complex because the tubing and annulus streams are closely coupled but have separate operating conditions. In some situations all specified operating conditions in the annulus can not be satisfied because of the tubing operating conditions. When tubing or annulus is shut in, specify the same operating conditions for both. This helps in pseudo-steady state initialization (see keyword \*TRANSIENT in Reservoir Description Section).

---

## Cyclic Steam Stimulation Groups

\*CYC\_GROUP

### PURPOSE:

Specify well groups for Cyclic Steam operation.

### FORMAT:

\*CYC\_GROUP group-number \*INCLUDES { well-name }

### DEFINITIONS:

group-number

An integer representing the cyclic wells group sequence number. Numbering must start from 1 and increase sequentially.

well-name

Well name enclosed in quotes. These names were defined in a series of \*WELL keywords. Only two wells (injector and a producer) are allowed per group.

### DEFAULTS:

If \*CYC\_GROUP is absent and steam stimulation option is used, then only one well cyclic group is assumed.

### CONDITIONS:

All the wells referred to by \*CYC\_GROUP must have been defined previously in the data.

### EXPLANATION:

When cyclic steam stimulation process is used in a field operations with more than one well pair (injector + producer) then it is necessary to define a well group for each well pair. Wells in a group will alternate cycles as specified.

## **Automatic Switching between Steam Cycles**

**\*INJ\_C\_SWT,**

**\*PROD\_C\_SWT, \*IN\_PR\_SHUT, \*PR\_IN\_SHUT**

### **PURPOSE:**

Specify conditions for switching between injection, injection - production soak, production and production - injection shut-in cycles for a Cyclic steam stimulation.

### **FORMAT:**

<b>*INJ_C_SWT</b>	(group_number)	(*MAX_BHP)	x
		(*TOT_TIME)	
		(*TOT_WATR)	
		(*TOT_HEAT)	
		(*MIN_QWTR)	
		(*DTWCYC)	
<b>*PROD_C_SWT</b>	(group_number)	(*MIN_BHP)	x
		(*TOT_TIME)	
		(*MIN_QOIL)	
		(*TOT_LIQ)	
		(*DEPL_NDX)	
		(*DTWCYC)	
<b>*IN_PR_SHUT</b>	(group_number)	(*TOT_TIME)	x
		(*DTWCYC)	
<b>*PR_IN_SHUT</b>	(group_number)	(*TOT_TIME)	x
		(*DTWCYC)	

### **DEFINITIONS:**

**\*INJ\_C\_SWT**

This keyword identifies injection cycle switching conditions.

group-number

An integer or list of integers representing the cyclic well group sequence number which was defined with **\*CYC\_GROUP**.

**\*MAX\_BHP x**

This keyword specifies a maximum bottom-hole pressure (kPa | psi | kPa) as a condition for injection cycle duration.

**\*TOT\_TIME x**

This keyword specifies total time for cycle duration (day | day | min).

**\*TOT\_WATR x**

This keyword specifies a total steam (CWE) injection ( $m^3$  | bbl |  $cm^3$ ) per each cycle as a switching condition.

**\*TOT\_HEAT x**

This keyword specifies a total heat injection (J | BTU | J) per each cycle as a switching condition.

**\*MIN\_QWTR x**

This keyword specifies a minimum steam (CWE) injection rate ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ) as a switching condition. This switching is activated only after steam injection rate was higher than the specified value.

**\*DTWCYCYC x**

This keyword specifies the timestep size (days | days | min) at the beginning of a cycle.

**\*PROD\_C\_SWT**

This keyword identifies production cycle switching conditions.

**\*MIN\_BHP x**

This keyword specifies a minimum bottom-hole pressure (kPa | psi | kPa) as a condition for a production cycle duration.

**\*MIN\_QOIL x**

This keyword specifies a minimum oil production ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ) as a switching condition. This constraint is checked only if the well's current \*OPERATE is \*MIN \*BHP, and only after oil production rate has exceeded the specified minimum value at least once in the cycle.

**\*TOT\_LIQ x**

This keyword specifies total liquid production ( $\text{m}^3$  |  $\text{bbl}$  |  $\text{cm}^3$ ) per each cycle as a switching condition.

**\*DEPL\_NDX x**

This keyword specifies a depletion index for a cycle group in each cycle. It is the ratio of produced liquid to injected steam in CWE in a current cycle.

**\*IN\_PR\_SHUT**

This keyword identifies switching conditions for a soak between injection and production cycle.

**\*PR\_IN\_SHUT**

This keyword identifies switching conditions for a soak between production and injection cycle.

**DEFUALTS:**

If keywords \*INJ\_C\_SWT, \*PROD\_C\_SWT, \*IN\_PR\_SHUT and \*PR\_IN\_SHUT are absent then there is no automatic switching between cycles.

When keyword \*INJ\_C\_SWT, \*PROD\_C\_SWT, \*IN\_PR\_SHUT or \*PR\_IN\_SHUT is specified without the group number, then the group number is assumed to be 1.

#### CONDITIONS:

Each occurrence of \*INJ\_C\_SWT or \*PROD\_C\_SWT or \*IN\_PR\_SHUT or \*PR\_IN\_SHUT must be followed by at least one switching condition. When a well fraction \*FRAC is used it will be applied to all rates or cumulative values. Only one well in a group may be active.

#### EXPLANATION:

When automatic switching between cycles is activated for a cyclic steam stimulation process then the simulator will check switching conditions for each group. When one of the conditions in a group is violated a message is printed and a grid output and restart will be written if requested. Then the next cycle will start for that group. The other groups will continue with the original cycle until that is violated. Cycles will alternate until the last specified \*TIME is reached. A \*TIME card may be used as in a regular data set. Switching conditions may be changed when a keyword \*INJ\_C\_SWT, \*PROD\_C\_SWT, \*IN\_PR\_SHUT or \*PR\_IN\_SHUT is specified with new conditions after a \*TIME card.

It is recommended that a maximum time for cycle duration is specified with \*TOT\_TIME, especially for a production cycle. Sometimes the other specified switching criteria are approached only asymptotically causing the cycle to last for very long time. For example, this may happen when \*MIN\_QOIL is specified as a switching criterion.

For example: A cyclic steam stimulation is used in a field that has 3 well pairs. Well 1:3 are injectors and 4:6 are producers. Well pairs are well 1 and 4, 2 and 5 and 3 and 6. After wells are defined and operating conditions specified then the switching may be specified.

```
*CYC_GROUP 1 *INCLUDES 1 4    ** Define well pair
*CYC_GROUP 2 *INCLUDES 2 5    ** groups
*CYC_GROUP 3 *INCLUDES 3 6
*INJ_C_SWT 1:3    ** switching conditions for an
                   injection cycle for group 1:3
*TOT_TIME 10.0
*TOT_WATR 2100.0
*DTWCYC 0.02
*IN_PR_SHUT 1:3    ** switching condition for soak
*TOT_TIME 7.0
*PROD_C_SWT 1     ** switching conditions for a
                   production cycle for group 1
*TOT_TIME 348.0
*MIN_QOIL 128.0
*DTWCYC 0.02
*PROD_C_SWT 2:3    ** switching conditions for a production
                   cycle for groups 2 and 3
*TOT_TIME 30.0
*DTWCYC 0.02
*TIME 30.0
*PROD_C_SWT 2:3    ** redefine switching conditions for a
                   production cycle for groups 2 and 3
*TOT_TIME 300.0
*MIN_BHP 20.0
*DTWCYC 0.02
*TIME 1095
```

---

## Gas Lift Option

\*GLIFT, \*INCOMPGL

### PURPOSE:

\*GLIFT specifies gas lift rates for a given set of production wells.

\*INCOMPGL specifies composition of an injected gas.

### FORMAT:

```
*GLIFT *RATE ( well_list ) values
*INCOMPGL ( well_list ) y(1).....y(numy)
```

### DEFINITIONS:

#### *well\_list*

One or more quoted well names to specify the wells to which this setting of lift gas rates or composition applies. These names must be on the same line as the \*GLIFT or \*INCOMPGL keyword. All listed wells must be producers. If more than one line is required for the well list, then the \*GLIFT keyword must be repeated. See **Wildcarding Well Names** at the beginning of this chapter.

#### \*RATE

Gas lift injection rates must be specified directly. This is the only sub-option allowed.

#### *values*

One number for each well identified by *well\_list* specifying the new value of the lift gas rate. *values* must appear on one or more new lines immediately following the \*GLIFT keyword. No values may appear on the same line as the \*GLIFT keyword. The unit is ( m<sup>3</sup>/day | SCF/day). The lift gas rates must be entered for a full well (not partial).

#### \*INCOMPGL

This keyword indicates that composition of injected gas is entered. It is used for the Semi-analytical wellbore model (\*SAMODEL) only.

#### *y(i)*

Mole fractions of injected gas phase. The allowed range for each is 0 to 1. They should sum to one, but will be normalized if not. See keyword \*MODEL. Values must appear on one or more new lines immediately following the \*INCOMPGL keyword.

### DEFAULTS:

When the keyword \*GLIFT is missing then it is assumed that there is no gas lift.

## CONDITIONS:

These keywords must be located in the WELL AND RECURRENT DATA section. \*GLIFT is used together with \*PHWELLBORE. When an element of symmetry is simulated, then the gas rate value must be entered for a full well.

All wells appearing in *well\_list* must already have had their type defined with \*PRODUCER. If any listed well has not yet had its type defined, or is an injector, an error message is issued and the run is terminated.

## EXPLANATION:

\*GLIFT allows for the specification of a gas lift injection rate for any producing well. These rates may be modified at different \*DATE or \*TIME keywords by using the \*GLIFT keyword again. When gas lift rates or composition are different for different wells then they must be specified for each well individually. Each of these injected gas rates are added to the corresponding producer's formation gas rate before calculating the wellbore heatloss and pressure drop. When \*PHWELLBORE \*SAMODEL is used then composition of injected gas **must be** specified by \*INCOMPGL. Each well's tubing hydraulics table specified by \*PHWELLBORE \*TABLE and entered using the \*PTUBE1 keyword should include large enough ranges of gas-oil ratios to model the effects of gas lift operations.

When \*GLIFT is in effect, specified rates are not included in reporting of field gas rates or cumulative gas produced.

Example:

```
When * SAMODEL is used
*GLIFT *RATE 1 3:5 7
1000.0      (all wells will have a rate of 1000.0)
*INCOMPGL 1 3:5 7
0.0 0.0 1.0  (components must have the same order as in *MODEL)
- or-
*GLIFT *RATE 1
200.0
*INCOMPGL 1
0.0 0.0 1.0  (components must have the same order as in *MODEL)
*GLIFT *RATE 2
3000
*INCOMPGL 2
0.0 0.0 1.0  (components must have the same order as in *MODEL)
```

---

## Other Well Attributes

\*TRANSIENT, \*SAMINFO

### PURPOSE:

Specify other well attributes.

### FORMAT:

```
*TRANSIENT well_list (*ON | *OFF)
*SAMINFO (*ON) (*TIME |freq)
- or -
*SAMINFO (*OFF)
```

### DEFINITIONS:

#### \*TRANSIENT

\*ON indicates that the transient behavior of the discretized wellbore associated with this well will be simulated, whereas \*OFF indicates that the transient behavior will not be simulated.

This keyword allows switching of this option with time in the recurrent data.

#### *well\_list*

One or more quoted well names assigned with the \*WELL keyword.

#### \*SAMINFO

\*ON enables a detailed printout for all wells that use \*PHWELLBORE. The quantities printed are pressure, temperature, steam quality (injectors), gas phase fraction (producer), formation temperature at \*RHOLE location and enthalpy at various depths. Printout frequency is controlled by the value of *freq* or \*TIME.

#### \*TIME

Write Semi-analytical wellbore model results to the output file at every time specified by subsequent recurrent \*TIME or \*DATE keywords in the input file.

#### *freq*

Write Semi-analytical wellbore model results to the output (.out) file every *freq* timesteps, where *freq* is a positive integer. If *freq* is 0, no results are written.

### DEFAULTS:

The \*TRANSIENT option in effect remains unchanged until it is explicitly modified.

When \*SAMINFO is absent, \*SAMINFO \*OFF is assumed. When \*SAMINFO \*TIME or *freq* is present then \*ON is assumed. If \*SAMINFO is not followed by *freq*, \*ON or \*OFF, \*SAMINFO \*ON \*TIME is assumed.

## **CONDITIONS:**

Both keywords are optional but, if present, they must appear after \*PERF cards if present or after \*OPERATE cards.

Example:

```
*WELL 2 'INJECTOR'  
*INJECTOR 2  
** For injector use SAM to calculate bottom-hole pressure and  
** quality. It is a horizontal well that enters the formation  
** under an angle different from 90 degrees. A sink/Source  
** well model is used.  
  
*PHWELLBORE *SAMODEL  
RTUBIN 0.15  
DEPTH 460.0  
WLENGTH 600.0  
  
. .  
*INCOMP WATER 1.0 0.0 0.0  
*QUAL 0.8 *TINJW 250.0  
*OPERATE .....  
  
. .  
PERF 2  
1:19 1 24 ** horizontal section  
  
. .  
*PRODUCER 3  
*OPERATE MIN BHP 150.0  
  
. .  
*TRANSIENT 3 *ON    **Well No. 3 is a discretized well and will be  
                    ** initialized to pseudo-steady state  
*SAMINFO *ON *TIME ** Results for SAM well will be written every  
                    ** time card
```

---

## Group Production Constraints (Optional)

\*GCONP

### PURPOSE:

\*GCONP is used to specify group production controls.

### FORMAT:

\*GCONP 'group\_name\_1' 'group\_name\_2' ... 'group\_name\_n'

(*MAX)	(*STO)	value	(*STOP)
	(*STG)		(*CONT)
	(*STW)		(*SHUTALL)
	(*STL)		(*SHUTMOWS)
	(*BHF)		(*SHUTMOW)
			(*SHUTMOL)
			(*SHUTMOLDOWN)
			(*SHUTMOLUP)
(*MIN)	(*MNP)	value	(*STOP)
	(*CPP)		(*CONT)
(*TARGET)	(*STO)	value	
	(*STG)		
	(*STW)		
	(*STL)		
	(*BHF)		
	(*MNP)		
	(*CPP)		
(*VREP)		vrep_frac	
(*RECYCLE)	(*GAS)	recyc_frac	
	(*WATER)		
(*PMAINT)	(*PMSECT)	'sector_name'	
	(*PMTARG)	p targ	
	(*PMCOEF)	c1 c2 c3	
(*IPP)	obsolete		

### DEFINITIONS:

'group\_name\_1', 'group\_name\_2', ... , 'group\_name\_n'

Are the groups to which the following constraints apply. The wells that are connected to each group must already have been specified using the \*WELL keyword. Production targets are apportioned using one of the available apportionment methods specified by \*APPOR-METHOD.

**\*MAX**

Specifies that the constraint is a maximum constraint. This value becomes a target for the group only as the result of a violation with the \*CONT action.

**\*MIN**

Specifies that the constraint is a minimum constraint. This value becomes a target for the group only as the result of a violation with the \*CONT action. \*STOP and \*CONT are the only actions available for MIN constraints, and MNP (manifold pressure) and CPP (compressor or surface pressure) are the only group production constraints which accept the MIN designation.

**\*TARGET**

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 \*TARGET keyword ceases to have an effect. There is no action associated with \*TARGET since a target is not checked for violation.

**\*VREP**

This subkeyword specifies a voidage replacement production target. This indicates that the production wells connected to this group produce an amount of the bottom-hole fluid in proportion to the total bottom-hole fluid injected into the reservoir by the injection wells connected to this group.

**\*RECYCLE**

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

**\*PMAINT**

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

**\*STO**

This subkeyword identifies a surface oil rate ( $\text{m}^3/\text{day}$  | STB/day) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

**\*STG**

This subkeyword identifies a surface gas rate ( $\text{m}^3/\text{day}$  | SCF/day |  $\text{cm}^3/\text{day}$ ) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

**\*STW**

This subkeyword identifies a surface water rate ( $\text{m}^3/\text{day}$  |  $\text{STB}/\text{day}$  |  $\text{cm}^3/\text{day}$ ) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

**\*STL**

This subkeyword identifies a surface liquid rate (oil + water) ( $\text{m}^3/\text{day}$  |  $\text{STB}/\text{day}$  |  $\text{cm}^3/\text{min}$ ) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

**\*BHF**

This subkeyword identifies a bottom hole fluid rate ( $\text{m}^3/\text{day}$  |  $\text{rbbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ ) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

**\*MNP**

This subkeyword introduces a manifold pressure (kPa | psi | kPa) constraint. This may be applied only if all the listed groups have had production specified as going through a manifold with the \*MANIFOLD keyword.

**\*CPP**

This subkeyword introduces a compressor (surface) pressure (kPa | psi | kPa) constraint. This may be applied only if all the listed groups have had production specified as going through a manifold with the \*MANIFOLD keyword.

**\*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) for the sector. The value p targ must be a real number larger than one atmosphere. Defaulted to the current sector pressure.

**\*PMCOEF**

Introduces the control coefficients used in the pressure control strategy. The values (c1, c2, c3) must be non-negative real numbers. Defaulted to the internally estimated values.

**\*IPP**

Obsolete, use \*APPOR-METHOD \*IP instead.

value	Constraint value -- see above for units.
vrep_frac	When the voidage replacement subkeyword is used (*VREP) the vrep_frac is the voidage replacement ratio. A ratio of 1.0 indicates that the bottom-hole fluid injected is completely produced.
recyc_frac	When the recycle subkeyword is used (*RECYCLE) the recyc_frac is the recycled fraction of the indicated surface stream, which is imposed as a group production target. A fraction of 1.0 indicates complete reproduction/recycling of the injected surface stream.
*STOP	Action subkeyword indicating that if the constraint cannot be met then the simulation should be stopped.
*CONT	Action subkeyword indicating that the simulation continues with the violated constraint that becomes the target constraint. This is the default action if no action subkeyword is specified.
*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.
*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 ones with the higher rates of the named surface stream) should be shut.
*SHUTMOW	Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending well (MOW -- the one with the highest rate of the named surface stream) should be shut.
*SHUTMOL	Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending layer (MOL) in the most offending well (the one the highest rate of the named surface stream) should be shut.

### **\*SHUTMOLDOWN**

Action subkeyword indicating that if a maximum stock tank rate is exceeded for a group, then the most offending layer (MOL) and the layers below it in the most offending well (the one with the highest rate of the named surface stream) should be shut.

### **\*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.

### **DEFAULTS:**

Optional keyword. Default is no production constraints on any group except the Field; for every rate (e.g. STO, STG, etc.) for which a target is assigned, the Field receives a default target of 1.0d+15 in order to initialize the apportionment algorithm. Any user-entered Field target overrides this default. 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 \*TARGET, \*VREP, \*RECYCLE and \*PMAINT are exclusive on one timecard, and only the latest entry counts. If a group is assigned by \*GCONP with any of 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. A number of apportionment methods (e.g. guide rate, instantaneous potential, priority ranking, and internal guide rate) can be used to distribute a group production target among the contributing wells or groups through the keyword \*APPOR-METHOD (which is described on a separate manual page). \*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 'Group1'  
*MAX      *STG 100000.0  
*TARGET   *STO  1000.0
```

---

## Group Injection Constraints (Optional)

\*GCONI

### PURPOSE:

\*GCONI is used to specify group injection controls.

### FORMAT:

\*GCONI 'group\_name\_1' 'group\_name\_2' ... 'group\_name\_n'

(*MAX)	(*STG)	value	(*STOP)
	(*STW)		(*CONT)
	(*BHG)		
	(*BHW)		
	(*SOR)		
	(*GMP)		
	(*WMP)		
	(*GCP)		
	(*WCP)		
	(*OIL)		
	(*STF)		
(*TARGET)	(*STG)	value	
	(*STW)		
	(*BHG)		
	(*BHW)		
	(*SOR)		
	(*GMP)		
	(*WMP)		
	(*GCP)		
	(*WCP)		
	(*OIL)		
	(*STF)		
(*VREP)	(*GAS)	vrep_frac	
	(*WATER)		
	(*GMKUP)		
	(*WMKUP)		
(*VREFP)	(*GAS)	vrefp_frac	
	(*WATER)		
(*RECYCLE)	(*GAS)	recyc_frac	(make_up_volume)
	(*WATER)		
(*PMAINT)	(*GAS)	(*PMSECT) 'sector_name'	
	(*WATER)	(*PMTARG) p targ	
		(*PMCOEF) c1 c2 c3	
		(*PMMAXR) d1 d2	
(*IIP)	obsolete		

## **DEFINITIONS:**

'group\_name\_1', 'group\_name\_2', ... , 'group\_name\_n'

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.

**\*TARGET**

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 \*TARGET keyword ceases to have an effect. There is no action associated with \*TARGET since a target is not checked for violation.

**\*VREP**

This subkeyword introduces a voidage fraction injection target. This indicates that the injection wells connected to this group inject such that the voidage created by the producers connected to this group is replaced. In this case \*GAS or \*WATER specifies which phase is to be injected to replace the voidage. If more than one phase is being injected to replace the voidage then there must be one \*VREP keyword for each phase. These primary voidage replacement streams are handled independently. One make-up stream can be supplemented with \*GMKUP or \*WMKUP to meet a total voidage replacement fraction. One of \*GAS, \*WATER, \*GMKUP or \*WMKUP must be present for each \*VREP keyword.

**\*VREFP**

This subkeyword introduces reference volume fraction injection target. This indicates that the injection wells connected to this group inject such that the volume of fluid created by the producers connected to this group is replaced at the same reference condition. The reference pressure is the field hydrocarbon pore-volume weighted average, and the reference temperature is the field bulk-volume weighted average. In this case \*GAS or \*WATER specifies which phase is to be injected to replace the reference volume. If more than one phase is being injected then there must be one \*VREFP keyword for each phase. These reference volume replacement streams are handled independently.

**\*RECYCLE**

This subkeyword introduces a recycling injection target. This indicates that the injection wells connected to this group inject such that the phase produced by the production wells connected to this group as specified by \*GAS or \*WATER is recycled (re-injected) 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).

**\*STG, \*BHG**

This subkeyword identifies a surface or reservoir gas rate (m<sup>3</sup>/day | SCF/day | cm<sup>3</sup>/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<sup>3</sup>/day | STB/day | cm<sup>3</sup>/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.

**\*STO**

This subkeyword identifies a surface oil rate (m<sup>3</sup>/day | STB/day | cm<sup>3</sup>/min) maximum or target. Zero rates are allowed and have the same effect as shutting in all the oil injection (INCOMP OIL) wells connected to that group.

**\*STF**

This subkeyword identifies a surface stock tank fluid rate (m<sup>3</sup>/day | STB/day | cm<sup>3</sup>/min) maximum or target. Zero rates are allowed and have the same effect as shutting in all the injection wells connected to that group.

**\*SOR**

This subkeyword identifies an instantaneous steam-oil-ratio. It is the ratio of injected steam as CWE into all the injectors in a group and the surface oil rate from all the producers in that group.

**\*GMP, \*WMP**

This subkeyword introduces a gas or water manifold pressure (kPa | psi | kPa) injection constraint. This subkeyword can be entered only if all of the listed groups have had gas or water injection identified as going through a manifold with the \*MANIFOLD keyword.

**\*GCP, \*WCP**

This subkeyword introduces a gas or water compressor (surface) pressure (kPa | psi | kPa) injection constraint. This subkeyword can be entered only if all of the listed groups have had gas or water injection identified as going

through a manifold with the \*MANIFOLD keyword. Also, a hydraulics table for calculation of the gas or water manifold-to-surface pressure drop must have been identified for all of the listed groups with the \*GPTABLE keyword.

**\*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) for the sector. The value p targ must be a real number larger than one atmosphere. Defaulted to the current sector pressure.

**\*PMCOEF**

Introduces the control coefficients used in the pressure control strategy. The values (c1 c2 c3) must be non-negative real numbers. Defaulted to the internally estimated values.

**\*PMMAXR**

Introduces the maximum surface rate of the specified injection stream that is available for the pressure maintenance. It is the summation of two sources: a fraction (d1) of the total produced (from the same group) and a make-up rate (d2). The values must be non-negative real numbers. The fraction d1 is nondimensional and defaults to 0. The make-up rate d2 is ( $m^3/day$  | SCF/day |  $cm^3/min$ ) for gas or ( $m^3/day$  | STB/day |  $cm^3/min$ ) for water, and is defaults to 1.0e+20.

**\*IIP**

Obsolete, use \*APPOR-METHOD \*IP instead.

**\*GAS, \*WATER**

Specifies that the gas or water phase is to be injected for voidage replacement, reference volume replacement, recycle or pressure maintenance.

**\*GMKUP, \*WMKUP**

Specifies that gas or water phase is the make-up stream supplemented to meet the total voidage replacement fraction.

**value**

Constraint value -- see above for units.

vrep_frac	When the voidage replacement subkeyword is used (*VREP) the vrep_frac is the voidage replacement ratio. A ratio of 1.0 indicates complete voidage replacement by the specified phase. When several values are entered (one for each of several phases) then the vrep_frac entered for a non make-up phase is applied as a target for that phase independently of the other specified values.
vrefp_frac	When the reference volume replacement subkeyword is used (*VREFP) the vrefp_frac is the reference volume replacement ratio. A ratio of 1.0 indicates complete reference volume replacement by the specified phase. When several values are entered (one for each of several phases) then the vrefp_frac entered for a phase is applied as a target for that phase independently of the other specified values.
recyc_frac	When the recycle subkeyword is used (*RECYCLE) the recyc_frac is the recycled fraction of the indicated surface stream, which is imposed as a group injection target. A fraction of 1.0 indicates complete recycling of the surface stream.
make_up_volume	The amount of make-up gas ( $\text{m}^3/\text{day}$   SCF/day   $\text{cm}^3/\text{min}$ ) or water ( $\text{m}^3/\text{day}$   STB/day   $\text{cm}^3/\text{min}$ ) to be injected with the recycled gas or water.
*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.

#### **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 \*TARGET, \*VREP, \*RECYCLE and \*PMAINT are exclusive in one timecard for the same injection stream (gas or water), and only the latest entry counts. If a group is assigned by \*GCONI with any of the following dependent constraints: \*VREP, \*RECYCLE, or \*PMAINT for any of the injection streams, such a group cannot be assigned by \*GCONP with these constraints for group production. In addition, only one injection stream can be assigned with voidage make-up (\*VREP \*GMKUP

/ \*WMKUP) or pressure maintenance (\*PMAINT) for the same targeted group. Error message will be issued if the consistency of the dependent constraints is violated.

If a non-zero water make-up rate is specified via \*GCONI \*RECYCLE \*WATER, the composition of that water must be supplied via keyword \*WMKCOMP.

When the \*SOR option of \*GCONI \*MAX or \*TARGET is used then the group **must** have at least one steam injector and at least one oil producer.

#### **EXPLANATION:**

\*GCONI is used to specify maximum or target fluid injection rates for the group. \*GCONI can also be used to specify voidage replacement, recycling or pressure maintenance targets. This allows group controls to adjust injection rates in response to production.

**Example:** This is an example using voidage replacement. Here, 50% of the voidage is being replaced by water 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%.

```
*GCONT 'Group1'  
*VREP *WATER 0.5  
*VREP *GAS 0.5
```

**Example:** This sets a stock tank gas target for Group1 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. Group1 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 'Group1'  
*TARGET *STG 1000000.  
*MAX *GMP 2394. *CONT
```

If there is any water cycling injector (defined by \*INCOMP \*CYCLING) contributing to the water recycling target, its injected water phase composition will be determined by summing up all the water components produced by the producers under the same group, in addition to the user-specified recycling make-up rate. The produced water amount can be reduced before re-injection selectively by component using the mask fraction keyword \*WRECYMASK.

A number of apportionment methods (e.g. guide rate, instantaneous potential, priority ranking, and internal guide rate) can be used to distribute a group injection target among the contributing wells or groups through the keyword \*APPOR-METHOD (which is described on a separate manual page).

Example: Use IIP to distribute injection

```
*GCONI 'Group1'  
*TARGET *STW 1000.0  
*APPOR-METHOD *GASI 'Group1' *IP
```

**Pressure Maintenance:** \*PMAINT specifies one special group control which instructs a group or groups to adjust the production (at reservoir condition) or injection rates (at surface condition) in order to maintain the average hydrocarbon pore-volume pressure in a particular region (sector) at a desired level. The group runs essentially on a production (or injection) target rate  $Q_{\text{targ}}$  of bottom-hole (or surface) fluid in reference to that of the group injection (or production),  $Q_{\text{ref}}$ .

$$Q_{\text{targ}} = Q_{\text{ref}} \pm \Delta Q$$

where  $\Delta Q$  is the bottom-hole (or surface) correction rate needed to reach or maintain a certain pressure. The sign  $\pm$  designates for production (+) or injection (-) target.  $\Delta Q$  has been set to zero for normal voidage replacement constraints for production control (\*VREP under \*GCONP) or reference volume replacement constraints for injection control (\*VREFP under \*GCONI) at the hydrocarbon pore-volume weighted average pressure and the bulk-volume weighted average temperature of the designated sector. However, setting  $\Delta Q = 0$  may not exactly maintain the pressure due to the involvement of compressible fluids, and the different reservoir conditions under which  $Q_{\text{targ}}$  and  $Q_{\text{ref}}$  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  $\Delta t$  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  $P_{\text{targ}}$ , and controls the overall accuracy. Larger values of  $c_1$  give a quicker approach to  $P_{\text{targ}}$  but may lead to pressure oscillations or even divergence if it is set too high. As  $P$  gets close to  $P_{\text{targ}}$ , 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  $P_{\text{targ}}$  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  $P_{\text{targ}}$ , at the possible cost of some oscillatory behavior.

The Well Management estimates and sets the above constants internally at the beginning of pressure maintenance if the subkeyword \*PMCOEF is absent after \*PMAINT, by assuming that the pressure follows an exponential decline to the set target under pure depletion of an isothermal point reservoir. It should be noted that there is no guarantee that such a rough estimate may work most favorably in general situations. Nevertheless, the user is recommended to run the simulator with the internally-set coefficients. By judging the pressure convergence behavior in terms of fluctuation and/or shifting, optimal values might then be achieved by tuning the internally-set values at possibly the same orders and input through subkeyword \*PMCOEF. Currently the internal calculation of  $c_1$ ,  $c_2$  and  $c_3$  does not take into account aquifer influx. If aquifer influx is strong it may be necessary for the user to optimize these coefficients.

Example:

```
*DATE 1985 1 1
    *GCONP 'Grp-1' *TARGET *BHF 300.      ** (1)
    *GCONI 'Grp-1' *TARGET *STW 150.      ** (2)
    *GCONI 'Grp-1' *TARGET *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).

**SOR:** Steam-oil-ratio is used to control maximum steam injection rate into injectors specified in a group. Steam injection rate as CWE is determined at the beginning of each timestep from the specified value of SOR and the produced oil from the group's producers. When a group has more injectors then the rate is apportioned among the injectors. The SOR value will not be honoured when the group's producers do not produce oil, e.g., at the start of production. For a SAGD process use the \*SOR option of \*GCONI at the beginning of the production period, not during the circulation period.

It may happen that steam injection will become very small when an SOR value is small. In that case the process might be choked.

---

## **Recycled Water Component Mask for Group Water Recycling (Optional)**

**\*WRECYMASK**

### **PURPOSE:**

\*WRECYMASK is used to specify component reduction factors for group re-injection of produced water remaining after water treatment.

### **FORMAT:**

\*WRECYMASK *group\_list fw(1) ... fw(numx)*

### **DEFINITIONS:**

*group\_list*

One or more group names in quotes.

*fw(i)*

Component i reinjection fraction in the range [0,1]. This fraction reduces the amount reinjected from the amount calculated volumetrically. Use *fw(i)* = 1 for a component which is fully available for reinjection and *fw(i)* = 0 for a component which is not reinjected at all.

### **DEFAULTS:**

If \*WRECYMASK is absent for a group then water recycling for that group will assume that *fw(i)* = 1 for all components.

### **CONDITIONS:**

A group name must have been defined previously by keyword \*GROUP or \*WELL before it can appear in *group\_list*.

### **EXPLANATION:**

The \*WRECYMASK keyword affects injection rates only if a group water recycling injection target is currently in effect (\*GCONI \*RECYCLE) and there is at least one water cycling injector (\*INCOMP \*CYCLING) contributing to that target.

### **Example**

Group “Group1” acquires the specified recycling mask for a five-component water phase description. The last three components will be removed from the reinjection if they are present in the produced water phase.

```
*WRECYMASK 'Group1' 1.0 1.0 0.0 0.0 0.0
```

---

## **Make-up Water Composition for Group Water Recycling (Optional)**

**\*WMKCOMP**

### **PURPOSE:**

\*WMKCOMP is used to specify the composition of water injected as part of a group water recycling target to supplement the recycled fluid.

### **FORMAT:**

\*WMKCOMP *group\_list* *w(1) ... w(numx)*

### **DEFINITIONS:**

*group\_list*

One or more group names in quotes.

*w(i)*

Mole fraction for component i in the make-up water phase used to supplement water reinjection. The allowed range is 0 to 1. The *w(i)* should sum to one; if they do not, they will be normalized internally.

### **DEFAULTS:**

No default.

### **CONDITIONS:**

A group name must have been defined previously by keyword \*GROUP or \*WELL before it can appear in *group\_list*.

If a non-zero group water make-up rate is specified via \*GCONI \*RECYCLE \*WATER, and there is at least one water cycling injector contributing to that target, then specification of make-up composition via keyword \*WKCOMP is mandatory for that group.

### **EXPLANATION:**

The \*WMKCOMP keyword affects injection rates only if a group water recycling injection target is currently in effect (\*GCONI \*RECYCLE \*WATER) with a non-zero make-up rate, and there is at least one water cycling injector (\*INCOMP \*CYCLING) contributing to the target.

### **Example**

Group "Group1" acquires the specified make-up water composition for a seven-component water phase description.

```
*WMKCOMP 'Group1'  
      0.99  0.01  0.0   0.0   0.0   0.0   0.0
```

---

## Water Make-up Target for Group Water Recycling (Optional)

\*WMKUPTO

### PURPOSE:

Specify a total recycling (produced plus make-up) group water injection rate target.

### FORMAT:

\*WMKUPTO *group\_list* *water\_rate\_list*

### DEFINITIONS:

*group\_list*

One or more group names in quotes.

*water\_rate\_list*

One or more non-negative total injected water rate values ( $\text{m}^3/\text{day}$  | STB/day /  $\text{cm}^3/\text{min}$ ). If only one rate value is entered, it is applied to all of the groups listed. If more than one rate value is entered, there must be one rate for each group in the list and the first rate is assigned to the first group, etc.

### DEFAULTS:

If keyword \*WMKUPTO is absent for a group, that group's water make-up rate is controlled by the *make\_up\_volume* entered via \*GCONI \*RECYCLE \*WATER (which itself defaults to zero).

### CONDITIONS:

A group name must have been defined previously by keyword \*GROUP or \*WELL before it can appear in *group\_list*.

### EXPLANATION:

A specified total injected water rate has no effect unless a group water recycling target is currently in force (see the \*GCONI entry). The make-up rate can be specified with \*GCONI \*RECYCLE \*WATER *make\_up\_volume*. The total injection rate can be specified only with \*WMKUPTO. 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 make-up rate entered. The group water injection rate will be set to the total water rate unless this total rate exceeds the injection rate of recycled water by more than a maximum make-up rate, in which case the total water injection rate is set to the recycling rate plus the maximum make-up rate. 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.

### Example

Groups "Group1" and "Group2" both acquire the single specified total water injection rate.

```
*WMKUPTO 'Group1' 'Group2' 300.
```

## 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\_name\_1' 'group\_name\_2' ... 'group\_name\_n'

*GOR	value	(*STOP) (*SHUTALL)
*WCUT		(*SHUTMOWS)
*WGR		(*SHUTMOW)
*MAXGAS		(*SHUTMOL)
*MAXSTW		(*SHUTMOLDOWN) (*SHUTMOLUP)
*MINOIL	value	(*STOP *SHUTALL)
*MINGAS	value	(*STOP *SHUTALL)
*MINBHF	value	(*STOP *SHUTALL)
*MINREC	value	(*STOP *SHUTCYCINJ *SHUTALL)

### DEFINITIONS:

'group\_name\_1', 'group\_name\_2', ... , 'group\_name\_n'

Are the groups to which the following constraints apply. The wells that are connected to each group have already been specified using the \*WELL keyword. The injection and production targets are met by using one of the available apportionment methods specified by \*APPOR-METHOD.

#### \*GOR

This subkeyword identifies a maximum gas-oil ratio ( $m^3/m^3$  | scf/STB |  $cm^3/cm^3$ ) monitor for group production. The \*STOP, \*SHUTALL, \*SHUTMOWS, \*SHUTMOW, \*SHUTMOL, \*SHUTMOLDOWN, and \*SHUTMOLUP actions are valid for this monitor.

#### \*WCUT

This subkeyword identifies a maximum water-cut (fraction) monitor for group production.. The \*STOP, \*SHUTALL, \*SHUTMOWS, \*SHUTMOW, \*SHUTMOL, \*SHUTMOLDOWN, and \*SHUTMOLUP actions are valid for this monitor.

**\*WGR**

This subkeyword identifies a maximum water-gas ratio ( $\text{m}^3/\text{m}^3$  | STB/scf |  $\text{cm}^3/\text{cm}^3$ ) monitor for group production. The \*STOP, \*SHUTALL, \*SHUTMOWS, \*SHUTMOW, \*SHUTMOL, \*SHUTMOLDOWN, and \*SHUTMOLUP actions are valid for this monitor.

**\*MAXGAS**

This subkeyword identifies a maximum gas rate ( $\text{m}^3/\text{day}$  | SCF/day |  $\text{cm}^3/\text{min}$ ) monitor for group production. The \*STOP, \*SHUTALL, \*SHUTMOWS, \*SHUTMOW, \*SHUTMOL, \*SHUTMOLDOWN, and \*SHUTMOLUP actions are allowed for this monitored constraint. The “most offending” well or layer for this constraint is deemed to be the one with the highest GOR rather than the one with the highest gas rate.

**\*MAXSTW**

This subkeyword identifies a maximum water rate ( $\text{m}^3/\text{day}$  | SCF/day |  $\text{cm}^3/\text{min}$ ) monitor for group production. The \*STOP, \*SHUTALL, \*SHUTMOWS, \*SHUTMOW, \*SHUTMOL, \*SHUTMOLDOWN, and \*SHUTMOLUP actions are allowed for this monitored constraint. The “most offending” well or layer for this constraint is deemed to be the one with the highest water cut, rather than the one with the highest water rate.

**\*MINOIL**

This subkeyword identifies a minimum oil rate ( $\text{m}^3/\text{day}$  | STB/day |  $\text{cm}^3/\text{min}$ ) monitor for group production. Only the \*STOP and \*SHUTALL actions are allowed for this monitored constraint.

**\*MINGAS**

This subkeyword identifies a minimum gas rate ( $\text{m}^3/\text{day}$  | SCF/day |  $\text{cm}^3/\text{min}$ ) monitor. Only the \*STOP and \*SHUTALL actions are allowed for this monitored constraint.

**\*MINBHF**

This subkeyword identifies a minimum bottom hole fluid rate ( $\text{m}^3/\text{day}$  | bbl/day |  $\text{cm}^3/\text{min}$ ) monitor. Only the \*STOP and \*SHUTALL actions are allowed for this monitored constraint.

**\*MINREC**

This subkeyword specifies that if the total group water rate available for recycling falls below the specified value ( $\text{m}^3/\text{day}$  | SCF/day |  $\text{cm}^3/\text{min}$ ) then perform the remedial action, which can be one of \*STOP, \*SHUTALL or \*SHUTCYCINJ. The total recycling rate includes the make-up volumes.

**value**

Constraint value -- units are given under \*GOR, \*WCUT, \*WGR, \*MAXGAS, \*MAXSTW, \*MINOIL, \*MINGAS, and \*MINBHF above.

**\*STOP**

Action subkeyword indicating that if the monitored constraint is violated for a group then the simulation should be stopped. This is the default action for all constraints if no action is entered explicitly.

**\*SHUTALL**

Action subkeyword indicating that if a monitored constraint is violated for a group, then all currently open wells in the group should be shut.

**\*SHUTCYCINJ**

Action subkeyword indicating that, if a monitored constraint is violated for a group, all currently open water cycling injectors in the group 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 values of GOR, WCUT, or WGR) should be shut.

**\*SHUTMOL**

This specifies that if a monitor is violated then the most offending layer (MOL) in the most offending well (the one with the greatest GOR, WCUT, or WGR) should be shut.

**\*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 value of GOR, WCUT, or WGR) should be shut.

**\*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 value of GOR, WCUT, or WGR) should be shut.

**\*SHUTMOW**

This specifies that if a GOR, WCUT, WGR, MAXGAS, or MAXSTW monitor is violated then the most offending well (MOW -- the one with the greatest value of GOR, WCUT, or WGR) should be shut.

**DEFUALTS:**

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. Automatic drilling of wells can be triggered when a group fails to meet a rate target by using a \*GAPPOR \*AUTODRILL \*ON line for the group and specifying the well's status as \*AUTODRILL. Please see the manual pages describing these keywords for more information.

## **Example:**

Automatic shutting off of the producer in group GROUP1 having the highest WCUT when the group surface water rate exceeds 50 STB/day (if field units are used).

```
*GCONM 'GROUP1'  
*MAXSTW 50. *SHUTMOW
```

---

## 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.

### FORMAT:

\*MANIFOLD (\*PROD | \*GASI | \*WATI) *group\_list*  
(\*ON | \*OFF)

### DEFINITIONS:

#### \*PROD

Indicates that group production should be treated as going through a manifold.

#### \*GASI

Indicates that group gas injection should be treated as going through a manifold.

#### \*WATI

Indicates that group water injection should be treated as going through a manifold.

#### *group\_list*

Names of groups to which the current manifold specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in *group\_list* must have been previously defined using the \*GROUP keyword.

#### \*ON

Specifies that the manifold treatment is to be imposed.

#### \*OFF

Specifies that the manifold treatment is not to be imposed (or is to be removed).

**DEFAULTS:**

Optional keyword. Default for all groups and target types is \*OFF.

**CONDITIONS:**

This keyword must be located in the Well and Recurrent Data keyword group. The spacing of the subkeywords and group names is not important – they can all be on one line, or lines may be broken arbitrarily. For example, \*ON or \*OFF may be on the same line as are the group names. Only one of the three possible target types may be specified in any one occurrence of the \*MANIFOLD keyword. All wells operating through manifolds must have a means specified for computing tubing-head pressure, as the tubing-head pressure for these wells is interpreted as the manifold pressure. \*MANIFOLD can only be specified for groups having wells attached directly to them. Groups having other groups attached cannot have \*MANIFOLD specified.

**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 \*PHWELLBORE 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 \*PHWELLBORE keyword.

---

## **Pressure-Constraint Translation for Manifolds (Optional)**

**\*PCON-MANIFOLD**

### **PURPOSE:**

\*PCON-MANIFOLD applies only to group streams treated as going through a manifold (see the pages for the \*MANIFOLD keyword). When \*PCON-MANIFOLD is specified, if the group is operating on a manifold or surface pressure constraint, that constraint is translated to a tubing-head pressure constraint and equal tubing-head pressure constraints are assigned to all wells attached to the manifold. Otherwise, the wells are assigned rates constraints which give the correct manifold or surface pressure at the start of the timestep. When \*PCON-MANIFOLD is specified, group pressure constraints are usually honored more accurately, and higher-level rate constraints less accurately, than if it is not specified.

### **FORMAT:**

**\*PCON-MANIFOLD (\*PROD | \*GASI | \*WATI) *group\_list* (\*ON | \*OFF)**

### **DEFINITIONS:**

#### **\*PROD**

Indicates that pressure translation should be applied for pressure constraints upon the production manifolds of the listed groups.

#### **\*GASI**

Indicates that pressure translation should be applied for manifold group gas injection for the listed groups.

#### **\*WATI**

Indicates that pressure translation should be applied for manifold group water injection for the listed groups.

#### ***group\_list***

Names of groups to which the current pressure translation specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in *group\_list* must have been previously defined using the \*GROUP keyword.

#### **\*ON**

Specifies that pressure translation is to be imposed.

#### **\*OFF**

Specifies that pressure translation is not to be imposed (or is to be removed).

### **DEFAULTS:**

Optional keyword. Default for all groups and target types is \*OFF.

## **CONDITIONS:**

This keyword must be located in the Well and Recurrent Data keyword group. \*PCON-MANIFOLD can have an effect only if the specified group streams have been designated to be treated as going through a manifold through use of the \*MANIFOLD keyword. It is not considered an error to specify \*PCON-MANIFOLD when \*MANIFOLD has not been specified, but pressure translation would then have no effect because a non-manifold group can never operate on a pressure constraint. The spacing of the subkeywords and group names is not important – they can all be on one line, or lines may be broken arbitrarily. For example, the group names may be on the same line as the specification of stream type. Only one of the three possible target types may be specified in any one occurrence of the \*PCON-MANIFOLD keyword. When a manifold for which \*PCON-MANIFOLD is specified is running on a rate constraint rather than upon a pressure constraint, \*PCON-MANIFOLD has no effect.

## **EXAMPLES:**

```
*PCON-MANIFOLD *PROD 'Group-1' *ON
```

This specifies that pressure constraints for the production manifold for group ‘Group-1’ are to be translated into well tubing head pressure (manifold pressure) constraints.

```
*PCON-MANIFOLD *GASI 'Group-1' 'Group-2'  
      'Group3' *ON
```

This specifies that pressure constraints for the gas injection manifolds for ‘Group-1’, ‘Group-2’, and ‘Group-3’ should be translated into well tubing-head pressure (manifold pressure) constraints.

---

## Specification of Hydraulics Tables for Calculating Pressure Difference between Manifold and Surface (Optional)

\*GPTABLE

### PURPOSE:

\*GPTABLE (Group Pressure Table) applies only to group targets treated as going through a manifold (see the pages for the \*MANIFOLD keyword). \*GPTABLE specifies the hydraulics table number to be used to compute the pressure drop between manifold and surface, for production and injection manifolds. The hydraulics table can be entered using the \*PTUBE1 keyword for production manifolds or \*ITUBE1 for injection manifolds.

### FORMAT:

\*GPTABLE ( \*PROD | \*GASI | \*WATI ) *group\_list*  
                  *table\_numbers*

### DEFINITIONS:

#### \*PROD

Indicates that the pressure table numbers are being specified for production manifolds.

#### \*GASI

Indicates that the pressure table numbers are being specified for gas injection manifolds.

#### \*WATI

Indicates that the pressure table numbers are being specified for water injection manifolds.

#### *group\_list*

Names of groups to which the current hydraulic table number specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in *group\_list* must have been previously defined using the \*GROUP keyword.

#### *table\_numbers*

List of positive integers giving the hydraulics table numbers. The list may consist either of a single integer, in which case all of the listed groups are assigned this single hydraulics table number, or *table\_numbers* must contain exactly as many entries as does the *group\_list*, in which case the table numbers are assigned to the groups in order.

### DEFAULTS:

Optional keyword. Default for all groups is no assigned hydraulics table number; in which case the surface (compressor) pressure is reported to be equal to the manifold pressure and no surface pressure constraint can be specified.

## **CONDITIONS:**

This keyword must be located in the Well and Recurrent Data keyword group. Only one of the three possible target types may be specified in any one occurrence of the \*GPTABLE keyword. The spacing of the subkeywords, group names, and table numbers is not important – they can all be on one line, or lines may be broken arbitrarily.

## **EXAMPLES:**

```
*GPTABLE *PROD 'Group-1' 'Group-2'  
1
```

This specifies that pressure drops between the production manifolds and surface for ‘Group-1’ and ‘Group-2’ will be calculated using the first hydraulics table.

```
*GPTABLE *GASI 'Group-1' 'Group-2'  
'Group-3' 1 7 4
```

This specifies that the gas injection manifolds of ‘Group-1’, ‘Group-2’, and ‘Group-3’ should have pressure drops between compressor and manifold calculated using hydraulics tables 1, 7, and 4 respectively.

---

## **Manifold Depth for Calculating Pressure Difference between Manifold and Surface (Optional)**

**\*GPHYDDEP**

### **PURPOSE:**

\*GPHYDDEP applies only to group targets treated as going through a manifold (see the pages for the \*MANIFOLD keyword). \*GPHYDDEP specifies the depth used to scale the pressure drop linearly relative to the depth specified with the particular hydraulics table.

### **FORMAT:**

\*GPHYDDEP (\*PROD | \*GASI | \*WATI) *group\_list*  
*depths*

### **DEFINITIONS:**

#### **\*PROD**

Indicates that the depths are being specified for production manifolds.

#### **\*GASI**

Indicates that the depths are being specified for gas injection manifolds.

#### **\*WATI**

Indicates that the depths are being specified for water injection manifolds.

#### *group\_list*

Names of groups to which the current depth specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in *group\_list* must have been previously defined using the \*GROUP keyword.

#### *depths*

List of real numbers (m | ft | cm) giving the manifold depths. The list may consist either of a single number, in which case all of the manifolds for the listed groups are assigned this single depth, or *depths* must contain exactly as many entries as does the *group\_list*, in which case the depths are assigned to the groups in order.

### **DEFAULTS:**

Optional keyword. Default for all manifolds is to use the depth specified with the hydraulics table used for the pressure drop calculation.

### **CONDITIONS:**

This keyword must be located in the Well and Recurrent Data keyword group. Only one of the three possible target types may be specified in any one occurrence of the \*GPHYDDEP keyword. The spacing of the subkeywords, group names, and depth values is not important – they can all be on one line, or lines may be broken arbitrarily.

**EXAMPLES:**

```
*GPHYDDEP *PROD 'Group-1' 'Group-2'  
3512.
```

This specifies that the production manifolds for ‘Group-1’ and ‘Group-2’ will be taken to be 3512 units below the surface when performing the pressure drop calculation between manifold and surface.

```
*GPHYDDEP *GASI 'Group-1' 'Group-2'  
'Group-3' 4564. 3256. 3987.
```

This specifies that the gas injection manifolds of ‘Group-1’, ‘Group-2’, and ‘Group-3’ are assigned depths of 4564, 3256, and 3987 units respectively to be used in scaling tabular calculations of the pressure drop between manifold and compressor.

## Group Artificial Lift Quantity Value (Optional)

\*GROUPALQ

### PURPOSE:

\*GROUPALQ applies only to group targets treated as going through a manifold (see the pages for the \*MANIFOLD keyword) and for which a manifold-to-surface hydraulics table has been specified using the \*GPTABLE keyword. It introduces the value which will be used for the ALQ quantity in interpolations in a hydraulics table entered using the \*PTUBE1 table format. The variable value is arbitrary and can be specified freely in the data. The units of the ALQ quantity are not important so long as the assigned values and the tabular values are consistent.

### FORMAT:

\*GROUPALQ (\*PROD | \*GASI | \*WATI) *group\_list*  
*alq\_values*

### DEFINITIONS:

#### \*PROD

Indicates that the ALQ values are being specified for production manifolds.

#### \*GASI, \*WATI

Indicates that the ALQ values are being specified for gas or water injection manifolds. This subkeyword is accepted but hydraulics tables for injection manifold-to-surface pressure drops do not include the ALQ variable, so this specification does not have any effect in the simulation.

#### *group\_list*

Names of groups to which the current ALQ specification is to apply. Each group name is up to 16 characters long and enclosed in single quotes. All names in *group\_list* must have been previously defined using the \*GROUP keyword.

#### *alq\_values*

List of real numbers giving the ALQ values. The list may consist either of a single number, in which case all of the manifolds for the listed groups are assigned this single value, or *alq\_values* must contain exactly as many entries as does the *group\_list*, in which case the ALQ values are assigned to the groups in order. The units are arbitrary but must correspond to the tabular units.

### DEFAULTS:

Optional keyword. Default for all manifolds is an ALQ value of zero.

### CONDITIONS:

This keyword must be located in the Well and Recurrent Data keyword group. Only one of the three possible target types may be specified in any one occurrence of the \*GROUPALQ keyword. The spacing of the subkeywords, group names, and ALQ values is not important – they can all be on one line, or lines may be broken arbitrarily.

**EXAMPLES:**

```
*GROUPALQ *PROD 'Group-1' 'Group-2'  
1.
```

This assigns ALQ values of 1 for the production manifolds of 'Group-1' and 'Group-2'.

```
*GROUPALQ *PROD 'Group-1' 'Group-2'  
'Group-3' 1. 0.5 0.3
```

This specifies that the production manifolds of 'Group-1', 'Group-2', and 'Group-3' are assigned ALQ values of 1., 0.5, and 0.3 respectively to be used in tabular calculations of the pressure drop between manifold and surface.

## Well Artificial Lift Quantity Value (Optional)

\*WELLALQ

### PURPOSE:

\*WELLALQ specifies an artificial lift quantity value to be used in the tabular tubing head pressure calculation for producers. It introduces the value which will be used for the ALQ quantity in interpolations in a hydraulics table entered using the \*PTUBE1 table format. The variable value is arbitrary and can be specified freely in the data. The units of the ALQ quantity are not important so long as the assigned values and the tabular values are consistent.

### FORMAT:

```
*WELLALQ well_list
          alq_values
```

### DEFINITIONS:

#### *well\_list*

Names or numbers of wells to which the current ALQ specification is to apply. Each well name is up to 40 characters long and enclosed in single quotes. All wells included in *well\_list* must have been previously defined using the \*WELL keyword. Both producers and injectors are valid as members of this list, but hydraulics tables for injectors do not include the ALQ variable, so specifying values for injectors will have no effect in the simulation.

#### *alq\_values*

List of real numbers giving the ALQ values. The list may consist either of a single number, in which case all of the wells in *well\_list* are assigned this single value, or *alq\_values* must contain exactly as many entries as does *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**

### **PURPOSE:**

\*DRILLQ allows the specification of the order in which wells must be drilled automatically.

### **FORMAT:**

\*DRILLQ      \*IPP | \*IIP | well\_list

### **DEFINITIONS:**

\*IPP

Specifies that the instantaneous production potential of producing wells with status \*AUTODRILL will be used to determine the order in which these wells will be drilled, with wells of higher potential opened first. \*IPP must be used for producers.

\*IIP

Specifies that the instantaneous injection potential of injection wells with status \*AUTODRILL will be used to determine the order in which these wells will be drilled, with wells of higher potential being opened first. \*IIP must be used for injectors.

well\_list

One or more quoted well names 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 **Wildcarding Well Names** at the beginning of this chapter.

### **DEFAULTS:**

Optional keywords. Default is to use the instantaneous injection / production potential to determine drilling priority.

### **CONDITIONS:**

\*DRILLQ 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 priority 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 until the list is exhausted. A second well\_list following \*DRILLQ will reset the whole sequence of automatic drilling.

Example:

```
*GCONP 'Group1'  
*MAX *STO 1000.0  
*DRILLQ 'well-4' 'well-5' 'well-6'
```

## Group Apportionment Options (Optional)

\*GAPPOR

### PURPOSE:

\*GAPPOR introduces subkeywords which control how the apportionment of group target among contributing wells or subgroups is to be done. Currently \*AUTODRILL is the only subkeyword supported.

### FORMAT:

```
*GAPPOR      'group_name_1' ... 'group_name_n'  
              *AUTODRILL (*ON | *OFF| stream_identifier )
```

### DEFINITIONS:

'group\_name\_1', ... , group\_name\_n'

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\_identifier )

The \*AUTODRILL apportionment option is turned on or off. \*ON specifies that if one of the listed groups is attempting to meet a rate target, and the group apportionment routine predicts that the contributing wells have insufficient injection or production potential to meet this target, any potentially contributing wells which are shut but have \*AUTODRILL status may be opened automatically to meet the target. \*OFF disables the option.

The *stream\_identifier* takes the form of \*STO, \*STG, \*STW, \*STL or \*BHF which are the available group target streams specified by group control keyword \*GCONP under \*TARGET (see 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.

### **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.

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.

```
*GAPPOR 'GROUP1' *AUTODRILL *ON
```

---

## **Apportionment Method for Meeting Group Targets (Optional)**

**\*APPOR-METHOD**

### **PURPOSE:**

\*APPOR-METHOD specifies the apportionment method to distribute the production or injection rates among contributing wells or groups for meeting the group target(s) defined by \*GCONP or \*GCONI.

### **FORMAT:**

\*APPOR-METHOD      (\*PROD | \*GASI | \*WATI) group\_names  
                          (\*IP | \*GUIDE | \*INGUIDE | \*PRIOR)

### **DEFINITIONS:**

#### **\*PROD**

Indicates that group production targets should be apportioned with the specified method.

#### **\*GASI**

Indicates that group gas injection targets should be apportioned with the specified method.

#### **\*WATI**

Indicates that group water injection targets should be apportioned with the specified method.

#### **group\_names**

A list of group names to which the apportionment method is applied. No wildcard characters may be used in the group names.

#### **\*IP**

This sub-keyword specifies the use of instantaneous potential (IP), which is computed internally by the simulator, to determine the apportionment of production or injection targets among wells. This is default. Each contributing well is assigned a rate in proportion to its maximum rate for the surface stream for which the target is referenced. This has the effect that either the target can be met with all contributors being assigned rates in the same fraction of their maximum rate, or the target cannot be met and all contributors are assigned exactly their maximum rates.

#### **\*GUIDE**

This sub-keyword specifies the use of user-supplied guide rates to determine the apportionment of production or injection targets among contributing wells or groups. Each contributing well or group receives a fraction of the target in proportion to its guide rate. If the assignment violates its maximum rate under the most restrictive constraint, such a well or group will be left out of 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 is not present. Unlike the user-supplied guide rate (\*GUIDE), the internally-generated guide rates are time dependent.

#### \*PRIOR

This sub-keyword specifies the use of priority ranking to determine the apportionment of group targets among contributing wells. Wells are ranked and opened in order of decreasing priority until reaching a swing well. The user-supplied priority formulae and numerical controls are specified by keyword \*PRIOR-FORM (refer to the explanation therein for details).

#### DEFAULTS:

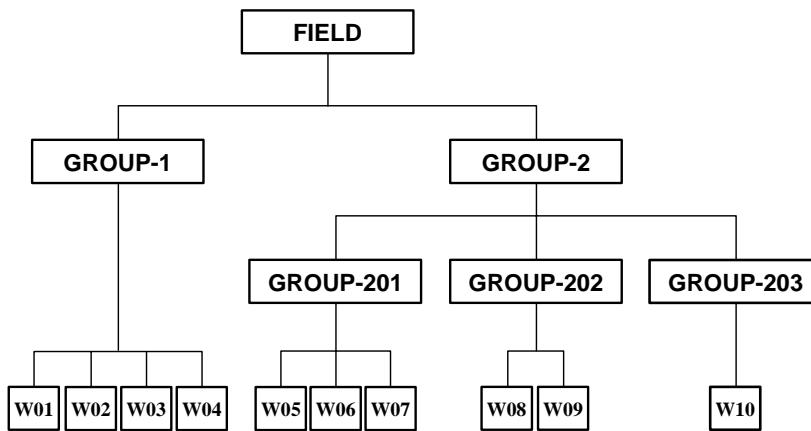
Optional keyword. Default is to use the instantaneous potential (\*IP) to determine rate distributions among wells to meet the group target rate.

#### 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. Priority ranking apportionment currently does not support the manifold groups.

#### 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 principle 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'      *TARGET   *STO  1000.
*GCONP  'GROUP-2'    *TARGET   *STO  600.
*APPOR-METHOD  *PROD  'FIELD'   *IP
*APPOR-METHOD  *PROD  'GROUP-2'  *PRIOR
*PRIOR-FORM   *PROD  'Group-2'
  *PRIOR-NUMER  0.    1.    0.    0.    0.    0.
  *PRIOR-DENOM  0.    0.    1.    0.    0.    0.
  
```

In case that the GROUP-2 target is to be apportioned using the guide rates method (\*GUIDE), the group rates for contributing groups GROUP-201, GROUP-202 and GROUP-203 and/or well rates for contributing wells W05 – W10 shall be supplied by \*GUIDEP. Guide rates supplied for non-contributing groups\wells are ignored. On the other hand, instantaneous potential (IP) is defaulted as the guide rate if it has not been supplied by the user for a contributing group or well that is under the guide rates control.

```

*APPOR-METHOD  *PROD  'GROUP-2'   *GUIDE
*GUIDEP   *STO   'GROUP-201'   'GROUP-202'   'GROUP-203'
               300.          200.         100.
  **and/or
*GUIDEP   *STO   'W05'    'W06'    'W07'    'W08'    'W09'    'W10'
               100.
  
```

### **Backward Compatibility Notes:**

The sub-keywords \*IPP under \*GCONP and \*IIP under \*GCONI in the previous versions, which set the \*IP apportionment flag globally for all the group production or injection targets, now become obsolete. They will be ignored with reminding message if \*APPOR-METHOD is encountered ahead in the 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
```

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
```

## 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 an injection target by which only the most “injectable” wells operate.

### FORMAT:

```
*PRIOR-FORM      (*PROD | *GASI | *WATI) group_names
                  (*PRIOR-RATE    (*MRC | *BHP (bhp_val)))
                  (*PRIOR-CTRL    freq tcr_min trc_max)
                  (*PRIOR-NUMER   A0 A1 ... Anph)
                  (*PRIOR-DENOM   B0 B1 ... Bnph)
```

### DEFINITIONS:

#### \*PROD

Indicates that group production targets should be apportioned with the specified method.

#### \*GASI

Indicates that group gas injection targets should be apportioned with the specified method.

#### \*WATI

Indicates that group water injection targets should be apportioned with the specified method.

#### group\_names

A list of group names to which the apportionment method is applied. No wildcard characters may be used in the group names.

#### \*PRIOR-RATE

This sub-keyword defines what type of rate is to be used in calculating the well priorities. \*MRC indicates that the well priorities are to be calculated using the stream rates under the most restrictive condition. \*BHP indicates that the well priorities are to be calculated using the production or injection potentials at a specified bottom hole pressure, ‘bhp\_val’. If ‘bhp\_value’ does not appear after \*BHP, the default value will be applied: 1 atmosphere for production targets and 3500 atmosphere for injection targets.

#### \*PRIOR-CTRL

This sub-keyword introduces the numerical parameters that control the priority ranking apportionment. ‘freq’ is the minimum time interval in days between well priority calculations. ‘tcr\_min’ is the lower limiting value of

the ratio between the pool target and its capacity, above which the priority ranking apportionment will be applied. ‘tcr\_max’ is the upper limiting value of the ratio between the pool target and its capacity, below which the priority ranking apportionment will be applied. \*PRIOR-CTRL is not necessary to and thus has no effect on the internal guide rate apportionment.

#### \*PRIOR-NUMER

This sub-keyword introduces the priority weighting coefficients for the numerator  $A_i$  ( $i = 0, nph$ ). ‘nph’ is the number of surface production or injection streams allowed for group targets.

#### \*PRIOR-DENOM

This sub-keyword introduces the priority weighting coefficients for the denominator  $B_i$  ( $i = 0, nph$ ).

#### DEFAULTS:

Conditional keyword. If any of the sub-keywords (\*PRIOR-CTRL, \*PRIOR-RATE, \*PRIOR-NUMER, \*PRIOR-DENOM) is absent after \*PRIOR-FORM, its default value(s) will be applied:

* PRIOR-RATE	*MRC		
* PRIOR-CTRL	0.0	0.0	1.0
* PRIOR-NUMER	1.0	0.0	0.0
* PRIOR-DENOM	1.0	0.0	0.0

Otherwise, all the required values by that sub-keyword must be filled in.

The default for the internal guide rate apportionment (\*INGUIDE) is to use the instantaneous potentials as the guide rates if \*PRIOR-FORM is not present.

#### CONDITIONS:

This keyword must be located in the WELL AND RECURRENT DATA keyword group after the group hierarchy is available. If a group target is apportioned with the priority ranking method set by \*APPOR-METHOD, a priority formula for that targeted group must be supplied by \*PRIOR-FORM. 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

$$\text{Priority(iw)} = \frac{A_0(\text{ig}) + \sum_{i=1}^{\text{nph}} A_i(\text{ig})Q_i(\text{iw})}{B_0(\text{ig}) + \sum_{i=1}^{\text{nph}} B_i(\text{ig})Q_i(\text{iw})}$$

where  $A_i$  and  $B_i$  ( $i = 0, \text{nph}$ ) are the weighting coefficients for the numerator and denominator, respectively, introduced by sub-keywords \*PRIOR-NUMER and \*PRIOR-DENOM. It is assumed that all wells contributing to a common group target should have the same priority definition. However, different targeted groups or different targeted injection streams may have different specifications of well priorities. Coefficients  $A_0$  and  $B_0$  are constants whilst the rest are the stream weighting coefficients to be multiplied correspondingly by  $Q_i$ , the stream rate under the most restrictive constraints (\*MRC) or the production/injection potential at the given bottom hole pressure (\*BHP) 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.

STARS (Production: nph = 5)						
i	CONST	STO	STG	STW	STL	BHF
NUMER	A0	A1	A2	A3	A4	A5
DENOM	B0	B1	B2	B3	B4	B5

STARS (Injection: nph = 2)			
i	CONST	STG	STW
NUMER	A0	A1	A2
DENOM	B0	B1	B2

The first value read by sub-keyword \*PRIOR-CTRL,  $freq$ , indicates the elapsed minimum time in days between priority calculations. The default is to update the well priority at every timestep ( $freq = 0.0$ ). It is designed to reduce in some cases the frequent alternating of the swing well among wells that may have very comparable priority values.

The last two values after sub-keyword \*PRIOR-CTRL,  $tcr\_min$  and  $tcr\_max$ , are the threshold values for the pool target-capacity ratio (between 0-1), within which the ranking apportionment will be applied. The instantaneous potential (IP) apportionment will otherwise be adopted if the said ratio is out of the range  $[tcr\_min, tcr\_max]$ . Default is  $[0, 1]$ , meaning that the priority ranking apportionment will always be in place regardless of the pool target-capacity ratio.

The upper limit  $tcr\_max$  is designed to avoid the constant violation (and therefore switching) of the most restrictive constraints that might happen to some wells if they were forced to produce or inject at their maximum rates ( $tcr\_max = 1$ ). Another consideration is that, if the pool has a very low capacity to meet a certain target (i.e. high pool target-capacity ratio), all contributing wells are required to operate at their maxima. The ranking apportionment may take the least effect because there is no swing well to choose.

The lower limit  $tcr\_min$ , on the other hand, provides a flexible control over the wells with lower priorities than the swing well. These wells can operate at the  $tcr\_min$  fraction of their maximum rates allowable instead of at the zero rate.

### Example

Supply the priority formula for targeted groups ‘Group-1’ and Group-2’. As invoked by \*APPOR-METHOD for the priority ranking method, wells will be opened in 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    STL    BHF
*PRIOR-NUMER  0.        1.        0.        0.        0.
*PRIOR-DENOM  0.        0.        1.        0.        0.
```

Wells can also be opened in order of decreasing WCUT :

```
*PRIOR-NUMER  0.        1.        0.        1.        0.        0.
*PRIOR-DENOM  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.        1.        0.        0.
*PRIOR-DENOM  0.        0.        0.01 1.        0.        0.
```

---

## Guide Rates for Groups or Wells

\*GUIDEP, \*GUIDEI

### PURPOSE:

\*GUIDEP specifies the guide rates as needed if the guide rate apportionment method is set by \*APPOR-METHOD \*GUIDE to distribute the production rates to groups or wells so as to meet the production target.

\*GUIDEI specifies the guide rates as needed if the guide rate apportionment method is set by \*APPOR-METHOD \*GUIDE to distribute the injection rates to groups or wells so as to meet the injection target.

### FORMAT:

*GUIDEP	*STO	(‘group_names’)	guide_rates
*GUIDEI	*STG	(‘well_names’)	guide_rates
	*STW		
	*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. Rates should be entered as (m<sup>3</sup>/day | STB/day). Not valid for use with \*GUIDEI.

#### \*STG

This subkeyword indicates that the gas stream is the reference stream to which the guide rates apply; i.e., that the guide rate values entered should be interpreted as gas rates or as being proportional to gas rates. Rates should be entered in (m<sup>3</sup>/day|SCF/day).

#### \*STW

This subkeyword indicates that the water stream is the reference stream to which the guide rates apply; i.e., that the guide rate values entered should be interpreted as water rates or as being proportional to water rates. Rates should be entered in (m<sup>3</sup>/day|STB/day).

#### \*STL

This subkeyword indicates that the guide rates apply to the total of the liquid stream rates (oil+water); i.e., that the guide rate values entered should be interpreted as total liquid rates or as being proportional to total liquid rates. Not valid for use with \*GUIDEI. Rates should be entered in (m<sup>3</sup>/day|STB/day).

#### \*BHF

Identifies that guide rate values are Bottom hole fluid rates. Rates should be entered in (m<sup>3</sup>/day|rbbl/day). Not valid for use with \*GUIDEI.

group_names	A list of 'group_names' to which the guide rates are applied.
well_names	One or more quoted well names to which the guide rates are applied. See <b>Wildcarding Well Names</b> at the beginning of this chapter. well_names cannot be mixed with group_names in the same list.
guide_rates	A list of guide rates for the list of groups or wells. See the entries for the stream designation strings for the proper units.

#### **DEFAULTS:**

Optional keywords. Default is to use the instantaneous injection / production potential (IIP or IPP) as the guide\_rates. \*GUIDEP and \*GUIDEI do not have an effect if the apportionment method is defined other than the guide rate method (\*GUIDE) by \*APPOR-METHOD.

#### **CONDITIONS:**

\* GUIDEP AND \*GUIDEI must be located in the WELL AND RECURRENT DATA keyword group after the wells or groups have been defined.

#### **EXPLANATION:**

Guide rates are set for each of the wells specified in 'well\_names' or each group specified by the 'group\_names'. If the guide rates apportionment method (\*APPOR-METHOD \*GUIDE) is being used then all producers or injectors connected to that group must have their guide\_rates specified with \*GUIDEP or GUIDEI. If no guide rates at all have been specified for the apportionment of the target of a particular group, then instantaneous injection/production potentials are used as the guide rates. Please consult the documentation for more information on the guide rate method under the keyword \*APPOR-METHOD.

When \*GUIDEI is used, the guide rates apply only for the target specified with the stream identifier; e.g., water injectors have their guide rates specified under \*GUIDEI \*STW.

When \*GUIDEP is used, the guide rates apply for the listed wells or groups to all target types, with the proper conversion made for the ratios at which the wells produce the different streams. That is, the target is apportioned so that the wells produce the reference stream set with the \*GUIDEP keyword in the ratios indicated by the rates entered with \*GUIDEP. For example, if the following lines are entered when SI units are used,

```
*GUIDEP *STO 'Well 1' 'Well 2' 'Well 3'
               100.      200.     300.
```

and an STG group target is to be apportioned among these producers, then if Well 1 has a GOR of 600., Well 2 a GOR of 300., and Well 3 a GOR of 200., then the STG guide rates used for the wells would be 60000, 60000, and 60000 respectively and the STG target would be distributed equally among the wells if no other constraints were violated in the process.

**Example:**

```
*GUIDEP *STO 'GR-PA' 'GR-PB' 'GR-PC'  
            300.00 100.00 400.00  
*GUIDEP *STO 'PA1' 'PA2' 'PA3'  
            100.0 100.0 100.0  
*GUIDEP *STL 'PB1' 'PB2' 'PB3'  
            100.0 100.0 100.0  
*GUIDEP *STG 'PC1' 'PC2'  
            100.0 200.0  
*GUIDEI *STW 'IA1' 'IB1' 'IC1'  
            100.0 200.0 100.0
```

---

## **Flag for Accompanying Groups or Wells Not Under Group Control (Optional)**

**\*GCPOFF, \*GCIOFF**

### **PURPOSE:**

\*GCPOFF specifies the accompanying groups or wells are not under group production constraint from higher level groups.

\*GCIOFF specifies the accompanying groups or wells are not under group injection constraint from higher level groups.

### **FORMAT:**

\*GCPOFF ('group\_names')  
('well\_names')

-or-

\*GCIOFF \*GAS ('group\_names')  
\*WATER ('well\_names')

### **DEFINITIONS:**

#### **\*GAS**

Indicates not under group control for gas injection calculations.

#### **\*WATER**

Indicates not under group control for water injection calculations.

#### **group\_names**

A list of 'group\_names' to which the group constraint from higher level should not be applied.

#### **well\_names**

One or more quoted well names to which the group constraint from higher level should not be applied. See **Wildcarding Well Names** at the beginning of this chapter. well\_names cannot be mixed with group\_names in the same list.

### **DEFAULTS:**

Optional keywords. Default is to apply the group constraint from higher level.

### **CONDITIONS:**

\*GCPOFF AND \*GCIOFF must be located in the Well and Recurrent Data keyword group, and must follow the \*GCONI, \*GCONP or \*GCONM keywords.

### **EXPLANATION:**

This keyword isolates the groups or wells so that they can produce or inject according to their own rate and pressure constraints, without being controlled by higher level constraints.

Example:

```
*GCPOFF 'GR-PA'   'GR-PB'   'GR-PC'  
*GCPOFF 'PA1'     'PB2'     'PC3'  
*GCIOFF *GAS      'GR-PA'  
*GCIOFF *WATER    'IB2'     'IC1'
```

---

## Well/Group On-time Fraction (Optional)

\*ON-TIME

### PURPOSE:

\*ON-TIME specifies the fraction of time during which a well or group operates.

### FORMAT:

\*ON-TIME      ( *well\_list | group\_list* )  
                  *otf\_list*

### DEFINITIONS:

*well\_list*

One or more quoted well names to specify the wells to which this on-time fraction data applies. See **Wildcarding Well Names** at the beginning of this chapter. 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.

*group\_list*

One or more quoted group names to specify the groups to which this on-time fraction data 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.

*otf\_list*

List of on-time fractions, one for each well in order in *well\_list* or group in order in *group\_list*. All values must appear on a new line immediately after the line containing \*ON-TIME. Repeat counts are allowed. 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, an on-time fraction of 1 are assigned to each well/group. Wells/groups never listed under \*ON-TIME retain the on-time fraction of 1 throughout the simulation.

### CONDITIONS:

A well or group must be defined before it can be assigned data via \*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.

The on-time fraction option is not available for a well attached to a discretized wellbore.

Well and group names may not appear together after the same \*ON-TIME keyword.

## **EXPLANATION:**

Wells and groups are treated in the same manner as 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 fraction 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 at a point in the hierarchy tree is the product of the OTF\_input factors on the path from that point to the field (top) level. If no group structure is specified then 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 ( $Q_{\text{instant}}$ ) for a well refers to the rate when the subject well is running on its full-flow conditions. 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 ( $Q_{\text{average}}$ ) for a well/group is the relative mean rate over the period while its parent group is operating. The actual rate ( $Q_{\text{actual}}$ ), 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  $\text{OTF\_input} = 1$ , since this well/group has no down-time ‘relative’ to its parent group. This does not necessarily mean that the actual rate equals to the instantaneous rate as well.

Theoretically, these three rates might have different values. In case that all group on-time fractions are unity but only well on-time fractions are not, one expects for all levels

$$\begin{aligned} \text{OTF\_actual} &= \text{OTF\_input} \\ Q_{\text{actual}} &= Q_{\text{average}} \end{aligned}$$

Moreover,  $Q_{\text{actual}} = Q_{\text{instant}}$ , for all groups (but not the wells). This is the setting 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  $< 1$ .

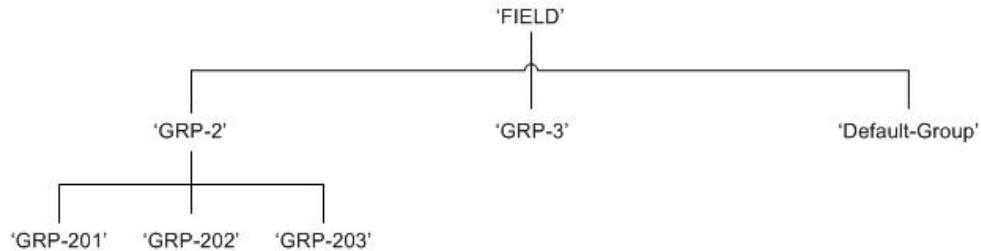
Any flow constraints, targets and limits specified for a well (e.g., \*OPERATE, \*MONITOR) or group (e.g. \*GCOMP, \*GCONI, \*GCONM) will apply to its instantaneous rate. For wells, reported values of the bottom-hole pressure, drawdown, well layer pressure, and well head pressures correspond to the specified (instantaneous) rates. 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 has the following group-well hierarchy:



The field is assigned an oil target (\*GCONP/\*TARGET) of 500 which is to be apportioned using the Guide Rate method (\*APPOR-METHOD/\*GUIDE) with supplied guide rates:

```
*GUIDEPE *STO  'GRP-2'  'GRP-3'  'GRP-201'  'GRP-202'  'GRP-203'
      3.        2.        1.        2.        3.
```

In a very general but unusual 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 (OTF\_input = 1), suggesting that they operate (100%) as long as their parent groups are operating.

In this example, if no well constraints have 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 would be:

	<b>On-Time Factor (OTF)</b>	<b>Rates (Q)</b>			
<b>Group Hierarchy</b>	<b>Input</b>	<b>Actual</b>	<b>Instant.</b>	<b>Average</b>	<b>Actual</b>
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 behavior of lower level groups (and wells) are concerned, the above data set would run exactly the same if the user were to 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 accumulations for all the wells/groups remain unchanged. However, since now the Field has a unity on-time fraction, that is, the instantaneous rate is indeed the actual rate for the field, the field target should be adjusted 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. Currently, the table input is only allowed for isothermal runs.

### FORMAT:

```
( *PTUBE1 | *ITUBE1 )      table_number  
  
*DEPTH refdepth  
Flo  
  flo(1) ... flo(nflo)  
Gfr  
  gfrl(1) ... gfrl(ngfr)  
Wfr  
  wfr(1) ... wfr(nwfr)  
Add  
  add(1) ... add(nadd)  
*WHP  
  whp(1) ... whp(nwhp)  
*BHP  
  
** iflo      igfr      iwfr      iadd      bhp(1)...      bhp(nwhp)  
   :         :         :         :         :         :
```

### DEFINITIONS:

#### \*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 (m<sup>3</sup>/day | bbl/day | cm<sup>3</sup>/min);
- \*LIQ      Liquid (oil+water) production rate (m<sup>3</sup>/day | bbl/day | cm<sup>3</sup>/min);
- \*GAS      Gas production rate (m<sup>3</sup>/day | ft<sup>3</sup>/day | cm<sup>3</sup>/min);
- \*TML      Total molar production rate (gmol/day).

For \*ITUBE1, it is identified by one of the following subkeywords:

- \*GAS      Gas injection rate ( $\text{m}^3/\text{day}$  |  $\text{ft}^3/\text{day}$  |  $\text{cm}^3/\text{min}$ );
- \*WAT      Water injection rate ( $\text{m}^3/\text{day}$  |  $\text{bbl}/\text{day}$  |  $\text{cm}^3/\text{min}$ );
- \*TML      Total molar injection rate (gmol/day).

#### Gfr

Independent Gas Fraction variable (\*PTUBE1 only), identified by one of the following subkeywords:

- \*GOR      Gas-oil ratio ( $(\text{m}^3/\text{m}^3)$  |  $\text{ft}^3/\text{bbl}$  |  $\text{cm}^3/\text{cm}^3$ );
- \*GLR      Gas-liquid ratio ( $\text{m}^3/\text{m}^3$  |  $\text{ft}^3/\text{bbl}$  |  $\text{cm}^3/\text{cm}^3$ );
- \*OGR      Oil-gas ratio ( $\text{m}^3/\text{m}^3$  |  $\text{bbl}/\text{ft}^3$  |  $\text{cm}^3/\text{cm}^3$ );
- \*AMW      Average molecular weight (gram/gmol).

#### Wfr

Independent Water Fraction variable (\*PTUBE1 only), identified by one of the following subkeywords:

- \*WOR      Water-oil ratio;
- \*WCUT     Water cut (water-liquid ratio);
- \*WGR      Water-gas ratio ( $\text{m}^3/\text{m}^3$  |  $\text{ft}^3/\text{bbl}$  |  $\text{cm}^3/\text{cm}^3$ );
- \*WMF      Water molar fraction.

#### Add

Independent Additional look-up variable (\*PTUBE1 only), identified by one of the following subkeywords:

- \*LFG      Lift gas injection rate ( $\text{m}^3/\text{day}$  |  $\text{ft}^3/\text{day}$  |  $\text{cm}^3/\text{min}$ );
- \*LFR      Injection gas-liquid ratio for gas lift ( $\text{m}^3/\text{m}^3$  |  $\text{ft}^3/\text{bbl}$  |  $\text{cm}^3/\text{cm}^3$ );
- \*LFT      Total gas-liquid ratio for gas lift ( $\text{m}^3/\text{m}^3$  |  $\text{ft}^3/\text{bbl}$  |  $\text{cm}^3/\text{cm}^3$ );
- \*DNO     Surface mass density of oil ( $\text{kg}/\text{m}^3$  |  $\text{lb}/\text{ft}^3$  |  $\text{g}/\text{cm}^3$ ). It takes the volumetric-weighted average value for a manifold group;
- \*DNG     Surface mass density of gas ( $\text{kg}/\text{m}^3$  |  $\text{lb}/\text{ft}^3$  |  $\text{g}/\text{cm}^3$ ). 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 | kPa).

**\*BHP**

This keyword introduces input of the bottom-hole (or manifold inlet) pressure table (kPa | psi | kPa).

**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.

**DEFUALTS:**

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 \*PHWELLBORE, \*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.

### 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
*GOR  **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

```

---

## Allow a Set of Keywords to be Processed When a Specified Condition (Trigger) is Satisfied (Optional)

\*TRIGGER

### PURPOSE:

\*TRIGGER allows the user to specify certain actions which are implemented when a specific condition or trigger is satisfied during the simulation.

### FORMAT:

```
*TRIGGER 'trig_name' trig_def *APPLY_TIMES napt *INCREMENT rinc  
*TEST_TIMES ntestt  
*TEST_AFTER_TIMER rtimedr *TEST_AFTER_TIMEA rtimedea  
{list of actions}  
*END_TRIGGER
```

Where **trig\_def** is one of

```
(*ON_WELL 'well_names' well_condition operator condition_value)  
-or-  
(*ON_GROUP 'group_names' group_condition operator condition_value)  
-or-  
(*ON_LAYER 'well_name' layer_UBA layer_condition operator condition_value)  
-or-  
(*ON_SECTOR 'sector_name' sector_condition operator condition_value)  
-or-  
(*ON_FIELD 'FIELD' field condition operator condition_value)  
-or-  
(*ON_ELAPSED 'TIME' time condition operator condition_value)
```

And where {list of actions} represents valid well and recurrent data keyword lines.

### DEFINITIONS:

#### trig\_name

Enter a string (in single quotes) of less than 40 characters to uniquely identify the trigger. The name must immediately follow the \*TRIGGER keyword. This token is required.

#### \*ON\_WELL

This subkeyword indicates that the test condition is to be applied to a well or list of wells (list, if a wildcard is used in the well name or a list of wells is specified). One of \*ON\_WELL or \*ON\_GROUP or \*ON\_SECTOR or \*ON\_LAYER or \*ON\_FIELD or \*ON\_ELAPSED is a required token immediately following the trigger name string.

### well\_names

Any number of well names (in quotes) to specify the wells to which this trigger applies. The well(s) must be previously defined. The well names must all be specified on a single line.

Note: wildcards may be used in the 'well\_names' string as follows:

\* replaces any number of characters at the end of a well name or can be used on its own to replace all wells (e.g. \*TRIGGER 'trig1' \*ON\_WELL '\*' or \*TRIGGER 'trig1' \*ON\_WELL 'wel\*').

? Replaces any single character anywhere in the well name (e.g. \*TRIGGER 'trig1' \*ON\_WELL '?ell1').

The two wild cards can be combined on any list and when wild cards are used the well list generated is printed out for the user to check.

**If \*ON\_WELL is specified, then at least one well name must immediately follow the \*ON\_WELL keyword.**

### Well\_condition:

Enter a single keyword identifying **one** of the following: a well stream rate or cumulative or composition or well bottom hole or tubing head pressure or backflow. The valid lists of conditions for a well are shown in Table 1 below: This is a required token and must immediately follow the well name or well list.

**Table 1: Well quantities:**

<b>Subkeyword</b>	<b>Meaning</b>
STO-RP	Stock Tank Oil - Rate of Production
STO-CP	Stock Tank Oil - Cumulative Production
STO-RI	Stock Tank Oil – Rate of Injection
STO-CI	Stock Tank Oil – Cumulative Injection
STW-RP	Stock Tank Water – Rate of Production
STW-CP	Stock Tank Water – Cumulative Production
STW-RI	Stock Tank Water – Rate of Injection
STW-CI	Stock Tank Water – Cumulative Injection
STG-RP	Stock Tank Gas – Rate of Production
STG-CP	Stock Tank Gas – Cumulative Production
STG-RI	Stock Tank Gas – Rate of Injection
STG-CI	Stock Tank Gas – Cumulative Injection
STL-RP	Stock Tank Liquid – Rate of Production
STL-CP	Stock Tank Liquid – Cumulative Production
BHF-RP	Bottom Hole Fluid – Rate of Production
BHF-CP	Bottom Hole Fluid – Cumulative Production
BHF-RI	Bottom Hole Fluid – Rate of Injection
BHF-CI	Bottom Hole Fluid – Cumulative Injection
STI-RP	Stock Tank Intermediate liquid – Rate of Production
STI-CP	Stock Tank Intermediate liquid – Cumulative Production

WTG-RP	Wet Gas – Rate of Production
WTG-CP	Wet Gas – Cumulative Production
BHP	Bottom Hole Pressure
WHP	Well Head Pressure
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
MXX	Mole percent of component XX in produced well stream hc
BKFLOW	Back flow (true if any layer is back flowing)
TEMP	Maximum temperature of all completions of a well
O2CONC	Oxygen component mole fraction – maximum of all completions of a well

#### \*ON\_GROUP

This subkeyword indicates that the test condition is to be applied to a group or list of groups (if a wildcard is used in the group name or a list of groups is specified). The group hierarchy must be previously defined to enable group based triggers to be used. One of \*ON\_WELL or \*ON\_GROUP or \*ON\_SECTOR or \*ON\_LAYER or \*ON\_FIELD or \*ON\_ELAPSED is a required token immediately following the trigger name string.

#### group\_names

Any number of group names (in quotes) to specify the groups to which this trigger applies. The groups(s) must be previously defined. There are no groups created by default. The names must all be specified on a single line.

Note: wildcards may be used in the 'group\_names' string as follows:

\* replaces any number of characters at the end of a group name or can be used on its own to replace all groups (e.g. \*TRIGGER 'trig1' \*ON\_GROUP '\*' or \*TRIGGER 'trig1' \*ON\_GROUP 'grp\*').

? Replaces any single character anywhere in the group name (e.g. \*TRIGGER 'trig1' \*ON\_GROUP '?rp1').

The two wild cards can be combined on any list and when wild cards are used the group list generated is printed out for the user to check.

At least one group name must immediately follow the \*ON\_GROUP keyword. Required token.

#### Group\_condition:

The valid list of quantities for groups and for the field is shown in Table 2 below:

**Table 2: Group and Field quantities:**

<b>Subkeyword</b>	<b>Meaning</b>
STO-RP	Stock Tank Oil – Rate of Production
STO-CP	Stock Tank Oil – Cumulative Production
STW-RP	Stock Tank Water – Rate of Production
STW-CP	Stock Tank Water – Cumulative Production
STW-RI	Stock Tank Water – Rate of Injection
STW-CI	Stock Tank Water – Cumulative Injection
STG-RP	Stock Tank Gas – Rate of Production
STG-CP	Stock Tank Gas – Cumulative Production
STG-RI	Stock Tank Gas – Rate of Injection
STG-CI	Stock Tank Gas – Cumulative Injection
STL-RP	Stock Tank Liquid – Rate of Production
STL-CP	Stock Tank Liquid – Cumulative Production
BHF-RP	Bottom Hole Fluid – Rate of Production
BHF-CP	Bottom Hole Fluid – Cumulative Production
BHF-RI	Bottom Hole Fluid – Rate of Injection
BHF-CI	Bottom Hole Fluid – Cumulative Injection
STI-RP	Stock Tank Intermediate liquid – Rate of Production
STI-CP	Stock Tank Intermediate liquid – Cumulative Production
WTG-RP	Wet Gas – Rate of Production
WTG-CP	Wet Gas – Cumulative Production
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
MPWS MXX	Mole percent of component XX in group production
GWGR	Gas-Wet Gas Ratio (production )
WWGR	Water-Wet Gas Ratio (production)
RECYSTG	Group Recycled Gas injection rate
RECYSTW	Group Recycled Water injection rate
VOIDRPG	Group voidage replacement ratio by gas injection.
VOIDRPW	Group voidage replacement ratio by water injection
VOIDRPT	Group voidage replacement ratio by all injection streams
STOR	Steam oil ratio (ratio of instantaneous steam injection / instantaneous oil production) for the group
STORC	Cumulative steam oil ratio (ratio of cumulative steam injected / cumulative oil production) for the group
OSTR	Oil Steam oil ratio (ratio of instantaneous oil production / instantaneous stream injection) for the group
OSTRC	Cumulative oil steam ratio (ratio of cumulative oil produced / cumulative steam injected) for the group

STOR2	Steam oil ratio (ratio of instantaneous steam injection / instantaneous oil production) for the group
STORC2	Cumulative steam oil ratio (ratio of cumulative steam injected / cumulative oil production) for the group
OSTR2	Oil Steam oil ratio (ratio of instantaneous oil production / instantaneous stream injection) for the group
OSTRC2	Cumulative oil steam ratio (ratio of cumulative oil produced / cumulative steam injected) for the group
STOR3	Steam oil ratio (ratio of instantaneous steam injection / instantaneous oil production) for the group
STORC3	Cumulative steam oil ratio (ratio of cumulative steam injected / cumulative oil production) for the group

**\*ON\_LAYER**

This subkeyword indicates that the test condition is to be applied to a well layer. The well must be fully defined previously for a trigger based on a layer condition to be specified.

**well\_name**

A single string (in quotes) of less than 40 characters representing a well name must immediately follow the \*ON\_LAYER keyword. The name identifies the well to which the layer belongs.

**Layer\_UBA**

To identify the layer, enter the layer user block address immediately following the well name. The user block address is specified in the following general format: i1 j1 k1 / i2 j2 k2 / ... please review the manual pages on the \*PERF keyword for more information.

Do NOT encapsulate the user block address in quotes.

**Layer\_condition:**

The valid list of quantities for layers is shown in table 3 below:

**Table 3 : Layer quantities:**

<b>Subkeyword</b>	<b>Meaning</b>
STO-R	Stock Tank Oil – Rate (sign tells whether Prod or Inj)
STW-R	Stock Tank Water – Rate
STG-R	Stock Tank Gas – Rate
STI-R	Stock Tank Intermediate Liquid – Rate
GOR	Gas-Oil Ratio (production)
WCUT	Water Cut (production)
WGR	Water Gas Ratio (production)
GLR	Gas Liquid Ratio (production)
BHP	Layer bottom-hole pressure
DWN	Layer drawdown – absolute value of block P – layer P

## **\*ON\_SECTOR**

This subkeyword indicates that the test condition is to be applied to a sector. GEM creates a default sector named ‘FIELD’ which includes all grid blocks and all wells. IMEX creates a default sector named (‘Entire Field’). Please note that there are 2 spaces between the word ‘Entire’ and ‘Field’.

Any other sector name used with a trigger must be previously defined.

### **sector\_name**

A single string representing a sector name (in quotes) of less than 16 characters to specify the sector to which this trigger applies. The sector must be previously defined. Only the ‘FIELD’ for GEM is created by default. The name must be on the same line as the \*TRIGGER keyword.

### **Sector\_condition:**

The valid list of quantities for sectors is shown in table 4 below:

**Table 4 : Sector quantities:**

<b>Subkeyword</b>	<b>Meaning</b>
PAVE	Pore-volume weighted pressure
SOAVE	Pore-volume weighted oil saturation
SWAVE	Pore-volume weighted water saturation
SGAVE	Pore-volume weighted gas saturation
STOIP	Stock tank oil in place
STWIP	Stock tank water in place
STFGIP	Stock tank free gas in place
STGIP	Stock tank gas in place, including free and dissolved

## **\*ON\_FIELD**

This subkeyword indicates that the test condition is based on a field level quantity. Note: The string ‘FIELD’ in single quotes must immediately follow the \*ON-FIELD keyword. The ON\_FIELD keyword on its own adequately identifies a trigger on the entire field. However to maintain consistency in format with other keywords such as ON\_WELL and ON\_GROUP, the field name string has been retained for ON\_FIELD as well. The user should enter the string ‘FIELD’ for the field name token even if the actual field level group or sector name assigned is different from ‘FIELD’. If a string other than ‘FIELD’ is entered following ON\_FIELD, a warning message will be generated and the simulation will proceed normally.

### **Field\_condition:**

The valid list of quantities for field is identical to that of groups as shown in table 2:

Operator:

The operator for the triggering condition must be one of:

‘<’ Less than

‘>’ Greater than

One of < or > (not in quotes) is required immediately following the specification of the test quantity using keywords such as \*STO-RP

condition\_value:

The value of the trigger condition. Enter a value based on the trigger condition and unit system selected for the simulation.

#### \*ON\_ELAPSED

This subkeyword indicates 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 (some time after 14.99 days – because the timestep sizes are not known apriori it is not possible to state exactly when this time will be). The inner ‘TRIG3’ will be defined when the outer trigger ‘TRIG2’ condition is satisfied, this can occur after at least  $10+4.99+5.99 = 20.98$  days have elapsed.

Note: The string ‘TIME’ in single quotes must immediately follow the \*ON-ELAPSED keyword. If a string other than ‘TIME’ is entered following ON\_ELAPSED, a warning message will be generated and the simulation will proceed normally.

#### Time\_condition:

Either the time elapsed from the start of the simulation can be entered or the time elapsed relative to the time when the trigger is first defined can be specified. To specify the absolute time use the sub keyword \*TIMSIM followed by a value. To specify elapsed time relative to the time when the trigger is parsed or defined use the sub keyword \*TRELTD followed by a value. The time should be in days for FIELD/SI/MODSI units and in minutes for laboratory (LAB) units.

#### TIMSIM:

Indicates that the time value entered is time elapsed from the start of the simulation, also referred to as absolute time.

#### TRELTD:

Indicates that the time value entered is relative to the time the trigger is defined, also referred to as relative time.

**Operator:**

The operator for the triggering condition must be one of:

‘<’ Less than

‘>’ Greater than

One of < or > (not in quotes) is required immediately following the specification of the test quantity using keywords such as \*TIMSIM

**condition\_value:**

The value of the trigger condition. Enter a value based on the time elapsed condition selected in units of days for field/SI units.

**\*APPLY\_TIMES:**

Subkeyword used to specify the maximum number of times that the actions specified with the trigger can be taken. An integer number must immediately follow this subkeyword. This subkeyword is optional.

**napt:**

Enter a single integer representing the maximum number of times that the specified list of actions can be executed. If no value is entered, then the default is 1. With the default of 1 the trigger condition is tested at the end of every timestep. As soon as the trigger condition is satisfied the list of actions is implemented and the trigger is removed from the list of active triggers. If more than 1 (say “n” times) is selected then the trigger remains active until the trigger condition is satisfied (“n”) times.

**\*INCREMENT:**

Subkeyword used to specify the increment to the trigger value. A single real number must follow this subkeyword. An integer number must immediately follow this subkeyword. This subkeyword is optional.

**Trigger Increment:**

Enter a single real value representing an increment to be applied to the trigger value, each time the trigger condition is satisfied. The trigger increment can be a negative number. The trigger increment does not need to be entered. The trigger increment can only be entered if the preceding number for the repetition times is also entered. The trigger increment is used only if a value greater than 1 is specified for the number of times the trigger condition can be satisfied. Once a trigger condition is satisfied the trigger value is recalculated as  $\text{value\_new} = \text{existing\_value} + \text{increment}$ . The new value is then used in testing the trigger condition for subsequent times until the trigger condition is once again satisfied.

**\*TEST\_TIMES:**

Subkeyword used to specify the maximum number of times that the trigger can be tested to ascertain if the trigger condition is satisfied. A single integer number must follow this subkeyword. This subkeyword is optional.

**ntestt:**

Enter a single integer representing the maximum number of times that the trigger can be tested to ascertain if the trigger condition is satisfied. If no value is entered, then the default is to test the trigger every timestep. If a value of 1 is entered then the trigger condition is tested only once at the end of the timestep during which the trigger is defined. The trigger is then removed from the active trigger list whether or not the trigger condition itself is satisfied. If more than 1 (say “n” times) is selected then the trigger condition is tested for “n” timesteps after the trigger is defined.

**\*TEST\_AFTER\_TIMER:**

Subkeyword used to specify the time delay which must elapse before the trigger condition will be tested. This time delay is relative to the time that the trigger comes into existence or is defined. A single real number must follow this subkeyword. This subkeyword is optional.

**rtimedr:**

Enter a single real number representing the delay in time (in days) after the time that the trigger is defined or parsed that the user wishes the trigger condition to be tested. If no value is entered, then the default is to assume a time delay of zero.

**\*TEST\_AFTER\_TIMEA:**

Subkeyword used to specify the time delay in days which must elapse before the trigger condition will begin to be tested. This time delay is relative to the start of the simulation or absolute time elapsed. A single real number must follow this subkeyword. This subkeyword is optional.

**rtimeda:**

Enter a single real number representing the delay in days after the start of the simulation that the user wishes to elapse before the trigger condition is tested going forward. If no value is entered, then the default is to assume a time delay of zero.

**{action\_list}**

The list of actions in the form of valid well and recurrent data keyword lines may be specified following the \*TRIGGER keyword and its subkeywords. The action list must start on a new line. The action list is optional. Following the action lines specify the keyword \*END\_TRIGGER on a new line to signal the end of the trigger definition. It is okay to not specify any action lines within a trigger.

**\*END\_TRIGGER**

This keyword marks the end of the list of actions or keyword lines associated with a given trigger. It must be on a new line. The \*TRIGGER and \*END\_TRIGGER must be used as a pair. For each \*TRIGGER keyword, an \*END\_TRIGGER keyword is required.

**\*STO-RP**

Oil production rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

**\*STO-R**

Oil phase rate at surface conditions for a layer. A positive number implies fluid is flowing in a normal direction. Specifically flow is expected to be from reservoir to well bore for a layer belonging to a producer and from well bore to reservoir if the layer belongs to an injection well. On the other hand if a negative number is entered, then the user is testing for a back flowing layer. The magnitude of the number indicated the severity of the back flow. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

Example 1:

```
*TRIGGER 'trig1' *ON_LAYER 'well1' 1 1 1 / 2 2 2 *STO-R > 300.0
```

If well1 is an injector, then the condition will be satisfied if flow for the layer is from the well to the reservoir (injection) and the flow rate is greater than 300. However if well1 is a producer, then the condition will be satisfied if flow is from reservoir to well bore (production) and the oil flow rate is greater than 300.0.

Example 2:

```
*TRIGGER 'trig2' *ON_LAYER 'well1' 1 1 1 / 2 2 2 *STO-R > -300.0
```

The condition will be satisfied if the layer is back flowing, for an injector flow is from the reservoir to the 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 for a layer. 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. A negative number implies a back flowing layer. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

**\*STG-R**

Gas phase rate at surface conditions for a layer. A positive number entered for a layer belonging to an injection well implies flow is expected to be from 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. A negative number implies back flowing layer. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

**\*STI-R**

Intermediate stream rate at surface conditions for a layer. A positive number entered for a layer belonging to an injection well implies flow is expected to be from 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. A negative number implies back flowing layer. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

**\*STO-CP**

Oil cumulative production at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in STB.

**\*STW-RP**

Water production rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

**\*STW-CP**

Water cumulative production at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in STB.

**\*STW-RI**

Water injection rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.

**\*STW-CI**

Water cumulative injection at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in STB.

**\*STG-RP**

Gas production rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in scf/D.

**\*STG-CP**

Gas cumulative production at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in scf.

- \*STG\_RI**  
Gas injection rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in scf/D.
- \*STG-CI**  
Gas cumulative injection at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in scf.
- \*STI-RP**  
Intermediate liquid stream production rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.
- \*STI-CP**  
Intermediate liquid stream cumulative production at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in STB.
- \*WTG\_RP**  
Wet gas stream production rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in scf/D.
- \*WTG-CP**  
Wet gas stream cumulative production at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in scf.
- \*STL\_RP**  
Liquid (oil + water) stream production rate at surface conditions. For SI units enter a value in m<sup>3</sup>/D and for field units in STB/D.
- \*STL\_CP**  
Liquid (oil + water) stream cumulative production at surface conditions. For SI units enter a value in m<sup>3</sup> and for field units in STB.
- \*BHF\_RP**  
The oil plus water plus gas phase production rate at bottom hole or reservoir conditions. For SI units enter a value in reservoir m<sup>3</sup>/D and for field units in reservoir barrels BBL/D.
- \*BHF\_CP**  
The oil plus water plus gas phase production cumulative at bottom hole or reservoir conditions. For SI units enter a value in reservoir m<sup>3</sup> and for field units in reservoir barrels BBL.
- \*BHF-RI**  
The oil plus water plus gas injection rate at bottom hole or reservoir conditions. For SI units enter a value in reservoir m<sup>3</sup>/D and for field units in reservoir barrels BBL/D.

**\*BHF-CI**

The oil plus water plus gas phase injection cumulative at bottom hole or reservoir conditions. For SI units enter a value in reservoir m<sup>3</sup> and for field units in reservoir barrels BBL.

**\*GOR**

Gas oil ratio at surface conditions. For SI units enter a value as surface m<sup>3</sup>/D of gas production per m<sup>3</sup>/D of surface oil production and for field units in scf of gas per STB of oil.

**\*WCUT**

Water cut at surface conditions. The water cut is the ratio of surface production of water divided by the total surface liquid or water + oil production. For SI units enter a value as surface water production rate (m<sup>3</sup>/D) divided by the sum of surface production rates of water and oil (m<sup>3</sup>/D) and field units in STB/D of water per STB/D of water + oil production.

**\*WGR**

Ratio of water production rate at surface conditions divided by the surface gas production rate. For SI units enter a value as surface water production rate (m<sup>3</sup>/D) divided by the surface production rates of gas (m<sup>3</sup>/D) and field units in STB/D of water per scf/D of gas production.

**\*GLR**

Ratio of gas production rate at surface conditions divided by the surface liquid (sum of water + oil phases) production rate. For SI units enter a value as gas production rate (m<sup>3</sup>/D) divided by the sum of surface production rates of water and oil (m<sup>3</sup>/D) and field units in scf/D of gas per STB/D of water + oil production.

**\*MPWS \*Mxx**

Mole percent of component “m” in the well stream. Mole percent is calculated as the molar rate of component “m” flowing into the well divided by the total molar hydrocarbon flow rate multiplied by 100. The total rate is the sum of all hydrocarbon components (does not include water). Enter the component number after the letter M. For example for component number 5 in the components list (as determined by the order that components are specified with \*COMPNAME keyword) use \*MPWS \*M5 <value>.

**\*MPVS \*Mxx**

Mole percent of component “m” in the surface gas stream. Mole percent is calculated as the molar rate of component “m” in the separator gas stream divided by the total molar hydrocarbon flow rate of the separator gas stream multiplied by 100. The total rate is the sum of all hydrocarbon components (does not include water). For example for component number 5 in the components list (as determined by the order that components are specified with \*COMPNAME keyword) use \*MPVS \*M5 <value>.

**\*MPLS \*Mxx**

Mole percent of component “m” in the separator “liquid” stream. If there is no intermediate liquid stream, then the liquid stream equals the surface oil stream, otherwise the “liquid” stream includes the oil and the intermediate liquid streams. Mole percent is calculated as the molar rate of component “m” in the separator liquid stream divided by the total molar flow rate of the separator liquid stream multiplied by 100. The total rate is the sum of all hydrocarbon components (does not include water). Enter the component number after the letter M. For example for component number 5 in the components list (as determined by the order that components are specified with \*COMPNAME keyword) use \*MPLS \*M5 <value>.

**\*BHP**

Bottom hole pressure of the well. Enter a value in kPa for SI units and psi for field units.

**\*WHP**

Tubing head pressure of the well. Enter a value in kPa for SI units and psi for field units.

**\*GWGR**

Ratio of gas production rate at surface conditions divided by the wet gas production rate at surface conditions. For SI units enter a value as surface gas production rate (m<sup>3</sup>/D) divided by the surface production rate of wet gas (m<sup>3</sup>/D) and field units in scf/D of gas per scf/D of wet gas production.

**\*WWGR**

Ratio of water production rate at surface conditions divided by the wet gas production rate at surface conditions. For SI units enter a value as water production rate (m<sup>3</sup>/D) divided by the wet gas surface production rate (m<sup>3</sup>/D) and field units in STB/D of water per scf/D of wet gas production.

**\*BKFLOW**

If any layer of the well is back flowing then this condition is satisfied. Note: To specify the case where action(s) are to be taken if the well IS backflowing use the \*BKFLOW keyword followed by the greater than operator and a value of 0. Alternatively to specify the case where the action(s) are to be taken if the well IS NOT back flowing use the \*BKFLOW keyword followed by the less than operator and a value of 0. If an increment is specified with the \*INCREMENT keyword, then that value will be ignored.

\*BKFLOW > 0 take actions if the well IS BACK FLOWING

\*BKFLOW < 0 take actions if the well IS NOT BACK FLOWING

\*TEMP

Maximum temperature of all completions (layers of a well). For SI units enter a value in degrees Centigrade and for field units enter a value in degrees Fahrenheit.

\*O2CONC

Maximum oxygen mole fraction of all completions of a well. Oxygen must be declared as a component in the gas phase for this trigger to be used.

\*GLR

Ratio of gas production rate at surface conditions divided by the surface liquid (sum of water + oil phases) production rate. For SI units enter a value as gas production rate (m<sup>3</sup>/D) divided by the sum of surface production rates of water and oil (m<sup>3</sup>/D) and in field units in scf/D of gas per STB/D of water plus oil production.

\*RECYSTG

Minimum group gas recycling rate. Group injection rate is calculated by summing the injection rates at surface conditions of all cycling gas injectors that belong to the group. For SI units enter a value in m<sup>3</sup>/D and in field units in scf/D.

\*RECYSTW

Minimum group water recycling rate. Group injection rate is calculated by summing the injection rates at surface conditions of all cycling water injectors that belong to the group. For SI units enter a value in m<sup>3</sup>/D and in field units in STB/D.

\*STOR

Steam oil ratio at any given time (instantaneous). For a specific group specified, the steam oil ratio is calculated by adding the steam injection rate from all steam injectors attached to the group and then dividing by the total oil production rate from all producers attached to the group. The steam injection rate is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of water/ stb of oil and is a dimensionless number. The group oil production is taken to be the maximum of actual oil production and 1.0e-20. Therefore when the oil production is close to zero the calculated steam oil ratio will be a very large number.

\*STORC

Cumulative steam oil ratio. For a specific group specified, the cumulative steam oil ratio is calculated by adding the total steam injected to date from all steam injectors attached to the group and then dividing by the total oil produced to date from all producers attached to the group. The steam injected volume is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production volume likewise is based on volume

at surface conditions. Therefore the ratio is stb of water/ stb of oil and is a dimensionless number. The group cumulative oil production is taken to be the maximum of actual cumulative oil production and 1.0e-20. Therefore when the oil cumulative production is close to zero the calculated cumulative steam oil ratio will be a very large number.

\*OSTR

Oil steam ratio at any given time (instantaneous). For a specific group specified, the oil steam ratio is calculated by adding the oil production rate from all producers and then dividing by the total steam injection rate for all steam/water injectors attached to the group. The steam injection rate is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of oil / stb of water and is a dimensionless number. The group steam injection rate is taken to be the maximum of actual injection rate and 1.0e-20. Therefore when the steam injection rate is close to zero the calculated oil steam ratio will be a very large number.

\*OSTRC

Cumulative oil steam ratio at any given time (instantaneous). For a specific group specified, the cumulative oil steam ratio is calculated by adding the oil production from all producers to date and then dividing by the total steam injection do date from all steam/water injectors attached to the group. The steam injected is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of oil / stb of water and is a dimensionless number. The group steam cumulative volume injected is taken to be the maximum of actual volume injected and 1.0e-20. Therefore when the steam injected is close to zero the calculated oil steam ratio will be a very large number.

\*STOR2

Steam oil ratio at any given time (instantaneous). For a specific group specified, the steam oil ratio is calculated by adding the steam injection rate from all steam injectors attached to the group and then dividing by the total oil production rate from all producers attached to the group. The steam injection rate is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of water/ stb of oil and is a dimensionless number. The trigger condition test based on SOR will only be performed if both group oil production rate and group water injection rate values are above 1.0e-20. Otherwise the trigger condition test will not be done and the trigger condition will not be satisfied.

**\*STORC2**

Cumulative steam oil ratio. For a specific group specified, the cumulative steam oil ratio is calculated by adding the total steam injected to date from all steam injectors attached to the group and then dividing by the total oil produced to date from all producers attached to the group. The steam injected volume is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production volume likewise is based on volume at surface conditions. Therefore the ratio is stb of water/ stb of oil and is a dimensionless number. The trigger condition test based on cumulative SOR will only be performed if both group oil production cumulative and group water injection cumulative values are above 1.0e-20. Otherwise the trigger condition test will not be done and the trigger condition will not be satisfied.

**\*OSTR2**

Oil steam ratio at any given time (instantaneous). For a specific group specified, the oil steam ratio is calculated by adding the oil production rate from all producers and then dividing by the total steam injection rate for all steam/water injectors attached to the group. The steam injection rate is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of oil / stb of water and is a dimensionless number. The trigger condition test based on SOR will only be performed if both group oil production rate and group water injection rate values are above 1.0e-20. Otherwise the trigger condition test will not be done and the trigger condition will not be satisfied.

**\*OSTRC2**

Cumulative oil steam ratio at any given time (instantaneous). For a specific group specified, the cumulative oil steam ratio is calculated by adding the oil production from all producers to date and then dividing by the total steam injection do date from all steam/water injectors attached to the group. The steam injected is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of oil / stb of water and is a dimensionless number. The trigger condition test based on cumulative SOR will only be performed if both group oil production cumulative and group water injection cumulative values are above 1.0e-20. Otherwise the trigger condition test will not be done and the trigger condition will not be satisfied.

**\*STOR3**

Steam oil ratio at any given time (instantaneous). For a specific group specified, the steam oil ratio is calculated by adding the steam injection rate from all steam injectors attached to the group and then dividing by the total oil production rate from all producers attached to the group. The steam injection rate is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production likewise is based on volume at surface conditions. Therefore the ratio is stb of water/ stb of oil and is a dimensionless number. The group oil production is taken to be the maximum of actual oil production and 1.0e-20. Therefore when the oil production is close to zero the calculated steam oil ratio will be a very large number. If operator is < then at some point actual SOR has to be greater than SOR specified as the trigger condition for the trigger condition to be satisfied. This additional criterion is designed to capture cases where the trigger condition can only be satisfied if the SOR has actually exceeded the specified value earlier. That is the SOR versus time graph shows a maximum and the test condition is on the decline portion of the curve following the maximum. If operator is > then at some point actual SOR has to be less than SOR specified as the trigger condition for the trigger condition to be satisfied. That is the SOR versus time graph shows a minimum and the test condition is on the ascending portion of the curve following the minimum.

**\*STORC3**

Cumulative steam oil ratio. For a specific group specified, the cumulative steam oil ratio is calculated by adding the total steam injected to date from all steam injectors attached to the group and then dividing by the total oil produced to date from all producers attached to the group. The steam injected volume is based on surface conditions or on CWE, that is cold water equivalent volume. The oil production volume likewise is based on volume at surface conditions. Therefore the ratio is stb of water/ stb of oil and is a dimensionless number. The group cumulative oil production is taken to be the maximum of actual cumulative oil production and 1.0e-20. Therefore when the oil cumulative production is close to zero the calculated cumulative steam oil ratio will be a very large number. If operator is < then at some point actual cumulative SOR has to be greater than cumulative SOR specified as the trigger condition for the trigger condition to be satisfied. This additional criterion is designed to capture cases where the trigger condition can only be satisfied if the SOR has actually exceeded the specified value earlier. That is the SOR versus time graph shows a maximum and the test condition is on the decline portion of the curve following the maximum. If operator is > then at some point actual cumulative SOR has to be less than cumulative SOR specified as the trigger condition for the trigger condition to be satisfied. That is the SOR versus time graph shows a minimum and the test condition is on the ascending portion of the curve following the minimum.

*DWN	Difference in absolute value well layer pressure minus the pressure of well block where well layer is completed. Enter a value in kPa for SI units and psi for field units.
*PAVE	Sector average pressure calculated on the basis of total pore volume. For SI units enter a value in kPa and for field units in psi.
*SOAVE	Sector average oil saturation calculated on the basis of total pore volume.
*SWAVE	Sector average water saturation calculated on the basis of total pore volume.
*SGAVE	Sector average gas saturation calculated on the basis of total pore volume.
*STOIP	Sector oil in place at standard conditions. For SI units enter a value in m <sup>3</sup> and in field units in STB.
*STWIP	Sector water in place at standard conditions. For SI units enter a value in m <sup>3</sup> and in field units in STB.
*STFGIP	Sector gas in place based on reservoir gas phase alone at standard conditions. For SI units enter a value in m <sup>3</sup> and in field units in scf.
*STGIP	Sector gas in place based on both reservoir gas and reservoir oil phase at standard conditions. For SI units enter a value in m <sup>3</sup> and in field units in scf.
*TIMSIM	Condition based on time elapsed since the beginning of the simulation in days.
*TRELTD	Condition based on time elapsed relative to the time when the trigger is defined, in days.

#### **DEFAULTS:**

\*TRIGGER is an optional keyword.

## **CONDITIONS:**

This keyword must be located in the WELL AND RECURRENT DATA keyword group.

## **EXPLANATION:**

This keyword allows for certain actions to be implemented when a specified condition or trigger is satisfied. The actions are specified in the form of a block of valid keywords encapsulated within the \*TRIGGER and \*END\_TRIGGER keywords. With the exception of the following keywords: \*DATE, \*TIME, \*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 GEM input data file can be used with triggers.

The trigger condition of each active trigger is tested at the bottom (end) of the timestep (that is after the timestep is considered to be completed, the cumulatives have been updated and the simulation is ready to proceed to the next timestep). If the trigger condition is satisfied then the list of associated actions are processed at the top (beginning) of the next timestep. The order of the triggers in the trigger list depends entirely on the sequence on which the triggers are defined. The triggers are not sorted in any other way. If nesting is used (that is a trigger is defined within another trigger) then the inner trigger is defined or becomes active only if the outer trigger condition is satisfied – such nesting may impact on the order that triggers appear in the trigger list.

Once a trigger condition is met, the count of the number of times remaining that the trigger can be satisfied is reduced by 1. If for example the default value of 1 is used for the maximum number of times the trigger can be satisfied, then the count remaining is 0 and the trigger is removed from the list of active triggers. The trigger is also removed from the active trigger list once the trigger condition has been tested for the maximum number of times as specified with the \*TEST\_TIMES keyword.

If a well change time (as specified with \*DATE or \*TIME card) coincides with the time that the trigger actions are to be processed, then the trigger actions are implemented FIRST followed by the list of actions or keywords following the \*DATE or \*TIME card.

It is possible to nest triggers by including the definition of the ‘inner’ trigger within the keyword set of the ‘outer’ trigger (that is before specification of the \*END\_TRIGGER keyword of the ‘outer’ trigger). This offers great flexibility, however such nesting should be properly thought out to avoid unintended consequences. For example it is possible to define a trigger ‘recursively’ using nesting. That is the inner and outer triggers can be assigned the same name. In this case when the outer trigger condition is first satisfied, the trigger redefines itself based on the inner trigger definition. For example consider the following block of keywords in the well and recurrent data section of a GEM data file

```

*TIME 0.0
...
...
...
*WELL 'well1'
    *OPERATE *STO 750.0
    *OPERATE *MIN *BHP 3000.0
*WELL 'well2'
    *OPERATE *STO 250.0
    *OPERATE *MIN *BHP 3000.0
*WELL 'well3'
    *OPERATE **STO 250.0
    *OPERATE *MIN *BHP 3000.0

*SHUTIN 'well2' 'well3'
*TIME 50.0  **Time when Trigger is defined
*TRIGGER 'trig1' *ON_WELL 'well1' *STO_RP < 500.0 *APPLY_TIMES 3
    *INCREMENT -25.0      ** outer trigger
        *WTMULT '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 stops. Suppose at time equal to 65 days the oil production rate for well 'well1' drops to below 500 STB/D. The trigger condition is satisfied, well 2 is opened with an initial oil rate target of 250 \*1.25 or 312.50. However because the inner trigger has the same name as the outer trigger, the trigger 'trig1' is also redefined at this time according to the inner definition. The maximum repeat count of 3 as well as the increment to the trigger condition of -25 STB/D specified with the outer trigger definition is redundant and can never be used. After 65 days the new trigger condition is based on GOR and if satisfied will open well 3 with an initial target of 250 STB/D.

The list of actions (keywords) appearing within a trigger definition are not validated in any way (that is checked for syntax or consistency with previous well and recurrent data etc) until the trigger condition is satisfied. In \*CHECKONLY mode however each line of the well and recurrent data including keywords (actions) within the trigger definition are processed.

With the implementation of trigger actions GEM restart runs are now handled differently. With previous versions of GEM, well management information required for a restart run was synthesized from both data saved to a restart file and from parsing the recurrent data to the last date/ time card prior to restart time. With the new version of GEM all information required for restart is written and read from the restart file. No recurrent data is read prior to restart time. Therefore whereas it was previously possible to have certain keywords (actions) take effect immediately upon restart by changing the recurrent data prior to the restart time, this is no longer possible.

The '@' symbol in quotes may be used as a place holder for a list of wells or groups that have satisfied the trigger condition. This list will be a subset of the list of wells or groups specified as part of the trigger definition. For example consider the following trigger:

```
*TIME 50.0
*TRIGGER 'trig1' *ON_WELL 'pr*' *GOR > 5000
    *SHUTIN '@'
*END TRIGGER
```

Suppose production wells named pr1, pr2, pr3, pr4 that match the wild card well list pr\* exist prior to the time that trigger 'trig1' is defined, that is at 50 days. Then when 50 days of simulation time has elapsed the GOR of each well pr1, pr2, pr3 and pr4 respectively will be tested against the trigger value of 5000. Suppose that at some subsequent time, say 70 days, wells pr1 and pr3 have GOR > 5000. In that case the trigger condition is satisfied and wells pr1 and pr3 will be shut in as directed, that is the \*SHUTIN '@' statement is equivalent to the user having specified \*SHUTIN 'pr1' 'pr3'. \*\*NOTE: When the '@' place holder is used ALL keyword(s) that PRECEDE the place holder must on the same line as the place holder.

Therefore for example

```
*TARGET *STO '@'
1.5
    is valid, but
*TARGET *STO
'@'
1.5
or
*TARGET
*STO '@'
1.5
```

are not, since in the latter cases TARGET and STO and '@' are not on the same line. Because of the special significance of the '@' symbol, please do not use this symbol as part of a well, group, sector or trigger name.

---

## Alter Well Constraint Value via a Multiplier (Optional)

**\*WTMULT**

### PURPOSE:

\*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.

### FORMAT:

*WTMULT	(*STO) (*STW) (*STG) (*STL) (*STI) (*BHP) (*WHP) (*DWN) (*DWA) (*DWB) (*BHF) (*BHG) (*STR)	(well_numbers) OR ('well_names') OR ‘@’ place holder	Multiplier(s)
---------	--	---	---------------

### DEFINITIONS:

#### \*STO

This subkeyword identifies a surface oil rate (m3/day | STB/day) constraint.

#### \*STW

This subkeyword identifies a surface water rate (m3/day | STB/day) constraint.

#### \*STG

This subkeyword identifies a surface gas rate (m3/day | scf/day) constraint.

#### \*STL

This subkeyword identifies a total surface liquid rate (oil+water) (m3/day | STB/day) constraint.

#### \*STI

This subkeyword identifies a surface intermediate liquid rate (m3/day | STB/day) constraint. Please see the manual entry for the \*WELSEP keyword for more information about the intermediate liquid stream.

#### \*BHF

This subkeyword identifies a total reservoir fluid rate (oil + water + gas + solvent) (m3/day | STB/day) constraint. This constraint is applicable to producers only.

**\*BHW**

This subkeyword identifies a reservoir water rate (m<sup>3</sup>/day | STB/day) constraint.

**\*BHP**

This subkeyword identifies a bottom hole pressure (kPa | psi) operating constraint.

**\*WHP**

This subkeyword identifies a well-head pressure (kPa | psi) constraint.

**\*DWN, \*DWA, \*DWB**

This subkeyword identifies a draw-down pressure (kPa | psi) constraint.

**\*BHG**

This subkeyword identifies a reservoir gas rate (m<sup>3</sup>/day | ft<sup>3</sup>/day) constraint.  
This keyword is applicable to injectors only.

**\*STR**

This subkeyword identifies a volume or molar percentage recycling rate constraint. This constraint is applicable to cycling injectors only.

**well\_numbers**

Any number of integers, or a range of integers to specify the well numbers to which this target applies. These well numbers must be on the same line as the \*WTMULT keyword. If more than one line is required to specify all the wells, then the \*WTMULT keyword must be repeated.

**well\_names**

Any number of 'well\_names' in quotes to specify the wells to which this target applies. These names must be on the same line as the \*WTMULT keyword. If more than one line is required for the well list, then the \*WTMULT keyword must be repeated. Limited wild-carding is available for this list; please see the explanation on the manual page for the \*SHUTIN keyword.

**'@'**

When the '@' symbol in quotes is specified in place of well numbers or well list, then it is considered to be a place holder for the list of wells that have satisfied the trigger condition. This list is necessarily a subset of the well list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. All keywords that precede the '@' place holder must appear on the same line as the '@' place holder. Therefore WTMULT and a keyword specifying a stream, such as STO must be on the same line as the '@' place holder. See the EXPLANATION section of \*TRIGGER keyword for examples of the use of @ place holder.

## Multiplier(s)

One number for each well identified by well\_numbers or 'well\_names' specifying a multiplying factor to be applied. The new value of the specified constraint = old value of the constraint \* the multiplier. Multipliers must appear on one or more new lines immediately following the \*WTMULT line. Multipliers must NOT appear on the same line as the list of well names or well numbers. A single multiplier can be applied to all wells in a well list.

### DEFAULTS:

Optional keywords. No defaults.

### CONDITIONS:

\*WTMULT 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.

\*WTMULT 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 \*WTMULT 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 is specified by using one of \*STO, \*STG, \*STW, \*STL, \*STI, \*BHP, \*WHP, \*DWN, \*DWA, \*DWB, \*BHF, \*BHW, \*BHG, or \*STR. One of these qualifiers must be present.

The \*WTMULT keyword should NOT be used to define a new operating constraint. The multiplier specified with the \*WTMULT keyword will ONLY be applied if the constraint of the type specified with \*WTMULT already exists for the well based on earlier input using the \*OPERATE keyword.

Examples:

```
*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
```

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

```

---

## Alter Well Constraint Value via an Increment (Optional)

\*WTINCR

### PURPOSE:

\*WTINCR allows modification of any previously specified well constraint value for well(s) listed by well\_numbers or 'well\_names' by applying an increment.

### FORMAT:

*WTINCR	(*STO) (*STW) (*STG) (*STL) (*STI) (*BHP) (*WHP) (*DWN) (*DWA) (*DWB) (*BHF) (*BHG)	(well_numbers) OR ('well_names') OR ‘@’ place holder	Increment(s)
---------	--	---	--------------

### DEFINITIONS:

#### \*STO

This subkeyword identifies a surface oil rate (m3/day | STB/day) constraint.

#### \*STW

This subkeyword identifies a surface water rate (m3/day | STB/day) constraint.

#### \*STG

This subkeyword identifies a surface gas rate (m3/day | scf/day) constraint.

#### \*STL

This subkeyword identifies a total surface liquid rate (oil+water) (m3/day | STB/day) constraint.

#### \*STI

This subkeyword identifies a surface intermediate liquid rate (m3/day | STB/day) constraint. Please see the manual entry for the \*WELSEP keyword for more information about the intermediate liquid stream.

#### \*BHF

This subkeyword identifies a total reservoir fluid rate (oil + water + gas + solvent) (m3/day | STB/day) constraint. This constraint is applicable to producers only.

**\*BHW**

This subkeyword identifies a reservoir water rate (m<sup>3</sup>/day | STB/day) constraint.

**\*BHP**

This subkeyword identifies a bottom hole pressure (kPa | psi) operating constraint.

**\*WHP**

This subkeyword identifies a well-head pressure (kPa | psi) constraint.

**\*DWN, \*DWA, \*DWB**

This subkeyword identifies a draw-down pressure (kPa | psi) constraint.

**\*BHG**

This subkeyword identifies a reservoir gas rate (m<sup>3</sup>/day | ft<sup>3</sup>/day) constraint.  
This keyword is applicable to injectors only.

**well\_numbers**

Any number of integers, or a range of integers to specify the well numbers to which this target applies. These well numbers must be on the same line as the \*WTINCR keyword. If more than one line is required to specify all the wells, then the \*WTINCR keyword must be repeated.

**well\_names**

Any number of 'well\_names' in quotes to specify the wells to which this target applies. These names must be on the same line as the \*\*WTINCR keyword. If more than one line is required for the well list, then the \*WTINCR keyword must be repeated. Limited wild-carding is available for this list; please see the explanation on the manual page for the \*SHUTIN keyword.

**'@'**

When the '@' symbol in quotes is specified in place of well numbers or well list, then it is considered to be a place holder for the list of wells that have satisfied the trigger condition. This list is necessarily a subset of the well list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. All keywords that precede the '@' place holder must appear on the same line as the '@' place holder. Therefore WTINCR and a keyword specifying a stream, such as STO must be on the same line as the '@' place holder. See the EXPLANATION section of \*TRIGGER keyword for examples of the use of @ place holder.

## Increment(s)

One number for each well identified by well\_numbers or 'well\_names' specifying an increment to be applied. The new value of the specified constraint = old value of the constraint + the increment. Increments must appear on one or more new lines immediately following the \*WTINCR line. Increments must NOT appear on the same line as the list of well names or well numbers. A single increment can be applied to all wells in a well list.

### DEFAULTS:

Optional keywords. No defaults.

### CONDITIONS:

\*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.

\*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 \*WTINCR 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 is specified by using one of \*STO, \*STG, \*STW, \*STL, \*STI, \*BHP, \*WHP, \*DWN, \*DWA, \*DWB, \*BHF, \*BHW or \*BHG. One of these qualifiers must be present.

The \*WTINCR keyword should NOT be used to define a new operating constraint. The multiplier specified with the \*WTINCR keyword will ONLY be applied if the constraint of the type specified with \*WTINCR already exists for the well based on earlier input using the \*OPERATE keyword.

Examples:

```
*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 \*WTINCR keyword may also look like this when several wells have been defined:

```

*WELL 1 'Producer 1'
*WELL 2 'Producer 2'
*WELL 3 'Producer 3'
*WELL 4 'Injector 1'
:
*PRODUCER 1
*OPERATE *MAX *STO 500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 2
*OPERATE *MAX *STO 750.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*PRODUCER 3
*OPERATE *MIN *BHP 2500.0
*MONITOR *MIN *STO 10.00 *SHUTIN
*INJECTOR 4
*OPERATE *MAX *STW 100.0
*MONITOR *MIN *STW 10.00 *SHUTIN
:
*TIME 1200.
** At a later date, want to adjust the target
** constraint values.

** well_numbers
*WTINCR *STO 1:2
** increments
2*125

** well_number
*WTINCR *BHP 3
** increment ** BHP_new = 2500-2000 = 500
-2000.0

** well_number
*WTINCR *STW 4
** values
-5.0

```

---

## **Group Production Constraints Multiplier (Optional)**

**\*GCONPMULT, \*GCONPINCR**

### **PURPOSE:**

\*GCONPMULT/\*GCONPINCR are used to modify existing group production target controls with the use of multipliers or increments.

### **FORMAT:**

\*GCONPMULT 'group\_name\_1' 'group\_name\_2' ... 'group\_name\_n'  
or

\*GCONPMULT '@'

\*GCONPINCR 'group\_name\_1' 'group\_name\_2' ... 'group\_name\_n'  
or

\*GCONPINCR '@'

(\*STO)                    Multiplier/Increment

(\*STG)

(\*STW)

(\*STL)

(\*STI)

(\*WTG)

(\*BHF)

(\*MNP)

(\*CPP)

(\*VREP)                    Multiplier/Increment

(\*PMAINT)

### **DEFINITIONS:**

'group\_name\_1', 'group\_name\_2', ... , 'group\_name\_n'

Are the groups to which the following constraint multiplier applies.

**\*STO**

This subkeyword identifies a surface oil rate (m3/day | STB/day) constraint. Zero multipliers 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 (m3/day | SCF/day) constraint. Zero multipliers 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 (m3/day | STB/day) constraint. Zero rates are allowed and will have the same effect as shutting in all the wells connected to that group.

**\*STL**

This subkeyword identifies a surface liquid rate (oil + water + intermediate liquid) (m<sup>3</sup>/day | STB/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

**\*STI**

This subkeyword identifies a surface intermediate liquid rate (m<sup>3</sup>/day | STB/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group. For more information on the intermediate liquid stream please see the description of the \*WELSEP keyword in this manual.

**\*WTG**

This subkeyword identifies a surface wet gas rate (m<sup>3</sup>/day | SCF/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group. For more information on the wet gas stream please see the description of the \*WELSEP keyword in this manual.

**\*BHF**

This subkeyword identifies a bottom hole fluid rate (m<sup>3</sup>/day | rbbl/day) constraint. Zero rates are allowed and have the same effect as shutting all the wells connected to the group.

**\*MNP**

This subkeyword introduces a manifold pressure (kPa | psi) constraint. This may only be applied if the listed groups have all had production specified as going through a manifold with the \*MANIFOLD keyword.

**\*CPP**

This subkeyword introduces a compressor (surface) pressure (kPa | psi) constraint. This may only be applied if the listed groups have all had production specified as going through a manifold with the \*MANIFOLD keyword.

**\*VREP**

This subkeyword identifies 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.

**\*PMAINT**

This subkeyword identifies that the group production rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector at a desired level. The multiplier applies only to the sector pressure target defined by \*PMTARG.

'@'

When the '@' symbol in quotes is specified in place of group names, then it is considered to be a place holder for the list of groups that have satisfied the trigger condition. This list is necessarily a subset of the group list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. The keyword \*GCONPMULT and the '@' place holder must appear on the same line. See the EXPLANATION section of \*TRIGGER keyword for examples of the use of @ place holder.

**Multiplier**

Multiplier to be applied to the existing constraint value -- see above for units.

**Increment**

Increment to be applied to the existing constraint value -- see above for units.

**DEFAULTS:**

Optional keyword.

**CONDITIONS:**

This keyword must be located in the WELL AND RECURRENT DATA keyword group. A group together with the target constraint for the particular stream or control must be defined using the \*GCONP 'grp\_name' \*TARGET keyword before the target constraint multipliers/increments can be applied. Do not use the \*GCONPMULT/GCONPINCR keywords to define new group target constraints.

**EXPLANATION:**

\*GCONPMULT/GCONPINCR are used to modify group production constraints which have previously been defined using the \*GCONP 'grp\_name' \*TARGET keyword.

Example:

```
*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' \*TARGET \*STW card.

Example:

```
*GCONPMULT 'Group-1'  
*MNP 1.25
```

This sets a manifold pressure target of 125 % of the value specified with the last \*GCONP 'Group-1' \*TARGET \*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:

```
*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' \*TARGET \*STW card less 300.

---

## **Group Injection Constraints Multipliers (Optional)**

**\*GCONIMULT, \*GCONIINCR**

### **PURPOSE:**

\*GCONIMULT/GCONIINCR are used to specify multipliers/increments which modify existing group injection target controls.

### **FORMAT:**

\*GCONIMULT 'group\_name\_1' 'group\_name\_2' ... 'group\_name\_n'

or

\*GCONIMULT '@'

\*GCONIINCR 'group\_name\_1' 'group\_name\_2' ... 'group\_name\_n'

or

\*GCONIINCR '@'

(\*STG)            Multiplier/Increment

(\*STW)

(\*BHG)

(\*BHW)

(\*GMP)

(\*WMP)

(\*GCP)

(\*WCP)

(\*VREP)        (\*GAS)            Multiplier/Increment

(\*WATER)

(\*GMKUP)

(\*WMKUP)

(\*PMAINT)      (\*GAS)            Multiplier/Increment

(\*WATER)

### **DEFINITIONS:**

'group\_name\_1', 'group\_name\_2', ... , 'group\_name\_n'

Are the groups to which the following constraint multiplier(s) apply.

**\*STG**

This subkeyword identifies a surface gas rate (m<sup>3</sup>/day | SCF/day). Zero rates are allowed and have the same effect as shutting in all the gas injection wells connected to that group.

**\*STW**

This subkeyword identifies a surface water rate (m<sup>3</sup>/day | STB/day). Zero rates are allowed and have the same effect as shutting in all the water injection wells connected to that group.

**\*BHG**

This subkeyword identifies a reservoir gas rate (m<sup>3</sup>/day | SCF/day). Zero rates are allowed and have the same effect as shutting in all the gas injection wells connected to that group.

**\*BHW**

This subkeyword identifies a reservoir water rate (m<sup>3</sup>/day | STB/day). Zero rates are allowed and have the same effect as shutting in all the water injection wells connected to that group.

**\*GMP**

This subkeyword identifies a gas manifold pressure (kPa | psi) injection constraint. This subkeyword can only be entered if all of the listed groups have had gas injection identified as going through a manifold with the \*MANIFOLD keyword.

**\*WMP**

This subkeyword identifies a water manifold pressure (kPa | psi ) injection constraint. This subkeyword can only be entered if all of the listed groups have had water injection identified as going through a manifold with the \*MANIFOLD keyword.

**\*GCP**

This subkeyword identifies a gas compressor (surface) pressure (kPa | psi) injection constraint. This subkeyword can only be entered if all of the listed groups have had gas injection identified as going through a manifold with the \*MANIFOLD keyword. Also, a hydraulics table for calculation of the gas manifold-to-surface pressure drop must have been identified for all of the listed groups with the \*GPTABLE keyword.

**\*WCP**

This subkeyword identifies a water surface pressure (kPa | psi) injection constraint. This subkeyword can only be entered if all of the listed groups have had water injection identified as going through a manifold with the \*MANIFOLD keyword. Also, a hydraulics table for calculation of the water manifold-to-surface pressure drop must have been identified for all of the listed groups with the \*GPTABLE keyword.

**\*VREP**

This subkeyword identifies a voidage fraction injection target. This indicates that the injection wells connected to this group inject such that the voidage created by the producers connected to this group is replaced. In this case \*GAS or \*WATER specifies which phase is to be injected to replace the voidage. If more than one phase is being injected to replace the voidage then there must be one \*VREP keyword for each phase. These primary voidage replacement streams are handled independently. One make-up stream can be supplemented with \*GMKUP or \*WMKUP to meet a total voidage replacement fraction.

One of \*GAS, \*WATER, \*GMKUP, or \*WMKUP must be present for each \*VREP keyword.

**\*PMAINT**

This subkeyword identifies that the group injection rates shall be adjusted so as to maintain the hydrocarbon volume weighted average pressure in a particular region/sector at a desired level. The multiplier applies only to the sector pressure target defined by \*PMTARG.

**\*GAS**

Specifies that the gas phase is to be injected for voidage replacement, recycle, or pressure maintenance.

**\*WATER**

Specifies that the water phase is to be injected for voidage replacement, recycle, or pressure maintenance.

**\*GMKUP**

Specifies that gas phase is the make-up stream supplemented to meet the total voidage replacement fraction.

**\*WMKUP**

Specifies that water phase is the make-up stream supplemented to meet the total voidage replacement fraction.

**'@'**

When the '@' symbol in quotes is specified in place of group names, then it is considered to be a place holder for the list of groups that have satisfied the trigger condition. This list is necessarily a subset of the group list input with the trigger keyword. The place holder should only be used in conjunction with trigger actions. The keyword \*GCONIMULT/\*GCONINCR and the '@' place holder must appear on the same line. See the EXPLANATION section of \*TRIGGER keyword for examples of the use of @ place holder.

**Multiplier**

Constraint multiplier value.

**Increment**

Constraint increment value.

**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 being specified with \*GCONIMULT/\*GCONIINCR must be also be previously defined using the keyword \*GCONI.

## **EXPLANATION:**

\*GCONIMULT/GCONIINCR are used to specify multipliers/increments which can be used to modify existing injection group constraint targets. \*GCONIMULT/\*GCONIINCR 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. The stock tank gas injection target must be previously specified with a data line such as:

```
*GCONI 'Group-1' 'Group-2' *TARGET *STG 5.555E+07
```

Example: This is an example using voidage replacement. The water voidage replacement fraction is reduced by half whereas the gas voidage replacement fraction is increased by a factor of 2.

```
*GCONIMULT 'Group1'  
*VREP *WATER 0.5  
  
*GCONIMULT 'Group1'  
*VREP *GAS 2.0
```

The voidage replacement targets must be previously specified with data lines such as:

```
*GCONI 'Group-1' 'Group-2' *VREP *WATER 0.6  
*GCONI 'Group-1' 'Group-2' *VREP *GAS 0.4  
  
*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' *TARGET *STG 5.555E+07
```

---

## Allow a Cycling Group to be Defined (Optional)

\*GCONCYCLE\_START, \*GCONCYCLE\_END

### PURPOSE:

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

### FORMAT:

\*GCONCYCLE\_START 'group\_names'

\*CYCSTREAMS 'stream\_name\_list' (total of nstreams) – required

\*NPARTS nparts – required

\*MAXRATES (# of streams must be equal to nstreams) – required

\*TARGETTYPES (# of streams must be equal to nstreams) – required

'stream\_name1' 'list\_of\_target\_types' (# of values must be equal to nparts)

'stream\_name2' 'list\_of\_target\_types' (# of values must be equal to nparts)

\*MAXRATES (# of streams must be equal to nstreams) -- required

'stream\_name1' maximum\_rate, imposed as a group target (# of values must be equal to nparts)

'stream\_name2' maximum\_rate, imposed as a group target (# of values must be equal to nparts)

\*VREPFR (# of streams must be equal to nstreams) -- required

'stream\_name1' voidage replacement fraction, imposed as a group target (# of values must be equal to nparts)

'stream\_name2' voidage replacement fraction, imposed as a group target (# of values must be equal to nparts)

\*MAXCUMS -- optional

'stream\_name1' maximum cumulative injection/production (# of values must be equal to nparts)

'stream\_name2' maximum cumulative injection/production (# of values must be equal to nparts)

\*TOTHET – optional, can be specified for injection streams ONLY, for STARS

stream\_name1 maximum cumulative enthalpy injection (# of values must be equal to nparts)

stream\_name2 maximum cumulative enthalpy injection (# of values must be equal to nparts)

\*MAXBHP – optional, can be specified for injection streams ONLY

stream\_name1 maximum BHP (# of values must be equal to nparts)

stream\_name2 maximum BHP (# of values must be equal to nparts)

\*MAXTIMES ntime1 ntime2 (# of values must be equal to nparts) –optional

\*MINTIMES ntime1 ntime2 (# of values must be equal to nparts) –optional

\*DTWCYC dtime1 dtime2 (# of values must be equal to nparts) --optional

\*NCYCLES ncycles –optional

\*MINBHP bhp1 bhp2 (# of values must be equal to nparts) –optional

\*MINQOIL minoil1 minoil2 (# of values must be equal to nparts) --optional

\*DEPNDX depndx1 depndx2 ((# of values must be equal to nparts) --optional

\*GCONCYCLE\_END --required

## DEFINITIONS:

\*GCONCYCLE\_START

This keyword indicates the start of keyword data input for group cycling control. The complementary keyword \*GCONCYCLE\_END signals the end of group cycling keyword data input.

\*CYCSTREAMS

This keyword is used to specify the list of streams that participate in the cycling process. A stream name can be one of the following: Production stream, or ‘PROD’, or injection streams, gas ‘GASI’, water ‘WATI’ or for the IMEX simulator only solvent ‘SOLI’. For each stream a target type must be entered for every cycle part using the keyword \*TARGETTYPES as well as a maximum rate for every cycle part using the keyword \*MAXRATES or a voidage replacement fraction (for injection streams) using the keyword \*VREPFR. 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')  Bottom hole fluid ('BHF') – includes oil, gas and water phase production at reservoir or bottom hole conditions as well as production of the 4 <sup>th</sup> phase, if defined such as solvent or polymer  Stock tank liquid ('STL') – includes production of oil and water phases at surface or stock tank conditions	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  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 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  Stock tank liquid ('STL') – includes production of oil and water phases at surface or stock tank conditions
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.

<b>INPUT UNIT SYSTEM / TARGET TYPE</b>	<b>SI &amp; MODI SI UNITS (IMEX, GEM, STARS)</b>	<b>FIELD UNITS (IMEX, GEM, STARS)</b>	<b>LAB UNITS (IMEX, STARS)</b>
STO, STW, STL, STI, BHF, BHW	m <sup>3</sup> /day	bbl/day	cm <sup>3</sup> /min
WTG, STG, BHG, STS, BHS	m <sup>3</sup> /day	ft <sup>3</sup> /day	cm <sup>3</sup> /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.

<b>INPUT UNIT SYSTEM / TARGET TYPE</b>	<b>SI &amp; MODI SI UNITS (IMEX, GEM, STARS)</b>	<b>FIELD UNITS (IMEX, GEM, STARS)</b>	<b>LAB UNITS (IMEX, STARS)</b>
STO, STW, STL, STI, BHF, BHW	m <sup>3</sup>	bbl	cm <sup>3</sup>
WTG, STG, BHG, STS, BHS	m <sup>3</sup>	ft <sup>3</sup>	cm <sup>3</sup>

#### \*TOTHET

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.

<b>INPUT UNIT SYSTEM / TARGET TYPE</b>	<b>SI &amp; MODI SI UNITS (STARS)</b>	<b>FIELD UNITS (STARS)</b>	<b>LAB UNITS (STARS)</b>
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.

<b>INPUT UNIT SYSTEM</b>	<b>SI &amp; MODI SI UNITS (IMEX, GEM, STARS)</b>	<b>FIELD UNITS (IMEX, GEM, STARS)</b>	<b>LAB UNITS (IMEX, STARS)</b>
BHP	kPa (SI) kg/cm <sup>2</sup> (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 <sup>2</sup> (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 <sup>3</sup> /day	bbl/d	cm <sup>3</sup> /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 <sup>3</sup> /day	bbl/d	cm <sup>3</sup> /min
GASI (STG/BHG)	m <sup>3</sup> /day	ft <sup>3</sup> /day	cm <sup>3</sup> /min
SOLI (STS/BHS) IMEX only	m <sup>3</sup> /day	ft <sup>3</sup> /day	cm <sup>3</sup> /day

## DEFUALTS:

There 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\_name1' 'group\_name2'

(Any of \*TARGETTYPES, \*MAXCUM, \*MAXTIMES, \*NCYCLES,  
MAXBHP, \*MININJ, \*MINBHP, \*MINOIL, \*DEPNDX, \*TOTHEAT

\*GCONCYCR\_END

### **DEFAULTS:**

The default is to continue applying values originally specified with \*GCONCYCLE\_START.

### **CONDITIONS:**

This block of keywords can only be specified for a cycling group if that group is previously defined using the keyword \*GCONCYCLE\_START. Any or all of the group cycle part switching criteria specified earlier with \*GCONCYCLE\_START can be revised or new criteria added. However this keyword block cannot be used to change the number of cycle parts (keyword \*NPARTS) or the number and type of streams (keyword CYCSTREAMS). Number of parts or streams can only be changed by redefining the group using \*GCONCYCLE\_START.

### **EXPLANATION:**

See keyword \*GCONCYCLE\_START

Once a cycling group is defined using \*GCONCYCLE\_START this keyword block can be used to modify some of the data used to test for cycle part duration. It is also possible to add new criteria to determine cycle part end that were not previously specified with \*GCONCYCLE\_START. The data entered will take effect immediately. Consider the following example:

```
Time 0.0
GCONCYCLE_START 'cyc_group1'
cycstreams 'PROD' 'WATI'
nparts 3
ncycles 40
targettypes
  'WATI' 'STW' 'STW' 'STW'
  'PROD' 'STL' 'STL' 'STL'
maxrates
  'WATI' 1000. 0. 0.
  'PROD' 0. 0. 1000.0
```

```
maxcums
'PROD' 0. 0. 600000.0
tothet
'WATI' 5.0E+09 0. 0.
maxbhp
'WATI' 1100.0 0.0 0.0
maxtimes 10.0 3.0 348.0
dtwcyc 0.02 1.0E+5 1.0

GCONCYCLE_END

time 365.0 ** modify BHP criterion

GCONCYCR_START 'cyc_group1'
*maxbhp
'WATI' 400.0 0.0 0.0
GCONCYCLE_END
```

---

## Allow Cycling Group Data to be Revised (Optional)

**\*GCONCYCLE**

### PURPOSE:

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

### FORMAT:

\*GCONCYCLE 'group\_name1' 'group\_name2' \*ON|OFF

### DEFAULTS:

Once data is specified is specified with \*GCONCYCLE\_START, the default is \*GCONCYCLE \*ON.

### CONDITIONS:

This block of keywords can only be specified for a cycling group if that group is previously defined using the keyword \*GCONCYCLE\_START. Group cycling control can be specified at any time once the group is defined using \*GCONCYCLE\_START.

### EXPLANATION:

See keyword \*GCONCYCLE\_START

Once a cycling group is defined using \*GCONCYCLE\_START this keyword can be used to turn off group cycling or to impose group cycling once more if group cycling was previously turned off. The desired state of group cycling control specified with this keyword will take effect immediately. If GCONCYCLE \*OFF is specified for a group previously under group control, then group cycling control will be terminated immediately and the group target(s) and well state existing when GCONCYCLE \*OFF is specified will continue to be in effect. If actions are specified as part of the special trigger provided to monitor the event of group cycling control being turned off, then that trigger condition will be satisfied and the stipulated actions will take effect at the top of the next timestep. If GCONCYCLE \*ON is specified at a time when group control is turned off then group cycling control will resume with the next cycle part following the part when group cycling was turned off. The count of number of cycle parts completed is preserved between applications of GCONCYCLE \*ON/\*OFF.

Consider the following example:

```
Time 0.0
GCONCYCLE_START 'cyc_group1'
cycstreams 'PROD' 'WATI'
nparts 3
ncycles 40
targettypes
  'WATI' 'STW' 'STW' 'STW'
  'PROD' 'STL' 'STL' 'STL'
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_name1' 'group_name2' cycle_part_start  
*CYCPRT_END 'group_name1' 'group_name2' cycle_part_end
```

#### DEFAULTS:

Once data is specified is specified with \*GCONCYCLE\_START, the default for group cycle starting part is 1 and for group cycle ending part is the value specified with \*NPARTS, that is the last cycle part.

#### CONDITIONS:

These can only be specified for a cycling group if that group is previously defined using the keyword \*GCONCYCLE\_START.

#### EXPLANATION:

See keyword \*GCONCYCLE\_START

Once a cycling group is defined using \*GCONCYCLE\_START these keywords can be used to select the starting and ending cycle parts. The values specified with these keywords will take effect immediately. If cycling group control is in effect when \*CYCPRT\_START is specified then the value specified with \*CYCPRT\_START will be ignored. If \*CYCPRT\_START is specified following \*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
```

---

## **Constant and Convective Heat Transfer Model**

\*HEATR, \*TMPSET,  
\*UHTR, \*UHTRAREAI-, \*UHTRAREAI+, \*UHTRAREAJ-, \*UHTRAREAJ+, \*UHTRAREAK-,  
\*UHTRAREAK+, \*AUTOHEATER, \*AUTOCOOLER

### **PURPOSE:**

Assign data for constant and convective heat transfer models.

### **ARRAY:**

\*HEATR  
\*TMPSET  
\*UHTR  
\*UHTRAREAI-  
\*UHTRAREAI+  
\*UHTRAREAJ-  
\*UHTRAREAJ+  
\*UHTRAREAK-  
\*UHTRAREAK+

### **FORMAT:**

\*AUTOHEATER (\*ON | \*OFF) *uba\_range*  
\*AUTOCOOLER (\*ON | \*OFF) *uba\_range*

### **DEFINITIONS:**

#### **\*HEATR**

Assign constant heat transfer rate to grid blocks (J/day | Btu/day | J/min). This constant rate is added to the proportional part given by \*UHTR and \*TMPSET.

#### **\*UHTR**

Proportional heat transfer coefficient, used in conjunction with \*TMPSET (J/day-C | Btu/day-F | J/min-C).

- |     |  |
|-----|--|
| > 0 | gain coefficient of a temperature controller. The rate of heat gain is UHTR * (TMPSET - T) while TMPSET > T; otherwise, the rate is zero. This can model a heater which surrounds a combustion tube jacket serving as a heat loss compensator. When the fire front reaches the heater and the tube temperature T exceeds TMPSET, the heater shuts off. |
| < 0 | overall convective heat transfer coefficient. The rate of heat loss is ABS(UHTR) * (T - TMPSET) while T > TMPSET; otherwise, the rate is zero.   |

\*UHTRAREAI-, \*UHTRAREAI+, \*UHTRAREAJ-, \*UHTRAREAJ-, \*UHTRAREAK-,  
\*UHTRAREAK+

Heat transfer coefficient  $U_a$  per unit area (J/m<sup>2</sup>-day-C | Btu/ft<sup>2</sup>-day-F | J/cm<sup>2</sup>-min-C), in the indicated direction. In the following  $A$  is the cross-sectional area in the indicated direction.

- > 0 Gain coefficient of a temperature controller. The rate of heat gain is  $U_a \cdot A \cdot (\text{TMPSET} - T)$  while  $\text{TMPSET} > T$ ; otherwise, the rate is zero. This can model heat transfer through a specified grid block face. When temperature  $T$  exceeds TMPSET, the heater shuts off.
- < 0 Overall convective heat transfer coefficient. The rate of heat loss is  $|U_a| \cdot A \cdot (T - \text{TMPSET})$  while  $T > \text{TMPSET}$ ; otherwise, the rate is zero.

Heat transfer is through a specified face of a grid block, where + or - refers to the face between the referenced block and neighbour with the higher index in that direction. For example, to access the face between blocks (i,j,k) and (i+1,i,j) use either \*UHTRAREAI+ from (i,j,k) or \*UHTRAREAI- from (i+1,j,k).

#### \*TMPSET

Temperature setpoint (C | F) of a temperature controller ( $\text{UHTR} > 0$ ), or reference temperature ( $\text{UHTR} < 0$ ). This temperature must be entered for each block which has a non-zero \*UHTR or \*UHTRAREAI-, etc. The allowed range is the same as the temperature limits for the run, which are determined by keywords \*MINTEMP and \*MAXTEMP.

#### \*AUTOHEATER ( \*ON | \*OFF ) *uba\_range*

The autoheater option is applied to a range of blocks. This option makes the constant and proportional heat gain models work together to mimic a heater control that operates with constant heat rate below a given temperature and shuts off above that temperature. See section **User Block Address** in Keyword Data Entry System chapter for a description of UBA ranges.

With this option, a block's heat rate is the minimum of (a) the constant rate specified via \*HEATR and (b) the rate calculated from \*UHTR, etc., and \*TMPSET. Typically, \*HEATR is the maximum heating rate that can be sustained by the heater in that block. The maximum desired temperature is assigned to \*TMPSET. The value to assign to \*UHTR is the constant rate from \*HEATR divided by the desired difference between block temperature and \*TMPSET at switchover from the constant model to the proportional model.

Normally a block starts heating at constant rate, and switches over to the proportional model as  $T$  nears TMPSET. If  $T > \text{TMPSET}$ , the heating rate is zero. A larger value of UHTR will sharpen the transition from constant to proportional heating, but usually at the expense of more Newton iterations. A typical switchover temperature difference (HEATR/UHTR) is 10 degrees.

### \*AUTOCOOLER (\*ON | \*OFF) *uba\_range*

The autocooler option is applied to a range of blocks. This option makes the constant and proportional heat loss models work together to mimic a cooler control that operates with constant cooling rate above a given temperature and shuts off below that temperature. See section **User Block Address** in Keyword Data Entry System chapter a description of UBA ranges.

With this option, a block's cooling rate is the minimum of (a) the constant rate specified via \*HEATR and (b) the rate calculated from \*UHTR, etc., and \*TMPSET. Typically, \*HEATR is the maximum cooling rate that can be sustained in that block. The minimum desired temperature is assigned to \*TMPSET. The value to assign to \*UHTR is the constant rate from \*HEATR divided by the desired difference between block temperature and \*TMPSET at switchover from the constant model to the proportional model.

Normally a block starts cooling at constant rate, and switches over to the proportional model as T nears TMPSET. If  $T < \text{TMPSET}$ , the cooling rate is zero. A larger value of UHTR will sharpen the transition from constant to proportional cooling, but usually at the expense of more Newton iterations. A typical switchover temperature difference (HEATR/UHTR) is 10 degrees.

#### DEFUALTS:

\*HEATR \*CON 0

\*UHTR \*CON 0

\*UHTRAREAI-, etc., all default to zero.

If \*AUTOHEATER is absent then \*OFF is assumed for all blocks.

If \*AUTOCOOLER is absent then \*OFF is assumed for all blocks.

#### CONDITIONS:

With the array-reading option \*IJK, referring to only select grid blocks is allowed.

For the convective heating model, it is recommended that you use either \*UHTR or \*UHTRAREAI-, etc., but not both. \*UHTR will overwrite previous data (on a block by block basis), whereas with \*UHTRAREAI-, etc., contributions will be added to what has been specified by \*UHTR up to that point.

\*TMPSET must be entered for each block which has a none-zero \*UHTR or \*UHTRAREAI-, etc.

\*AUTOHEATER may not be used with heat loss, that is, negative values of \*HEATR or \*UHTR. \*AUTOCOOLER may not be used with heat gain, that is, positive values of \*HEATR or \*UHTR.

If an attempt is made to enable both \*AUTOHEATER and \*AUTOCOOLER for the same block at the same time, \*AUTOCOOLER will be ignored for that block. All other combinations of those two keywords are allowed.

A heater model may be applied only to a non-null block that is not the parent of a refine grid.

## **EXPLANATION:**

### **Data Echo and Output**

Heater control data is echoed in the “.out” file under “Summary of Current Heater Specifications” each time heater data is changed. Heater performance from all heater types lumped together can be dumped to SR2 via \*OUTSRF \*GRID and to the “.out” file via \*OUTPRN \*GRID using subkeywords \*CCHLOSS for rate and \*CCHLOSSCUM for accumulation. Lumped heater performance is available also for sectors (SR2 and .out file).

The “.out” file includes a detailed summary “Heater Rate Split” which reports how much of each block’s heating rate comes from the individual heating models (constant, proportional, adiabatic or heater well). This is especially useful with \*AUTOHEATER and \*AUTOCOOLER which switch automatically between constant and proportional models.

### **Inheritance Issues**

Extrinsic quantities \*HEATR, \*UHTR and \*UHTRAREAI-, etc., use inheritance only to assign values to fine blocks. The parent value is not split amongst the child blocks. For example, fundamental block (1,1,1) is refined into a grid of  $2 \times 2 \times 2 = 8$  fine blocks which inherit their \*HEATR values from their parent block.

\*HEATR \*IJK 1:2 1 1 200.

assigns the value 200 to each of the 8 fine blocks (not  $200 / 8 = 25$ ) as well as to block (2,1,1). On the other hand, if you had intended that the value of \*HEATR be the same on a per-volume basis for these blocks, then the data should be

\*HEATR \*IJK 2 1 1 200.

\*HEATR \*IJK 1 1 1 25.

-or-

\*HEATR \*IJK 1:2 1 1 200.

\*MOD 1 1 1 / 8

Care must be taken to assign values appropriate to the individual quantity. For example, \*HEATR could be regarded as a quantity based either on volume or on area in some direction. Also, \*UHTR and \*UHTRAREAI-, etc., usually are assigned to blocks on a grid boundary. When a boundary block is refined, not all the fine blocks are on the boundary so \*UHTR must be assigned to only the fine blocks on the boundary, requiring the use of \*RG instead of inheritance.

### **Matrix/Fracture Addressing Defaults**

Use of \*AUTOHEATER or \*AUTOCOOLER together with a natural fracture grid option can lead to unexpected results when matrix/fracture address defaulting is used. This defaulting occurs when reference is made to a block without explicitly indicating matrix or fracture. This situation usually occurs when a natural fracture option is enabled in a pre-existing single-porosity data set and explicit matrix/fracture addressing is not added.

Keyword \*AUTOHEATER uses a User Block Address (UBA) to refer to one or more blocks. The UBA default is “fracture”, so in a natural fractured grid the UBA “i,j,k” implies the fracture alone which would be equivalent to UBA “i,j,k FR”. On the other hand, keyword \*HEATR is a grid array (see **Input of Grid Property Arrays** in the Keyword Data Entry System chapter). The grid array default is “matrix and fracture”, so in a natural fractured grid an assignment to block (i,j,k) via

```
*HEATR *IJK i j k value
```

would be the same as

```
*HEATR *MATRIX *IJK i j k value  
*HEATR *FRACTURE *IJK i j k value
```

Since \*AUTOHEATER and \*HEATR have different matrix/fracture addressing defaults, the autoheater control (on fracture only) does not match the heater (on matrix and fracture), leading to unexpected results.

For this reason, it is recommended that block assignments be given explicit matrix/fracture addresses, that is, (i,j,k FR) and/or (i,j,k MT) for UBA and \*MATRIX and/or \*FRACTURE for grid arrays.

---

## Adiabatic Heat Transfer Control

\*ADHEAT

### PURPOSE:

Assign data for controlling adiabatic heat gain.

### FORMAT:

```
*ADHEAT heat_blk heat_coef(T_diff) *REF ref_blk  
*ADHEAT heat_blk heat_coef(T_diff)
```

### DEFINITIONS:

*heat\_blk*

List of blocks to which heat will be added, in UBA single or range format. A warning is issued for each block that is null and an error is issued for each block that is the parent of a refined grid. See section **User Block Address** in Keyword Data Entry System chapter.

*heat\_coef*

Proportional heat gain coefficient (J/day-C | Btu/day-F | J/min-C). This number includes the cross-sectional area through which heat is transferred, similar to  $*UHTR > 0$ . The value must not be negative. A zero value implies no heat gain.

*T\_diff*

Temperature difference cut-off ( C deg | F deg ). The difference between heater and reference temperature must be greater than *T\_diff* for heat transfer to occur. *T\_diff* must not be less than 0. This quantity is optional and is assumed to be unchanged if absent. The unit is temperature difference, so the value will be the same in C and K, and will be the same in F and R.

*\*REF ref\_blk*

List of blocks whose temperatures help control the rate of heat gain, in UBA single or range format. A warning is issued for each block that is null and an error is issued for each block that is the parent of a refined grid.

This subkeyword establishes the association between *heat\_blk* and *ref\_blk*, and is required the first time *heat\_blk*'s \*ADHEAT parameters are defined in the data. Changes to *heat\_coef* and *T\_diff* may be made later in the run without *\*REF ref\_blk* if *ref\_blk* does not change.

*ref\_blk* must refer to the same number of blocks as *heat\_blk*, so that a one-to-one correspondence can be made. See **User Block Address** in the Keyword Data Entry System chapter for comments on UBA range ordering.

### DEFAULTS:

If \*ADHEAT is not specified for a block, *heat\_coef* is assumed to be zero. However, other heater options like \*HEATER or \*UHTR may produce a non-zero rate.

Each *heat\_blk*'s *T\_diff* has the value 0 until it is explicitly changed.

## CONDITIONS:

The adiabatic control of a range of blocks may be specified via *heat\_blk*, but there must a one-to-one correspondence between the *heat\_blk* range and the *ref\_block* range.

Each *heat\_blk*'s *ref\_blk* must be explicitly defined when *heat\_coef* is first defined. A *heat\_blk* may not be its own *ref\_blk*, that is, two distinct blocks must be used.

Keyword \*HEATSLAVE may not use *heat\_blk* as either slave or master block.

## EXPLANATION:

The rate of heat gain to block *heat\_blk* as a function of the temperatures T in *heat\_blk* and *ref\_blk* is

$$\begin{aligned} &= \text{heat\_coef} * [ T(\text{ref\_blk}) - T_{\text{diff}} - T(\text{heat\_blk}) ] \\ &\quad \text{when } T(\text{heat\_blk}) < T(\text{ref\_blk}) - T_{\text{diff}}; \\ &= 0 \text{ otherwise.} \end{aligned}$$

In this expression,  $T(\text{ref\_blk})$  is the value at the beginning of the timestep, to maximize stability. On the other hand,  $T(\text{heat\_blk})$  is the most current value since it is one of the primary iterating variables for the heater block. The chief difference between \*ADHEAT and \*UHTR is that the temperature setpoint is constant with \*UHTR but is a block temperature with \*ADHEAT.

\*ADHEAT may be used together with \*UHTR which can model the heat loss from the outer heater block independently of the heater action. A summary of heater specifications is echoed in the output (.out) file each time \*ADHEAT appears, and a summary of adiabatic heater block rates is printed at full grid printouts (controlled by \*WPRN \*GRID).

Example: Assume heating coefficient of 10, loss coefficient of 1.2, ambient temperature of 20 deg and temperature cut-off of 2 deg. The grid is cylindrical with ni = 5, nj = 1 and nk = 40. The following are some valid uses of \*ADHEAT.

```
** Adiabatic control of skin heater
    using probe T in center, with heat loss
*ADHEAT 5 1 1 10. 2. *REF 1 1 1
*UHTR *IJK 5 1 1 1.2
*TMPSET *IJK 5 1 1 20.
** Adiabatic control of series of heaters
*ADHEAT 5 1 1:40 10. 2. *REF 1 1 1:40
*UHTR *IJK 5 1 1:40 1.2
*TMPSET *IJK 5 1 1:40 20.
** Change value of heat_coef later in run
*ADHEAT 5 1 1:40 16.
** Turn off heater but not heat loss
*ADHEAT 5 1 1:40 0.
** Control of several heaters from single
    reference block; T_diff = 0.
*ADHEAT 5 1 1 10. *REF 1 1 1
*ADHEAT 5 1 2 10. *REF 1 1 1
```

## Data Echo and Output

See the EXPLANATION for \*HEATR.

---

## Slaved Heater Control

\*HEATSLAVE

### PURPOSE:

Assign data for slaved heater control.

### FORMAT:

\*HEATSLAVE *slave\_block* *factor\_option* *master\_block*  
\*HEATSLAVE *slave\_block* \*OFF

### DEFINITIONS:

*slave\_block*

Address in UBA format of “slave” block to which heat will be added. An error is issued when *slave\_block* is null.

*factor\_option*

Specifies the factor to use when calculating the slave block’s heater rate from the master block’s heater rate. In the following table, “ratio” refers to the slave block’s value divided by the master block’s value.

*OFF	<i>factor</i> = 0; used to turn off an active heater. In this case, <i>master_block</i> is absent.
*USER x	<i>factor</i> = x; allows manual factor entry when the other options are not appropriate.
*GROSSVOL	<i>factor</i> = ratio of block gross volumes. This corresponds to uniform heating throughout a block’s volume.
*IAREA	<i>factor</i> = ratio of cross-sectional areas normal to the I direction. This corresponds to heating occurring on a block’s J-K face.
*JAREA	<i>factor</i> = ratio of cross-sectional areas normal to the J direction.
*KAREA	<i>factor</i> = ratio of cross-sectional areas normal to the K direction.

*master\_block*

Address in UBA format of “master” block whose heat rate determines the heat rate for the “slave” block. An error is issued when *master\_block* is null. *Master\_block* must not appear when *factor\_option* is \*OFF.

### DEFAULTS:

If \*HEATSLAVE is not specified for *slave\_block*, *factor* is assumed to be zero. However, other heater options \*HEATER or \*UHTR may produce a non-zero heat rate.

### CONDITIONS:

The master block must not be a slave to another block through \*HEATSLAVE, that is, nested slaving is not allowed. In particular, the slave block must be different from the master block.

The master block may not be heated via \*ADHEAT.

Both slave and master blocks must not be the parent of a refined grid.

## **EXPLANATION:**

The heater rate of the *slave\_block* is

$$\text{slave\_block heater rate} = \text{factor} * \text{master\_block heater rate}$$

where *factor* is determined by *factor\_option*.

Any heating option may be used for *master\_block*, with the exception that its heat rate must not depend directly on *slave\_block* via \*ADHEAT or \*HEATSLAVE.

\*HEATSLAVE may be used together with other heater options. A summary of heater specifications is echoed in the output (.out) file each time \*HEATSLAVE appears, and a summary of adiabatic heater block rates is printed at full grid printouts (controlled by \*WPRN \*GRID).

## **Data Echo and Output**

See the EXPLANATION for \*HEATR.

## **Example**

A radial grid with 20 nonuniform K direction layer thickness has a strip heater in the centre that passes through layers 8 to 15. The reservoir temperature at layer 10 controls the output of the entire strip. The grid is refined near the centre in layer 11. The following keywords represent this situation.

```
** Single-point control of multi-layer strip heater
.
.
.*GRID *RADIAL 15 1 20      ** 20 K layers
.*DK 5*5. 4. 3. 2. 1.4 1.2 1.1 1.5 2.1 3.5 6*5.
.*REFINE 1 1 11 *INTO 1 1 3  ** Layer 11 is refined
.
.
*TIME 100.
*UHTRAREAI- *IJK 1 1 10 30.5  ** Heater control
*TMPSET     *IJK 1 1 10 360.
*HEATSLAVE 1 1 8 *IAREA 1 1 10
*HEATSLAVE 1 1 9 *IAREA 1 1 10
*HEATSLAVE 1 1 11 / 1 1 1 *IAREA 1 1 10
*HEATSLAVE 1 1 11 / 1 1 2 *IAREA 1 1 10
*HEATSLAVE 1 1 11 / 1 1 3 *IAREA 1 1 10
*HEATSLAVE 1 1 12 *IAREA 1 1 10
*HEATSLAVE 1 1 13 *IAREA 1 1 10
*HEATSLAVE 1 1 14 *IAREA 1 1 10
*HEATSLAVE 1 1 15 *IAREA 1 1 10
*TIME 140.
*UHTRAREAI- *IJK 1 1 10 0.    ** Turn off heater
.
.
*TIME 240.
*UHTRAREAI- *IJK 1 1 10 30.5  ** Heater on again
*TMPSET     *IJK 1 1 10 450.
```

---

## Heater Well

\*HTWELL

### PURPOSE:

Assign data for a heater well.

### FORMAT:

\*HTWELL *well\_name* (*rate\_option*) (*T\_option*) (*I\_option*)

or

\*HTWELL *well\_name* \*OFF

where

*rate\_option* = \*HTWRATE  $Q_{hspec}$  | \*HTWRATEPL  $Q_{hspec}$

*T\_option* = \*HTWTEMP  $T_{wspec}$

*I\_option* = \*HTWI

### DEFINITIONS:

*well\_name*

Single quoted well name previously defined by keyword \*WELL. No wildcarding is allowed.

\*HTWRATE  $Q_{hspec}$

Specify maximum total heating rate  $Q_{hspec}$  (J/day | Btu/day) which is distributed evenly over the total length of the well by assuming a uniform per-length heating rate. A positive value denotes heating and a negative value denotes cooling.

\*HTWRATEPL  $Q_{hspec}$

Specify maximum total heating rate per well length  $Q_{hspec}$  (J/day-m | Btu/day-ft). This quantity is multiplied internally by the total well length to get total well heat rate  $Q_{hspec}$ , and then execution proceeds the same as if  $Q_{hspec}$  had been entered via \*HTWRATE.

\*HTWTEMP  $T_{wspec}$

Specify wellbore temperature  $T_{wspec}$  (C | F).

\*HTWI

Specify heat index  $I_{hk}$  (J/day-C | Btu/day-F) directly via the \*WI option of \*PERF or \*PERFV together with well fraction \*WELL \*FRAC. The resulting value is interpreted as heat index instead of fluid index. This option assumes that the fluid well is shut in since the \*WI quantity cannot be used for both fluid and heat index at the same time.

\*OFF

Stop the transfer of heat, that is, disable the heater well. Heat will still move between grid blocks by conduction and fluid convection.

## **DEFAULTS:**

Keyword \*HTWELL is optional.

If \*HTWI is absent after \*HTWELL, the quantity specified by the \*WI option of \*PERF and \*PERFV is not interpreted as heat conduction index.

## **CONDITIONS:**

At least one of *rate\_option*, *T\_option* or \*OFF must be present after \*HTWELL.

\*HTWELL and other heater options (\*HEATR, \*UHTR, \*ADHEAT, etc.) may be applied to the same block, but this practice is not recommended.

At any one time, all enabled heater wells may not have any active completion layers in common. This does not apply to heater wells that are disabled as well as individual completion layers that are shut in. Since different fluid wells are allowed to have grid cells in common, this is an additional restriction of heater wells.

## **EXPLANATION:**

Of the several methods for specifying heating or cooling for certain grid blocks, the heater well option allows you to use \*WELL and \*PERF data already entered for a fluid well.

Keyword \*HTWELL may be used with a fluid well of any type (\*INJECTOR, \*PRODUCER) and open status (\*OPEN, \*SHUTIN) and with any perforation data (\*WELL, \*GEOMETRY, \*PERF, \*PERFV, \*LAYERIJK, \*LAYERXYZ).

In the following,

$L_k$  - length of the layer k well completion,

$L_w$  - total well length (sum of  $L_k$ ),

$q_{hk\text{spec}}$  - specified (target) heat rate for layer k,

$q_{hk}$  - resulting heat rate for layer k,

$I_{hk}$  - conduction index for layer k,

$T_k$  - block temperature for layer k,

$Q_h$  - sum of  $q_{hk}$  over the well's layers.

## **Heat conduction index**

Since thermal conductivity is a direct analogue of permeability, conduction index  $I_{hk}$  can be viewed as the *same function* of geometry as the fluid-flow index. The fluid-flow index is a potentially complex function of the geometry (location and orientation) of the well in the block as well as the block's absolute permeabilities in the three spatial directions. For radial inflow performance the most common calculation is that of Peaceman, but also supported are radial, hybrid and corner-point grids, as well as the tube-end option.

There are two methods for specifying the heat conduction index:

1. Enter  $I_{hk}$  directly via the \*WI option of \*PERF and \*PERFV, using \*HTWI to flag interpretation of the \*WI value as heat conduction index. This option assumes that the fluid well is shut in since the \*WI quantity cannot be used for both fluid and heat conduction index at the same time. If you need both the fluid and heater wells active at the same time and need to use the \*WI option for both, then for the heater well make a duplicate of the fluid well.

- Specify internal calculation via the other options of \*PERF and \*PERFV, using information from \*GEOMETRY, \*LAYERXYZ, etc. The geometrical factor in the heat conduction index will be the same as that in the fluid index, providing a “match” between the two indices. For example, for fluid index 5000 md-ft and permeability 1000 md (in plane normal to well direction), the geometrical factor is  $(5000 \text{ md-ft})/(1000 \text{ md}) = 5.00 \text{ ft}$ . For a thermal conductivity of 24 Btu/ft-day-F, the heat conduction index in the same layer will be  $(24 \text{ Btu/ft-day-F}) \cdot (5.00 \text{ ft}) = 120 \text{ Btu/day-F}$ .

If the fluid-flow index is entered directly via subkeyword \*WI but \*HTWI is absent, the heat conduction index is calculated by the \*GEO method using the last wellbore data entered via keyword \*GEOMETRY (or its defaults if it is absent). In this case the conduction index may not “match” the fluid flow index. In addition the specified (or more likely defaulted) well radius may be too large for the cell size resulting in a negative heat conduction index, in which case a fatal error message is issued.

### **Rate control**

When \*HTWRATE or \*HTWRATEPL is specified but \*HTWTEMP is not, the heater well operates on specified heat rate  $Q_{hspec}$  only. In this case a uniform per-length heating rate is assumed, resulting in constant layer k heat rate

$$q_{hk} = q_{hkspec} = Q_{hspec} \cdot L_k / L_w$$

The calculated well total  $Q_h$  will be  $Q_{hspec}$ .

### **Temperature control**

When \*HTWTEMP is specified but \*HTWRATE and \*HTWRATEPL are not, the heater well operates on specified wellbore temperature  $T_{wspec}$  only. In each layer the heat rate is

$$q_{hk} = I_{hk} \cdot (T_{wspec} - T_k)$$

and the resulting  $Q_h$  varies. Note that in this case  $Q_h$  will be positive or negative, depending only on the relative values of  $T_{wspec}$  and  $T_k$ . If you wish to prevent heat transfer of a particular sign, you must use dual controls.

### **Dual rate/temperature control**

When both \*HTWTEMP and a heat rate (\*HTWRATE or \*HTWRATEPL) are specified, control switches automatically between them on a per-layer basis.  $Q_{hspec}$  is distributed among the layers as

$$q_{hkspec} = Q_{hspec} \cdot L_k / L_w$$

For heating ( $Q_{hspec} \geq 0$ ) the layer heat rate expression

$$q_{hk} = \min[ I_{hk} \cdot (T_{wspec} - T_k), q_{hkspec} ]$$

causes  $Q_h$  to be the minimum of  $Q_{hspec}$  and the rate calculated from  $T_{wspec}$  (zero when  $T_k > T_w$ ), similar to \*AUTOHEATER.

For cooling ( $Q_{hspec} < 0$ ) the layer heat rate expression

$$q_{hk} = \max[ I_{hk} \cdot (T_{wspec} - T_k), q_{hkspec} ]$$

causes  $Q_h$  to be the maximum of  $Q_{hspec}$  and the rate calculated from  $T_{wspec}$  (zero when  $T_k < T_w$ ), similar to \*AUTOCOOLER.

If you use \*HTWTEMP and wish to restrict heat transfer to one sign, specify via \*HTWRATE a small non-zero rate of the opposite sign. For example, to get only positive heat transfer (heat transfer is zero instead of negative for  $T_k > T_{wspec}$ ), specify a very small negative rate (e.g.,  $-10^{-6}$ ). Remember that a rate of exactly zero turns the heater well off completely.

### Data Echo and Output

Heat rates and accumulations generated by \*HTWELL can be viewed for separate heater wells via \*OUTSRF \*SPECIAL \*HTRWELL. Accumulation of shut-in layers is included in a heater well's total accumulation.

Cell-based output is obtained from the PRN\_GRID list quantities CCHLOSS and CCHLOSSCUM which can be used with \*OUTPRN \*GRID, \*OUTSRF \*GRID and \*OUTSRF \*SPECIAL. However, these cell-based outputs are for all the heater options lumped together.

See the EXPLANATION for \*HEATR.

### Examples

#### Combustion Tube Heater

The following heater definition from template "sttst01.dat"

```
*TIME 0
...
    *HEATR *IJK 1 1 12 200 ** Turn on heater
*TIME .5
    *HEATR *CON 0 ** Shut off heater
...

```

can be converted to a heater well using the existing well "INJECTOR".

```
*TIME 0
...
    *PERF 'INJECTOR' ** i j k wi(gas)
        1 1 12 5.54
    *HTWELL 'INJECTOR' *HTWRATE 200 ** Turn on heater
*TIME .5
    *HTWELL 'INJECTOR' *OFF ** Shut off heater
...

```

#### Deviated Wellbore

Well "EDGE PRDCR", which is deviated from vertical, heats with a well temperature of 200 degrees and stops heating when the cell is hotter than the heater.

```
*LAYERXYZ 'EDGE PRDCR'
** block -- entry(x,y,z) --- -- exit(x,y,z) --- length
  5,5,1 131.2 131.2 1000. 131.2 131.2 1010. 10
  5,5,2 131.2 131.2 1010. 131.2 118.3 1030. 23.83
  5,4,3 131.2 116.6 1031. 131.2 116.6 1055. 23.75
  5,5,4 131.2 116.7 1055. 131.2 145.8 1080. 38.41

*HTWELL 'EDGE PRDCR' *HTWTEMP 200 *HTWRATE -1E-6
```

#### Cooling Well

Well "Cooling Well" has temperature of -10 F and maximum cooling rates that change several times during the run. This well has no fluid flow.

```
*TIME 0
  *WELL 'COOLING WELL' *VERT 9 1 *FRAC 0.125
  *PRODUCER 'Cooling Well'
  *GEOMETRY *K 0.3 0.249 1 0
  *PERFV *GEO 'Cooling Well'
    ** k   fh
      1  0.4  ** Partial completion
      2  1.0
      3  1.0
      4  0.62 ** Partial completion
  *SHUTIN 'Cooling Well'
  *HTWELL 'Cooling Well' *HTWTEMP -10 *HTWRATE -6.0E6
*TIME 2
  *HTWELL 'Cooling Well' *HTWTEMP -10 *HTWRATEPL -50000
*TIME 4
  *HTWELL 'Cooling Well' *HTWTEMP -10 *HTWRATE -2.0E6
```

---

## Wellbore Block Transmissibility Multipliers (Optional)

\*TRANSWB

### PURPOSE:

\*TRANSWB modifies the transmissibility multiplier of select connections between a discretized wellbore block and the parent block containing it.

### ARRAY:

\*TRANSWB (the \*IJK array reading option is allowed)

### DEFAULTS:

If \*TRANSWB is absent from a recurrent data segment, then these transmissibility multipliers remain unchanged. If \*TRANSWB is absent completely from the data, the multipliers are one.

Block-wellbore connections that are not referenced (e.g., when the \*IJK array reading option is used) remain unchanged.

### EXPLANATION:

Transmissibilities are calculated in the simulator using grid block dimensions, area modifiers, and permeabilities. Then the transmissibilities are multiplied with the entered modifiers and used in the flow equations.

Transmissibility multipliers are dimensionless and must not be negative.

Example:

Transmissibility multipliers \*TRANSI, \*TRANSJ and \*TRANSK are applied to wellbore flow. Array qualifiers \*RG, \*WELLBORE, \*TUBING and \*ANNULUS may be used.

Example:

A discretized wellbore is modelled from the surface to depth and then turns horizontal. Only the horizontal part is completed.

```
*WELLBORE 0.15          ** Horizontal well
*RANGE 1 1 1:9           - vertical (from surface)
                    1:4 1 1           - horizontal
** Perforate producer only in horizontal section
*TRANSWB *WELLBORE 1 1 2:9 *CON 0.0
```

See "Keywords from Other Sections" at the beginning of this chapter regarding fine-grid inheritance.

---

## **Pressure Dependent Transmissibility Multipliers**

\*PFRAC,  
\*PFRACF, \*PTRANSI, \*PTRANSJ, \*PTRANSK, \*PTRANSIJ+, \*PTRANSIJ-, \*PTRANSIK+,  
\*PTRANSIK-

### **PURPOSE:**

Assign pressure-dependent transmissibility multiplier parameters and locations.

### **ARRAY:**

\*PFRAC  
\*PFRACF  
\*PTRANSI  
\*PTRANSJ  
\*PTRANSK  
\*PTRANSIJ+  
\*PTRANSIJ-  
\*PTRANSIK+  
\*PTRANSIK-

### **DEFINITIONS:**

#### **\*PFRAC**

Lower reference pressure (kPa | psi), at which the fracture is practically closed. The suggested range is zero to the value given by \*PFRACF.

#### **\*PFRACF**

Upper reference pressure (kPa | psi), at which the fracture is practically opened. The suggested range is from the value given by \*PFRAC to 10 MPa.

#### **\*PTRANSI**

I-direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

#### **\*PTRANSJ**

J-direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

#### **\*PTRANSK**

K-direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

#### **\*PTRANSIJ+**

I+J+ diagonal direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

#### **\*PTRANSIJ-**

I+J- diagonal direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

**\*PTRANSIK+**

I+K+ diagonal direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

**\*PTRANSIK-**

I+K- diagonal direction variable transmissibility multiplier at maximum pressure effect. The suggested range is from zero to 1e5.

**DEFAULTS:**

By default the option is disabled. When the option is enabled, the default multipliers are 1.

**CONDITIONS:**

With the array-reading option \*IJK, referring to only select grid blocks is allowed.

**EXPLANATION:**

If the higher and lower reference pressures (\*PFRAC and \*PFRACF) are equal, then the variable multiplier calculation is not done.

The variable multiplying factor for transmissibility is

$$F = R + (1-R) * ptrans$$

where ptrans is one of \*PTRANSI, etc.. The ratio R is

$$R = 1 / (1 + \exp(x))$$

where x is the pressure dependent argument

$$x = 10 * (P - Pav) / (pfracf - pfrac)$$

Here P is the pressure in the grid block just upstream (with respect to water) of the block face in question, and Pav is the average of pfrac and pfracf in the same upstream block. When P = Pav then x = 0 and R = 0.5 corresponding to a midpoint effect of ptrans. Note that at low pressures R will be very nearly equal to 1, not exactly 1.

At pressure pfrac the variable multiplying factor is  $0.9933 + 0.0067 * ptrans$ . At pressure pfracf the variable multiplying factor is  $0.0067 + 0.9933 * ptrans$ . where PTMX is one of PTMRXI, PTMRXJ or PTMRXK. For example, if ptrans = 1000 then the variable part of the transmissibility multiplier is 993 at pressure pfracf.

Because P and Pav are taken from the upstream block, unexpected results may occur at the boundary between regions of different values of pfrac, pfracf and \*PTRANSI, etc.

**Inheritance for Refined Grids**

There is no fine-grid inheritance of connection-based quantities \*PTRANSx. Therefore, you must assign values explicitly to refined blocks instead of relying on them inheriting values from parent blocks.

**Recommendation**

The pressure-dependent transmissibility multiplier option is considered obsolete and may become unavailable in a future release. This option cannot be applied to some types of interblock connections, for example, hybrid grids, matrix-fracture and discretized wellbore. The full-feature variable permeability options (\*PERMCK, etc.) are recommended instead.

---

## **Automatic Rock-Fluid Switching**

**\*TEMLIM, \*KRNOPR, \*KRPRGRID, \*KRPRDET**

**\*KRSWITCH, \*KRRESET, \*SGLIM,**

### **PURPOSE:**

Specify data and printout for automatic rock-fluid switching.

### **FORMAT:**

<b>*KRSWITCH</b>	(*OFF   ikswch )
<b>*KRRESET</b>	ikreset
<b>*SGLIM</b>	sglm
<b>*TEMLIM</b>	temlim
<b>*KRNOPR</b>	
<b>*KRPRGRID</b>	
<b>*KRPRDET</b>	

### **DEFINITIONS:**

#### **\*OFF**

Prevents further switching from this point on in the simulation, or until  
**\*KRSWITCH** is encountered.

#### **ikswch**

Rock-fluid data is switched to this set number when the following switching criterion is satisfied:

( Sg > or = sglm ) and ( T > or = temlim )

The checking and switching is done just before the timestep is started. No switching is done during convergence.

#### **ikreset**

All grid blocks are reset to this rock-fluid set number.

#### **sglm**

Gas saturation limit. The allowed range is 0 to 2. A zero value makes switching independent of Sg, and a value of 2 prevents switching.

#### **temlim**

Temperature limit (C | F). The allowed range is 0 to 2000 K. A zero value makes switching independent of temperature. A value larger than the highest expected temperature prevents switching.

#### **\*KRNOPR**

Printout for this option is turned off.

**\*KRPRGRID**

Rock-fluid data set number for the entire grid is printed at the same time as the other reservoir properties.

**\*KRPRDET**

In addition to the printout caused by \*KRPRGRID, a message is printed when a grid block switches data set number due to the criteria.

**DEFUALTS:**

\*KRSWITCH \*OFF

\*KRNOPR

\*SGLIM 0

\*TEMLIM 0

---

## **Reset Adaptive Implicit**

**\*AIMSET**

### **PURPOSE:**

Over-ride automatic adaptive implicit switching.

### **ARRAY:**

**\*AIMSET**

### **DEFINITIONS:**

The value assigned to a block via \*AIMSET causes that block to assume an implicitness state given by

- 0 - IMPES
- 1 - Fully Implicit

If the \*STAB option is being used, then \*AIMSET acts like an implicitness mask: each block flagged as fully implicit will never be allowed to switch to IMPES.

If the \*THRESH option is being used, this is the only way to switch a block from fully implicit to IMPES since the threshold criteria has no method for switching automatically from implicit to IMPES.

### **DEFAULTS:**

If \*AIMSET is absent, the distribution of IMPES and fully implicit blocks is not over-ridden.

### **CONDITIONS:**

This keyword is useful only if an adaptive implicit switching option has been specified via \*AIM.

### **EXPLANATION:**

Example:

```
** With stability adaptive implicit, keep all blocks
   fully implicit in a communication path.
.
.
.
*AIM *STAB
.
.
.
** Recurrent data
*AIMSET *CON 0
   *MOD 3:4 8:9 3:3 = 1      ** Communication path
```

See Appendix F.9 for a detailed discussion of the Adaptive Implicit option.

## Dynamic Grid Amalgamation Control (Optional)

\*DYNAGRID

### PURPOSE:

Create, or remove, amalgamated grid cells and refinements statically, or under simulator control.

### FORMAT:

\*DYNAGRID \*AMALGAMATE ( *control\_list* ) ( \*EVEN-IF-CMPLX )  
\*INTO *nir njr nkr*

{ *i1:i2 j1:j2 k1:k2* }

\*DYNAGRID \*DEREFINE

{ *i1:i2 j1:j2 k1:k2* }

\*DYNAGRID \*DEAMALGAMATE

{ *i1:i2 j1:j2 k1:k2* }

\*DYNAGRID \*REREFINE

{ *i1:i2 j1:j2 k1:k2* }

where

*control\_list* =

( \*SATUR *dsn* | \*SATWAT *dsw*  
| \*SATOIL *dso* | \*SATGAS *dsg*  
| \*TEMPER *dtn* | \*GMOLAR *dzn*  
| \*GMOFRC *dgn* | \*OMOFRC *don*  
| \*WMOFRC *dwn* | \*PRESS *prs*  
| \*ENTHAL *den* )

\*DYNAGRID-TSINT *tsint*

\*DYNAGRID-TINT *tint*

\*DYNAGRID-IGN-DIR (\*IDIR | \*JDIR | \*KDIR | \*IJ | \*IK | \*JK )

\*DYNAGRID-WLN-A *iwlna*

\*DYNAGRID-WLN-V *iwlnv*

\*DYNAGRID-PICK-TYPE (\*MIN | \*MAX | \*1ST )

### DEFINITIONS:

\*AMALGAMATE ( *control\_list* ) ( \*EVEN-IF-CMPLX ) \*INTO *nir njr nkr*

Over the specified cell region, amalgamate each regular group of *nir* × *njr* × *nkr* cells together into a single cell. Smaller groups near the edges of the region are amalgamated if a whole number of the requested amalgamations cannot be made to fit.

If one or more keywords from *control\_list* appear, then amalgamation (or de-amalgamation) occurs dynamically, triggered by the degree of uniformity of properties specified by the *control\_list* keywords being satisfied (or not) by the cells in and around each group.

If keyword \*EVEN-IF-CMPLX appears then amalgamation is permitted in the specified cell groups even when complex connections are present. See **Complex Connections**, below.

## **\*DEREFINE ( *control\_list* )**

De-activate a refinement created by \*REFINE, for the specified parent cells. This keyword causes all the child cells in each specified cell to be replaced with the corresponding fine grid's single parent cell, effectively reversing the refinement process.

If one or more keywords from *control\_list* appear, then de-refinement (or re-refinement) occurs dynamically, triggered by the degree of uniformity of properties specified by the *control\_list* keywords being satisfied (or not) by the cells in and around each group.

## **\*DEAMALGAMATE**

Return a previously amalgamated cell range to its constituent cells, 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 cells.

{ *i1:i2 j1:j2 k1:k2* }

A list of triples of cell I-J-K ranges, one per line, specifying groups of fundamental cells. There must be at least one I-J-K range. See the Keyword Data Entry System section (for instance, the discussion of \*IJK) for how ranges work.

*control\_list*

One or more of the following tolerance specifiers.

*SATUR <i>dsn</i>	Any fluid phase saturation
*SATWAT <i>dsw</i>	Water phase saturation
*SATOIL <i>dso</i>	Oil phase saturation
*SATGAS <i>dsg</i>	Gas phase saturation
*TEMPER <i>dtn</i>	Temperature (C deg   F deg)
*GMOFRC <i>dgn</i>	Gas phase fraction
*OMOFRC <i>don</i>	Oil phase fraction
*WMOFRC <i>dwn</i>	Water phase fraction
*GMOLAR <i>dzn</i>	global mole fraction
*PRESS <i>prs</i>	pressure (kPa   psi   kPa)
*ENTHAL <i>den</i>	fluid enthalpy

## **\*DYNAGRID-TSINT *tsint***

Specify a minimum number of timesteps *tsint* that should exist between checks for grid changes due to dynamic amalgamation and de-refinement. Specifically, such checks are done before timesteps whose number minus 1 is evenly divisible by *tsint*. For example, if *tsint* = 5 then grid change checks are done before timestep numbers 1, 6, 11, 16, 21, etc. Smaller values of *tsint* will cause a larger

fraction of the total run time to be spent in grid change checking but may result in overall CPU savings and/or increased accuracy. On the other hand, very small values (e.g.,  $tsint = 1$ ) may not produce the optimal combination of CPU and accuracy, so some calibration may be beneficial.

**\*DYNAGRID-TINT *tint***

Specify a minimum interval of time *tint* (days) that should exist between checks for grid changes due to dynamic amalgamation and de-refinement.

\*DYNAGRID-TSINT (timestep base) is recommended over this option (time base) since timesteps more successfully track the rate of change of process variables.

**\*DYNAGRID-IGN-DIR ( \*IDIR | \*JDIR | \*KDIR | \*IJ | \*IK | \*JK )**

Specify a direction or plane that will be ignored when checking for amalgamations. \*KDIR is often a useful choice.

*IDIR	Ignore I direction	(check in J-K plane only)
*JDIR	Ignore J direction	(check in I-K plane only)
*KDIR	Ignore K direction	(check in I-J plane only)
*IJ	Ignore I-J plane	(check in K direction only)
*IK	Ignore I-K plane	(check in J direction only)
*JK	Ignore J-K plane	(check in I direction only)

**\*DYNAGRID-WLN-A *iwlna***

Specify how far away from active wells in the areal directions to protect from dynamic amalgamation and de-refinement checks. Integer *iwlna* refers to counting on the finest grid.

**\*DYNAGRID-WLN-V *iwlsv***

Specify how far away from active wells in the vertical direction to protect from dynamic amalgamation and de-refinement checks. Integer *iwlsv* refers to counting on the finest grid.

**\*DYNAGRID-PICK-TYPE *ctype***

Specify the rock type for the coarse cell that results from the amalgamation of fine cells. This keyword is needed only when an amalgamation region contains fine cells possessing different rock types. The choices for *ctype* are:

*MIN	Minimum of rock type numbers in cell group
*MAX	Maximum of rock type numbers in cell group
*1ST	Rock type of first cell in group when cells are ordered by increasing I, then J, then K address

**DEFUALTS:**

All these keywords are optional.

If there are no keywords from *control\_list* following \*AMALGAMATE or \*DEREFINE, the corresponding action is done immediately, that is, at the current (reading) time.

If the keyword \*EVEN-IF-CMPLX is absent, then amalgamation is not done for any cell group that contains a complex connection.

If \*DYNAGRID-TSINT and \*DYNAGRID-TINT are absent and grid checking is dynamic, then grid checking is done each timestep ( $tsint = 1$ ).

If \*DYNAGRID-PICK-TYPE is absent, then \*MIN is assumed.

#### **CONDITIONS:**

None of these keywords may be used with any of the following grid options in the Reservoir Description section: discretized wellbore (\*WELBORE), natural fracture (\*DUALPOR, DUALPERM, \*SUBDOMAIN and \*MINC) and nine-point discretization (\*NINEPOINT).

Keywords \*DYNAGRID-TINT, \*DYNAGRID-TSINT, \*DYNAGRID-IGN-DIR, \*DYNAGRID-WLN-A and \*DYNAGRID-WLN-V are applicable only for dynamic gridding.

If \*DEREFINE is used, then keyword \*REFINE in the Reservoir Description data section must have been used to form the refinement.

In a given recurrent data segment, any \*DYNAGRID keywords should appear at the end of that segment, following any well specification keywords.

#### **EXPLANATION:**

The recurrent gridding features are designed to save computing time during a simulation by reducing the number of grid cells used in the model. The keywords described here can alter the grid in both a static or dynamic manner. If keywords from *control\_list* do not appear, then the grid alterations are static, in that they are applied immediately at a time specified by a \*DATE or \*TIME keyword. If keywords from *control\_list* do appear, the changes are dynamic and depend on the specified trigger criteria interacting with the changing conditions in the specified cell ranges.

#### **Amalgamation and De-refinement**

The \*AMALGAMATE keyword causes specified groups of cells, described by the triples of cell range descriptors listed after the keyword, to be lumped together into larger cells for further computation. When this lumping is done, new cell properties are assigned to the amalgamated cells in such a way as to preserve the total mass of the simulator's components and energy, and to provide appropriate averaged properties for the new cell. The extended variables are then computed accordingly. Inter-cell connections are done differently (see below). Amalgamation in and around wells and fault-type connections is not recommended (see below).

The \*DEREFINE subkeyword is similar to the \*AMALGAMATION subkeyword, although the ranges following this keyword are interpreted in a slightly different way. Each cell referred to in the list of ranges that is a parent cell (and hence was refined earlier using the \*REFINE keyword) will be reformed back into its original state, just as if an amalgamation had been applied to the group consisting of all the child cells in the refinement. The parent cell thus becomes active, acting as if it was never refined, and the child cells are ignored in the on-going simulation.

The \*DEAMALGAMATE, or \*REREFINE, subkeywords cancel the actions of earlier encountered \*AMALGAMATE, or \*DEREFINE, subkeywords, respectively. Ranges following these keywords need only partially overlap earlier ranges to cause the cancellations to occur.

Note that if a group of cells on a grid were specified for \*AMALGAMATE, and one or more of these cells had themselves been refined at earlier times, then amalgamation would still proceed. What would happen is that all refinements of any child cells in the region would be averaged up to their parents (using primarily volume weighted averaging as described earlier), after which the final group of cells would be averaged up as described above. Similarly, if the parent of a group of child cells was selected for ‘\*DEREFINE’, and some or all of those cells were themselves refined, then averaging would be done throughout all refinement levels until the parent described by the cell descriptor for the \*DEREFINE keyword had been reformed.

### Dynamic Trigger Criteria

The *control\_list* specifies what level of activity in which properties will be tolerated before cells are amalgamated or de-amalgamated. At a time when the amalgamation state of a cell (or group of cells) is to be evaluated, the values of the properties are examined. For a range of cells described in the \*AMALGAMATE keyword’s list, if values of each property for the cells and its neighbours are the same within that property’s specified threshold, that group will be marked for amalgamation. For example, if \*TEMPER 1.0 is used and a cell group and neighbours has temperatures all within 0.5 of each other, then that cell group would be amalgamated. On the other hand, if a group of cells had passed these tests previously and were amalgamated under the \*AMALGAMATE keyword, that amalgamated cell’s value would be checked against its neighbours using the thresholds. If the difference of cell’s value and its neighbours lie outside of the specified threshold(s), then the group will be marked for de-amalgamation. Similar calculations are done in the case of \*DEREFINE. Thus, groups of cells will be amalgamated and de-amalgamated dynamically, depending on grid activity.

Dynamic grid keywords are first applied at the time specified by the \*TIME or \*DATE keyword at which they appear. However, dynamic amalgamation or de-refinement will be reconsidered each timestep thereafter, except as controlled by the \*DYNAGRID-TSINT and \*DYNAGRID-TINT qualifiers.

### Addressing Grid Cells

All references made in the data set to cells, such as those made for grid property input, initialization, well perforations, etc., must always be made with respect to the finest grid, which is the grid resulting from the removal of all \*DYNAGRID data. In fact, if all \*DYNAGRID keywords were removed then the data set would (and must) still be a valid data set and would run a normal simulation. The simulator makes all the internal changes required to run a simulation with amalgamated cells and does not allow the user to address the amalgamated cells directly.

### Interblock Connections

Flow connections between cells constructed by \*AMALGAMATE or \*DEREFINE are built up from the flow connections calculated on the finest grid, that is, from the usual connections that are constructed when no amalgamations or de-refinements of any type are present. When a connection between an amalgamated cell and some other cell (amalgamated or not) needs to be calculated, the connection is built by summing over the connection pairs between finest grid cells, where one member of the pair is in the amalgamated region and the other member is the other cell (or a member of its amalgamation, if there is one). The overlap areas (which incorporate transmissibility multipliers, if present) are summed and an overall permeability is formed by taking an area-weighted sum of the harmonically-averaged permeabilities that come from the fine-scale connections. An overall transmissibility is then computed by multiplying

the overall permeability by the summed areas, and dividing by the cell centre-to-centre distance for the cells involved. Each cell's centre-to-face distance comes directly from the finest grid for an un-amalgamated cell, and otherwise is constructed when other property construction takes place for amalgamated cells.

Note that under the current flow connection regime described here, the only permeabilities that matter are those that were assigned to the fine grid cells on the boundaries of the amalgamated region (as these enter into the only fine grid transmissibilities that were used to build the new connection). This means that flow barriers (zero permeabilities and zero transmissibilities) strictly within amalgamated regions will not affect the overall flow. Thus, amalgamated regions should consist of relatively homogeneous cells. Obviously, all the cells in an amalgamated region should share the same rock types, etc., to avoid any deviations from the original settings.

### Complex Connections

By default a cell group is not amalgamated if it contains complex fault-type connections, that is, certain faces of some cells have multiple, or no, connections to their neighbours. If used, the keyword \*EVEN-IF-CMPLX must be used with care. As described above, the only permeabilities that matter for amalgamated regions are those that were assigned to the fine grid cells on the boundaries of the amalgamated region. Consequently, if these keywords are used then a flow barrier entirely inside an amalgamated cell effectively disappears.

### Active Wells

A cell group is not allowed to amalgamate if it contains at least one well block, whether or not the well is shut in. This rule is applied not only at the beginning of the run but at any time a new well perforation is specified in a block. Therefore, if a new perforation is specified later in the run in a block in a cell group that is amalgamated, the cell group is de-amalgamated immediately.

In addition, amalgamation too near a cell with an active well makes it difficult, if not impossible, to match the results of original fine grids for many simulations. Keywords \*DYNAGRID-WLN-A and \*DYNAGRID-WLN-V let you control how near to an active well amalgamation is allowed.

### Property Models and Grid Changes

A property model is a mathematical function that relates a physical property (e.g., density, viscosity, relative permeabilities) to reservoir conditions (e.g., pressure, temperature, phase saturations). The application of a property model to a number of cells and timesteps is called homogeneous when the same function is used and heterogeneous when a different function is used. A property model can be applied heterogeneously only if some of its parameters are entered via (1) grid arrays (per-cell values) or (2) rock types (per-cell-region values). Spatial heterogeneity is usually due to geology, that is, different rock layers may have different properties.

Amalgamation of a cell group into one coarse cell replaces those cells' property models with the coarse cell's property model. If a property model is homogeneous in that cell group then the coarse cell will have the same property model as the cell group, which lends a degree of consistency to grid changes. When a coarse cell is split into fine cells, each fine cell inherits the coarse cell's reservoir conditions; if a property model is homogeneous in that fine cell

group then the resulting property values will be homogeneous as well, which corresponds most consistently to the coarse cell representation.

In a practical amalgamation case, conditions differ somewhat but not greatly between fine cells according to tolerances in the *control\_list*. If the fine cells have the same property model then the coarse cell solution will be close to all the fine cell solutions, resulting in a well-behaved amalgamation. On the other hand, if the fine cells have different property models, the coarse cell solution will be arbitrarily different from the fine cell solutions merely because of a property model change, which is undesirable behavior.

### **In-Place Amounts**

To ensure that solutions of the conservation equations are physical and consistent, property models that contribute to the amounts of flowing components and energy “in-place” ( $K$  values, phase densities and phase enthalpies) must be homogeneous in time. In STARS the above property models are applied homogeneously so amalgamation of a cell group will not lead to change in these property models.

Adsorption also contributes to “in-place” amounts, but the adsorption property model can be applied heterogeneously with keywords \*ADSROCK and \*ADSTYPE. To maintain consistency of adsorbed amounts upon amalgamation or refinement, the adsorption property model must be applied homogeneously inside each cell group flagged for amalgamation.

Porosity contributes to “in-place” amounts by providing the pore volume  $V \cdot \phi_r \cdot f(p, T)$  where  $V$  is the gross cell volume,  $\phi_r$  is the reference porosity entered via \*POR and  $f(p, T)$  gives the pressure and temperature dependence entered via keywords \*CPOR, etc. The property model for  $f(p, T)$  can be applied heterogeneously with keywords \*ROCKTYPE and \*THTYPE. This expression for pore volume applies to both a coarse cell and each of its corresponding fine cells. Internally, the coarse cell  $\phi_r$  is given a value such that  $V \cdot \phi_r$  for the coarse cell is equal to the sum of the fine cell values, that is,  $V \cdot \phi_r$  is conserved upon a grid change. To conserve the current pore volume  $V \cdot \phi_r \cdot f(p, T)$  as well, the property model for  $f(p, T)$  must be applied homogeneously inside each cell group flagged for amalgamation.

Porosity parameters and rock heat capacity in the \*ROCKTYPE keyword group can be changed in recurrent data but the manual recommends that only compressibility be changed. In fact, manually changing rock types in recurrent data is no longer recommended and has been replaced by options made specifically for compaction and dilation hysteresis.

### **Heterogeneous Property Models**

Heterogeneities in properties that do not contribute to “in-place” amounts behave more like varying well rates; these do not present the same consistency problems and so are more acceptable. Other type-capable properties which affect only the flow between cells include rock-fluid (\*RPT), variable permeability (\*ROCKTYPE) and viscosity (\*VISCTYPE). Any different \*ROCKTYPE data types intended for amalgamation should differ only in the variable permeability and thermal conductivity data.

Even for flow-based properties amalgamation can be problematic when fine cell models differ significantly in response to the same conditions, leading to ill defined and arbitrary results. When fine cells in an amalgamation group have different relative permeability rock types (\*KRTYPE), there is no explicit method to choose any one rock type for the coarse cell. Especially problematic are the per-cell relative permeability end-points (e.g., \*BSWR). The coarse cell’s volume-weighted average probably does not reflect the flowing behavior

that a more rigorous upscaling technique would provide. Also, the averaged value is used for all flow directions, which again may not be appropriate.

Per-cell absolute permeabilities (\*PERMI/J/K) can vary between fine cells; the coarse cell value may be a suitable average but does not contain blocking information not found in the first plane of fine cells (see **Interblock Connections**, above).

Therefore, if significant rock-type heterogeneity is important to capture the desired physics, a cell region containing such heterogeneity should not be in the \*DYNAGRID cell list. When the degree of heterogeneity is not great and dynamic gridding is specified for those regions, you can use keyword \*DYNAGRID-PICK-TYPE to select one of the rock types for the coarse cell, but with caution.

### Properties with Hysteresis

Even if the fine cells in a region possess the same rock type, the determination of some properties of the region is still problematic when hysteresis is involved. Specifically, a cell property depends not only on the current value but also the history of the independent variable. If the histories of a group of fine cells deviated significantly from each other, it is uncertain what history the coarse cell should assume upon amalgamation of that cell group.

The remedy is to avoid such deviation by specifying a lower amalgamation tolerance value for the independent variable via the *control\_list*, ensuring that the histories of that variable in the fine cells are not far apart upon amalgamation. This will result in somewhat less amalgamation overall, but it will greatly increase the accuracy of simulations involving hysteresis. Amalgamation of cells with different hysteresis definitions is not recommended.

Property	*DYNAGRID Control
*DILATION	*PRESS
*HYS_KRO	*SATUR
*HYS_KRW	*SATUR
*HYS_KRG	*SATUR
*HYS_PCOW	*SATUR
*HYS_PCOG	*SATUR

### Amalgamation versus Upscaling

In general, the \*DYNAGRID amalgamation option is not a substitute for a competent upscaling facility. However, it would be possible to use upscaling to obtain flow properties of the coarser coarse cell and then apply that single flow-based data type also to its fine cells. The fine grid with coarse properties would afford the benefits of smaller cells (better gradients, sharper fronts) but perhaps would not reproduce all the characteristics of the fine-property case.

### Viewing Dynamic Grid Changes

Results 3D lets you view grid changes when a property plot is advanced in time.

A property plot in Results is affected by the mode with which the grid data was written to the SR2. For mode switch \*DYNSR2MODE in the Input/Output Control chapter the default mode \*STATIC gives the best result. In this mode, data is saved in the form of a static fine grid that does not change with time. Fine cells that are amalgamated in the simulator computation have the same value and hence plot with the same colour. Additional information allows Results 3D to leave out boundaries between amalgamated fine cells, giving

the appearance of a single large cell. In addition, the probe indicates property values using the fine-grid address, and special histories and Quick Plot X-Y plots are available for each fine cell.

Grid changes may happen at times between adjacent grid dump times. Consequently, a grid change that appears first at a certain plot time may have happened any time since the previous grid dump time. Keyword \*DYNGRDFREQ in the Input/Output Control chapter can be used to specify additional grid dumps to show more frequent grid changes. Production runs usually specify grid dumps sparingly via keyword \*WSRF \*GRID in order to control the size of output files. This corresponds to default \*DYNGRDFREQ 0 which indicates that no additional grid dumps are done. On the other hand, a run with \*DYNGRDFREQ 1 and only selected grid properties could be used to generate output for a presentation that shows all grid changes when they happen in the computation.

### **Removing a Discretized Wellbore**

A discretized wellbore may be removed in the Recurrent data with \*DYNAGRID \*DEREFINE.

### **Starting Dynamic Gridding Later in Run**

Generally, dynamic gridding starts when a \*DYNAGRID keyword appears in the recurrent data. Like other recurrent data items, that keyword takes effect at the simulation time specified by the immediately preceding \*TIME or \*DATE keyword. Therefore, if \*DYNAGRID appears in the first segment of recurrent data, dynamic gridding takes effect from the start of the run (timestep #1). However, you are allowed to specify the first \*DYNAGRID keyword in a later segment of recurrent data, in which case dynamic gridding is not in effect until the indicated simulation time.

Note that the possibility of dynamic gridding must be known at the beginning of a multi-run simulation. Therefore, you are not allowed to add dynamic gridding to a restart run whose parent run did not have \*DYNAGRID. For example, suppose you have a completed parent run whose data goes only to 100 days. If you wish to restart from the 100 day point and simulate until 200 days, the restart run's recurrent data must be extended to the 200 day point. If the parent data (0-100 days) has \*DYNAGRID then that keyword is allowed also in the restart run's extended recurrent data (100-200 days). However, if the parent run did not have \*DYNAGRID then that keyword may not be used in the restart run.

---

## **Discretized Wellbore in Recurrent Data (Conditional)**

\*WELBORE-REC, \*RELROUGH, \*LAMINAR, \*TRANSIENT, \*CIRCWELL, \*WELLINFO,  
\*REGIME, \*WELLWALL, \*TUBINSUL, \*ANNULUSWAL, \*CASING, \*FILM\_COND, \*RANGE

### **PURPOSE:**

Define wells which are to be discretized anytime in the recurrent data.

### **FORMAT:**

```
*WELBORE-REC rw (*RELROUGH relrof)
*LAMINAR
*TRANSIENT (*ON | *OFF))
*CIRCWELL ra i j k nwbwt (*RELROUGH relrof)
*WELLINFO
*REGIME

*WELLWALL      rwo      hcww
*TUBINSUL      rins     hcins    nwbwin
*ANNULUSWAL    rao      hcaw
*CASING        rcas     hccas    nwbwca
*FILM_COND

*RANGE          i1(:i2)  j1(:j2)  k1(:k2)
               ( i1(:i2)  j1(:j2)  k1(:k2) )
```

### **DEFINITIONS:**

See \*WELBORE in Reservoir Description Section.

### **DEFAULTS:**

See \*WELBORE in Reservoir Description Section.

### **CONDITIONS:**

See \*WELBORE in Reservoir Description Section.

If \*WELBORE is present, then \*RANGE must be present also.

### **EXPLANATION:**

See \*WELBORE in Reservoir Description Section.

The only difference between specifying a discretized wellbore (DW) in Reservoir Description and Recurrent data is in the assignment of relative permeability curves. When DW is specified in Reservoir Description and relative permeability is not specified via \*KRTYPE then STARS will automatically assign straight line relative permeability curves.

When DW is specified in Recurrent data and relative permeability is not specified via \*KRTYPE then the DW will inherit the relative permeability curves of the block where it is located. The same is valid for \*THTYPE.

DW may be removed with the same \*DYNAGRID \*DEREFINE keywords as any other reservoir block.

When DW needs to be altered, e.g., tubing is pulled, then remove the existing DW with the \*DYNAGRID \*DEREFINE keyword and specify a new DW with the \*WELLBORE-REC keywords.

## Electrical Heating Boundaries (Conditional)

\*ELBOUND,

\*ELTARGET

### PURPOSE:

Specify electrical heating boundaries and operating conditions.

### FORMAT:

```
*ELBOUND 'name' uba_range (dir)
          uba_range (dir)
          ...
*ELTARGET      *POTENTIAL      'name' VRmax ( VImax )
*ELTARGET      *CURRENT        'name' Imax
*ELTARGET      *POWER          Qmax
*ELTARGET      *NOFLASH        ΔT
```

### DEFINITIONS:

\*ELBOUND 'name' uba\_range (dir)

String 'name' is the name of the boundary in quotes. Characters after the first 12 are ignored.

*uba\_range* is the address of a single block or range of blocks in UBA format. See section **User Block Address** in chapter Keyword Data Entry System. A range is allowed in any number of directions at any grid level. Reference to a null block, a zero-porosity block (if that block's rock conductivity is zero for any direction or temperature) or a refined parent block will result in a warning message and that block will be skipped. If an electrical boundary is attached to the inside of the inner-most ring of a hybrid-type locally refined block, be sure to specify a non-zero value for the well radius via \*REFINE \*HYBRID \*RW (the default value of zero represents a zero area for electrical conduction).

*dir* is the optional boundary face specifier. Allowed values are subkeywords -I, +I, -J, +J, -K and +K. This specifies exactly the face of the host block(s).

Here, + or - refers to the face between the host block and its neighbour with an index in the indicated direction (I, J or K) that is one lower (-) or one higher (+). For example, to access the face between blocks (i,j,k) and (i+1,j,k) use +I from (i,j,k) or use -I from (i+1,j,k).

Multiple sets of "uba\_range (dir)" may be specified as well, one per line.

For example, in a 13x1x4 grid with \*KDIR \*UP the entire top of the grid is indicated by the range "1:13 1 4" with direction specifier "+K". This is a case where the default algorithm (see below) will choose the default.

Normally *dir* is used to indicate a block face on the outer reservoir boundary, but an interior block face may be specified.

**\*ELTARGET \*POTENTIAL 'name'  $V_{Rmax}$  (  $V_{Imax}$  )**

Initialize potential (V) for boundary 'name' to  $V_{Rmax}$  (+  $\mathbf{j}V_{Imax}$  if multi-phase) and do not let it exceed that value. For alternating potential cases, these are root-mean-squared potentials. The potential specified for any boundary is independent of any other boundary. For a boundary consisting of multiple block faces, each of those faces is assigned the specified potential. It is recommended that at least one electrical boundary be assigned as ground ( $V_{Rmax} = 0$ , and  $V_{Imax} = 0$  for multi-phase) when \*POWER or \*NOFLASH constraint types are used.

**\*ELTARGET \*CURRENT 'name'  $I_{max}$**

Enforce a maximum current (A) constraint of  $I_{max}$  for boundary 'name'. For multi-phase runs this is the maximum magnitude of the current vector. The maximum current specified for any boundary is independent of any other boundary. For a boundary consisting of multiple block faces, the total current summed over the faces is exactly  $I_{max}$  but its distribution amongst those faces is updated only when the potential field is updated. A \*CURRENT constraint may not be applied to a boundary with  $V_{Rmax} = 0$  (and  $V_{Imax} = 0$  for multi-phase). Note that \*CURRENT constraints may not converge satisfactorily with larger values of \*EHEATCYC.

**\*ELTARGET \*POWER  $Q_{max}$**

Enforce a maximum power (electrical heating rate in kW) constraint of  $Q_{max}$  for the entire grid as a whole. The potential level of the entire grid is adjusted so that the total heat generation rate does not exceed  $Q_{max}$ .

**\*ELTARGET \*NOFLASH  $\Delta T$**

Adjust the potential level of the entire grid so that the temperature in any block  $i$  does not exceed  $T_{target} = T_{sat}(p_i) - \Delta T$ , where  $T_{sat}(p_i)$  is the saturation temperature of water at block pressure  $p_i$ . Unit of  $\Delta T$  is (C deg | F deg). This is done by calculating for each block  $i$  a power reduction factor

$$r_i = 1 - dQ_i / Q_i$$

where  $Q_i$  is the present heating rate of block  $i$ . The excess heating rate is

$$dQ_i = ( T_i - T_{target} ) * C_{pi} / \Delta t$$

where  $T_i$  is block temperature,  $C_{pi}$  is block heat capacity and  $\Delta t$  is the timestep size. The minimum  $r_i$  over all the blocks is chosen.

Constraint type \*NOFLASH can control the temperature only approximately, so some experimentation may be necessary to obtain an effective and stable result. This option is intended to be used in a situation where heat is being continuously removed, usually by the influx of colder fluid, whereby a quasi-static  $Q_i$  results from the balance with heat loss.

## **DEFAULTS:**

The algorithm for the \*ELBOUND *dir* default is to choose the unique face in the direction toward the reservoir boundary that is common to all blocks in the indicated range. This algorithm fails when there is no range, or more than one face is common to the range. For a grid direction that is not discretized (has 1 block) an index of 1 is deemed to specify a range. You are notified if the default algorithm yields no result.

## **CONDITIONS:**

Any \*ELTARGET with a boundary name must appear after the first \*ELBOUND defining that name.

Each electrical boundary must have a specified \*POTENTIAL constraint, which provides also an initial potential distribution as well as the phase in multi-phase mode.

## **EXPLANATION:**

If keyword \*ELECHEAT is present in the Other Reservoir Properties data section, then you must specify at least one electric boundary via \*ELBOUND and \*ELTARGET. Any current flow and resistive heating requires at least two electrical boundaries at two different potentials.

Any and all combinations of these constraints are allowed.

Boundary values and types can be changed at any time in recurrent data.

See Appendix G for a detailed description of the electrical heating option.

### **Multi-phase Mode**

If  $V_{lmax}$  appears at least once in a data set, that data is run in multi-phase mode; otherwise, it is run in single-phase mode. If a data set is running in multi-phase mode and  $V_{lmax}$  is absent for a well,  $V_{lmax} = 0$  is assumed. In multi-phase mode  $V_{Rmax}$  and  $V_{lmax}$  determine the phase and initial magnitude of a boundary's potential. Other constraint types (e.g., \*CURRENT) may change the magnitude of a boundary's potential, but the boundary's phase is not changed until subsequent \*POTENTIAL data is specified.

See Appendix G "Electrical Heating".

### **Examples:**

```
** Change electrode voltage during run
*TIME 0
  *ELBOUND 'Electrode' 1 1 1:6 -i
  *ELTARGET *POTENTIAL 'Electrode' 220
  *ELTARGET *POWER 25
*TIME 10
  *ELTARGET *POTENTIAL 'Electrode' 160
  *ELTARGET *CURRENT    'Electrode' 40
** Hybrid grid
*ELBOUND 'Electrode' 4 4 3:5 / 1 1 1:2 -i
** UBA list cannot be specified with single range set
```

```
*ELBOUND 'Electrode' 4 4 3 / 1 1 1 -i
4 4 3 / 1 1 2 -i
4 4 5 / 1 1 1 -I    ** skip k = 4
4 4 5 / 1 1 2 -i

** Three-phase 220 V configuration
*ELTARGET *POTENTIAL 'E1' 220 0      ** 0 deg
*ELTARGET *POTENTIAL 'E2' -110 +190.5 ** 120 deg
*ELTARGET *POTENTIAL 'E3' -110 -190.5 ** 240 deg
*ELTARGET *POWER 55
```

# Tables

**Table 1: Ordering of Components**

COMPONENT		MAY EXIST IN THIS PHASE				
Number	Type	Water	Oil	Gas	Adsorbed	Solid
1	Aqueous	X	X	X	X	
.	"	X	X	X	X	
NUMW	"	X	X	X	X	
NUMW+1	Oleic	X	X	X	X	
.	"	X	X	X	X	
NUMX	"	X	X	X	X	
NUMX+1	Non-condensable			X	X	
.	"			X	X	
NUMY	"			X	X	
NUMY+1	Solid					X
.	"					X
NCOMP	"					X

**Table 2: K-Value Coefficients for Selected Components**

$$K = (KV1 / P) * \text{EXP}( KV4 / (T - KV5) )$$

where KV1, KV4 and KV5 correspond to the units of p and T. from Appendix A "The Properties of Gases and Liquids", third edition, R.C. Reid, J.M. Prausnitz and T.K Sherwood, McGraw-Hill, Inc., 1977.

		<b>KV1</b>		<b>KV4</b>	
<b>Component</b>	<b>(kPa)</b>	<b>(psi)</b>	<b>(atm)</b>	<b>(K,C)</b>	<b>(F,R)</b>
H2O	1.1860e+7	1.7202e+6	1.1705e+5	-3816.44	-6869.59
H2S	1.3147e+6	1.9068e+5	1.2975e+4	-1768.69	-3183.64
N2	4.1636e+5	6.0388e+4	4.1091e+3	-588.72	-1059.70
O2	6.5514e+5	9.5020e+4	6.4657e+3	-734.55	-1322.19
CO	2.3182e+5	3.3622e+4	2.2878e+3	-530.22	-954.40
CO2	8.6212e+8	1.2504e+8	8.5084e+6	-3103.39	-5586.10
CH4	5.4547e+5	7.9114e+4	5.3834e+3	-879.84	-1583.71
C2H6	8.4644e+5	1.2277e+5	8.3538e+3	-1511.42	-2720.56
C3H8	9.0085e+5	1.3066e+5	8.8907e+3	-1872.46	-3370.43
C4H10	8.5881e+5	1.2456e+5	8.4758e+3	-2154.90	-3878.82
C5H12	1.0029e+6	1.4546e+5	9.8978e+3	-2477.07	-4458.73
C6H14	1.0062e+6	1.4594e+5	9.9305e+3	-2697.55	-4855.59
C7H16	1.0442e+6	1.5145e+5	1.0306e+4	-2911.32	-5240.38
C8H18	1.1187e+6	1.6226e+5	1.1041e+4	-3120.29	-5616.52
C9H20	1.1465e+6	1.6628e+5	1.1315e+4	-3291.45	-5924.61
C10H22	1.1984e+6	1.7381e+5	1.1827e+4	-3456.80	-6222.24
C12H26	1.3271e+6	1.9248e+5	1.3097e+4	-3774.56	-6794.21
C15H32	1.4077e+6	2.0418e+5	1.3893e+4	-4121.51	-7418.72
C17H36	1.3779e+6	1.9985e+5	1.3599e+4	-4294.55	-7730.19
C18H38	1.3402e+6	1.9437e+5	1.3226e+4	-4361.79	-7851.22
C20H42	1.8929e+6	2.7454e+5	1.8681e+4	-4680.46	-8424.83

**KV5**

<b>Component</b>	<b>(deg K)</b>	<b>(deg C)</b>	<b>(deg F)</b>	<b>(deg R)</b>
H2O	46.13	-227.02	-376.64	83.03
H2S	26.06	-247.09	-412.76	46.91
N2	6.60	-266.55	-447.79	11.88
O2	6.45	-266.70	-448.06	11.61
CO	13.15	-260.00	-436.00	23.67
CO2	0.16	-272.99	-459.38	0.29
CH4	7.16	-265.99	-446.78	12.89
C2H6	17.16	-255.99	-428.78	30.89
C3H8	25.16	-247.99	-414.38	45.29
C4H10	34.42	-238.73	-397.71	61.96
C5H12	39.94	-233.21	-387.78	71.89
C6H14	48.78	-224.37	-371.87	87.80
C7H16	56.51	-216.64	-357.95	101.72
C8H18	63.63	-209.52	-345.14	114.53
C9H20	71.33	-201.82	-331.28	128.39
C10H22	78.67	-194.48	-318.06	141.61
C12H26	91.31	-181.84	-295.31	164.36
C15H32	111.80	-161.35	-258.43	201.24
C17H36	124.00	-149.15	-236.47	223.20
C18H38	129.90	-143.25	-225.85	233.82
C20H42	141.10	-132.05	-205.69	253.98

**Table 3: Critical Properties for Selected Components**

<b>Critical Temperature</b>				
<b>Component</b>	<b>(deg K)</b>	<b>(deg C)</b>	<b>(deg F)</b>	<b>(deg R)</b>
H <sub>2</sub> O	647.30	374.15	705.47	1165.14
H <sub>2</sub> S	373.20	100.05	212.09	671.76
N <sub>2</sub>	126.20	-146.95	-232.51	227.16
O <sub>2</sub>	154.60	-118.55	-181.39	278.28
CO	132.90	-140.25	-220.45	239.22
CO <sub>2</sub>	304.20	31.05	87.89	547.56
CH <sub>4</sub>	190.60	-82.55	-116.59	343.08
C <sub>2</sub> H <sub>6</sub>	305.40	32.25	90.05	549.72
C <sub>3</sub> H <sub>8</sub>	369.80	96.65	205.97	665.64
C <sub>4</sub> H <sub>10</sub>	425.20	152.05	305.69	765.36
C <sub>5</sub> H <sub>12</sub>	469.60	196.45	385.61	845.28
C <sub>6</sub> H <sub>14</sub>	507.40	234.25	453.65	913.32
C <sub>7</sub> H <sub>16</sub>	540.20	267.05	512.69	972.36
C <sub>8</sub> H <sub>18</sub>	568.80	295.65	564.17	1023.84
C <sub>9</sub> H <sub>20</sub>	594.60	321.45	610.61	1070.28
C <sub>10</sub> H <sub>22</sub>	617.60	344.45	652.01	1111.68
C <sub>12</sub> H <sub>26</sub>	658.30	385.15	725.27	1184.94
C <sub>15</sub> H <sub>32</sub>	707.00	433.85	812.93	1272.60
C <sub>17</sub> H <sub>36</sub>	733.00	459.85	859.73	1319.40
C <sub>18</sub> H <sub>38</sub>	745.00	471.85	881.33	1341.00
C <sub>20</sub> H <sub>42</sub>	767.00	493.85	920.93	1380.60

<b>Critical Pressure</b>				
<b>Component</b>	<b>(kPa)</b>	<b>(psi)</b>	<b>(atm)</b>	<b>(bar)</b>
H <sub>2</sub> O	22048.	3198.	217.6	220.5
H <sub>2</sub> S	8937.	1296.	88.20	89.37
N <sub>2</sub>	3394.	492.3	33.50	33.94
O <sub>2</sub>	5046.	731.9	49.80	50.46
CO	3496.	507.0	34.50	34.96
CO <sub>2</sub>	7376.	1070.0	72.80	73.76
CH <sub>4</sub>	4600.	667.2	45.40	46.00
C <sub>2</sub> H <sub>6</sub>	4884.	708.3	48.20	48.84
C <sub>3</sub> H <sub>8</sub>	4246.	615.8	41.90	42.46
C <sub>4</sub> H <sub>10</sub>	3800.	551.1	37.50	38.00
C <sub>5</sub> H <sub>12</sub>	3374.	489.4	33.30	33.74
C <sub>6</sub> H <sub>14</sub>	2969.	430.6	29.30	29.69
C <sub>7</sub> H <sub>16</sub>	2736.	396.8	27.00	27.36
C <sub>8</sub> H <sub>18</sub>	2482.	360.1	24.50	24.82
C <sub>9</sub> H <sub>20</sub>	2310.	335.1	22.80	23.10
C <sub>10</sub> H <sub>22</sub>	2108.	305.7	20.80	21.08
C <sub>12</sub> H <sub>26</sub>	1824.	264.5	18.00	18.24
C <sub>15</sub> H <sub>32</sub>	1520.	220.4	15.00	15.20
C <sub>17</sub> H <sub>36</sub>	1317.	191.0	13.00	13.17
C <sub>18</sub> H <sub>38</sub>	1206.	174.9	11.90	12.06
C <sub>20</sub> H <sub>42</sub>	1115.	161.7	11.00	11.15

## Table 4: Liquid Viscosity Coefficients for Selected Components

$$VISC = AVISC * EXP(BVISC / T)$$

where the values of the coefficients correspond to the units of viscosity VISC and absolute temperature T, and are based on VISB and VISTO in Appendix A, "The Properties of Gases and Liquids", third edition, R.C. Reid, J.M. Prausnitz and T.K Sherwood, McGraw-Hill, Inc., 1977.

Example: Estimate the viscosity in cp units of liquid water at 300 C.

- a) Temperature is  $T = 300\text{ C} + 273 = 573\text{ K}$ .
- b) Look up for water:  $AVISC = 0.0047352$ ,  $BVISC = 1515.7$ .
- c) Calculate  $VISC = 0.0047352 * \exp(1515.7 / 573)$   
 $= 0.0667\text{ cp}$ .

Component	AVISC (cp)	AVISC (kPa-day)	AVISC (kPa-hr)	BVISC (K,C)	BVISC (F,R)
H <sub>2</sub> O	0.0047352	5.480E-14	1.315E-12	1515.7	2728.2
H <sub>2</sub> S	0.0084969	9.833E-14	2.360E-12	789.30	1420.7
N <sub>2</sub>	0.0110386	1.277E-13	3.066E-12	207.92	374.26
O <sub>2</sub>	0.0216926	2.510E-13	6.025E-12	197.29	355.11
CO	0.0119257	1.380E-13	3.312E-12	216.58	389.85
CO <sub>2</sub>	0.0007573	8.764E-15	2.103E-13	1331.1	2395.9
CH <sub>4</sub>	0.0104328	1.207E-13	2.897E-12	262.82	473.07
C <sub>2</sub> H <sub>6</sub>	0.0229832	2.660E-13	6.383E-12	360.58	649.05
C <sub>3</sub> H <sub>8</sub>	0.0214257	2.479E-13	5.951E-12	512.72	922.89
C <sub>4</sub> H <sub>10</sub>	0.0219066	2.535E-13	6.084E-12	612.12	1101.8
C <sub>5</sub> H <sub>12</sub>	0.0191041	2.211E-13	5.306E-12	722.23	1300.0
C <sub>6</sub> H <sub>14</sub>	0.0177073	2.049E-13	4.918E-12	835.35	1503.6
C <sub>7</sub> H <sub>16</sub>	0.0132383	1.532E-13	3.677E-12	1005.6	1810.1
C <sub>8</sub> H <sub>18</sub>	0.0131242	1.519E-13	3.645E-12	1090.7	1963.3
C <sub>9</sub> H <sub>20</sub>	0.0117124	1.355E-13	3.253E-12	1210.1	2178.3
C <sub>10</sub> H <sub>22</sub>	0.0115577	1.337E-13	3.210E-12	1286.2	2315.2
C <sub>12</sub> H <sub>26</sub>	0.0104376	1.208E-13	2.899E-12	1454.4	2617.9
C <sub>15</sub> H <sub>32</sub>	0.0095777	1.108E-13	2.660E-12	1654.4	2978.0
C <sub>17</sub> H <sub>36</sub>	0.0096344	1.115E-13	2.676E-12	1745.1	3141.1
C <sub>18</sub> H <sub>38</sub>	0.0095671	1.107E-13	2.657E-12	1790.0	3222.1
C <sub>20</sub> H <sub>42</sub>	0.0095545	1.106E-13	2.654E-12	1868.1	3362.5

## Table 5: Gas Heat Capacity Coefficients for Selected Components

Gas phase heat capacities of these selected components can be estimated as a function of temperature, using data in this table which correspond to ideal-gas (zero pressure) conditions. Two correlation options are available. Use the more accurate four-coefficient correlation

$$C_{pg} = CPG1 + CPG2 \cdot T + CPG3 \cdot T^2 + CPG4 \cdot T^3$$

when gas is the reference phase for enthalpy data. These coefficients are based on Appendix A of "The Properties of Gases and Liquids", third edition, by R.C. Reid, J.M. Prausnitz and T.K Sherwood, McGraw-Hill, Inc., 1977. The less accurate two-coefficient correlation

$$C_{pg} = CPG1 + CPG2 \cdot T$$

was used in older STARS versions.

Coefficients are given in SI units and British units. If the STARS input temperature unit is either C or K then use the SI unit values; if the input temperature unit is either F or R then use the British unit values. Conversion factors resulting from mixing energy or mole units should be applied to each coefficient.

When doing hand calculations ensure that T in the correlation is in absolute degrees. For example, the N2 value in SI units at 63°C ( $T = 336.15\text{K}$ ) is  $31.15 - 1.357\text{e-}2 \cdot T + 2.68\text{e-}5 \cdot T^2 - 1.168\text{e-}8 \cdot T^3 = 29.173 \text{ J/gmol-C}$  whereas in British units the value at the same temperature 145.4°F ( $T = 605.07\text{R}$ ) is  $7.44 - 1.8\text{e-}3 \cdot T + 1.975\text{e-}6 \cdot T^2 - 0.4784\text{e-}9 \cdot T^3 = 6.968 \text{ Btu/lbmol-F}$ .

SI Units:  $C_{pg}$  unit is  $\text{J/gmol-C}$

Component	4-coefficient				2-coefficient	
	CPG1	CPG2	CPG3	CPG4	CPG1	CPG2
H2O	32.243	1.924e-3	1.055e-5	-3.596e-9	31.876	6.493e-3
H2S	31.941	1.436e-3	2.432e-5	-1.176e-8	31.190	1.114e-2
N2	31.150	-1.357e-2	2.680e-5	-1.168e-8	30.288	-2.572e-3
O2	28.106	-3.680e-6	1.746e-5	-1.065e-8	27.627	6.437e-3
CO	30.869	-1.285e-2	2.789e-5	-1.272e-8	29.987	-1.542e-3
CO2	19.795	7.344e-2	5.602e-5	1.715e-8	16.864	.1063
CH4	19.251	5.213e-2	1.197e-5	-1.132e-8	19.031	5.559e-2
C2H6	5.409	.1781	-6.938e-5	8.713e-9	8.226	.1445
C3H8	-4.224	.3063	-1.586e-4	3.215e-8	1.885	.2324
C4H10	9.487	.3313	1.108e-4	-2.822e-9	4.686	.3876
C5H12	-3.626	.4873	-2.580e-4	5.305e-8	6.292	.3674
C6H14	-4.413	.5820	-3.119e-4	6.494e-8	7.551	.4371
C7H16	-5.146	.6762	-3.651e-4	7.658e-8	8.845	.5067
C8H18	-6.096	.7712	-4.195e-4	8.855e-8	9.965	.5766
C9H20	3.144	.6774	-1.928e-4	-2.981e-8	12.441	.5712
C10H22	-7.913	.9609	-5.288e-4	1.131e-7	12.291	.7160
C12H26	-9.328	1.149	-6.347e-4	1.359e-7	14.919	.8550
C15H32	-11.916	1.433	-7.972e-4	1.720e-7	18.502	1.064
C17H36	-13.967	1.624	-9.081e-4	1.972e-7	20.650	1.204
C18H38	-14.470	1.717	-9.592e-4	2.078e-7	22.107	1.273
C20H42	-22.383	1.939	-1.117e-3	2.528e-7	19.901	1.426

---

## Table 5: Gas Heat Capacity Coefficients for Selected Components (Continued)

British Units: Cpg unit is Btu/lbmol-F (= cal/gmol-C)

Component	4-coefficient				2-coefficient	
	CPG1	CPG2	CPG3	CPG4	CPG1	CPG2
H2O	7.701	2.553e-4	7.781e-7	-.1473e-9	7.613	8.616e-4
H2S	7.629	1.906e-4	1.793e-6	-.4818e-9	7.450	1.478e-3
N2	7.440	-1.800e-3	1.975e-6	-.4784e-9	7.234	-3.412e-4
O2	6.713	-4.883e-7	1.287e-6	-.4362e-9	6.599	8.541e-4
CO	7.373	-1.706e-3	2.056e-6	-.5207e-9	7.162	-2.046e-4
CO2	4.728	9.744e-3	4.130e-6	.7025e-9	4.028	1.411e-2
CH4	4.598	6.917e-3	8.827e-7	-.4635e-9	4.545	7.376e-3
C2H6	1.292	2.363e-2	-.5114e-6	.3568e-9	1.965	1.918e-2
C3H8	-1.009	4.064e-2	-.1169e-5	1.316e-9	0.450	3.084e-2
C4H10	2.266	4.396e-2	8.170e-6	-.1156e-9	1.119	5.143e-2
C5H12	-0.866	6.467e-2	-.1902e-5	2.173e-9	1.503	4.874e-2
C6H14	-1.054	7.722e-2	-.2299e-5	2.660e-9	1.804	5.800e-2
C7H16	-1.229	8.972e-2	-.2691e-5	3.136e-9	2.113	6.724e-2
C8H18	-1.456	.1023	-.3093e-5	3.626e-9	2.380	7.652e-2
C9H20	0.751	8.989e-2	-.1422e-5	-.1221e-9	2.972	7.580e-2
C10H22	-1.890	.1275	-.3898e-5	4.631e-9	2.936	9.501e-2
C12H26	-2.228	.1524	-.4679e-5	5.566e-9	3.563	.1134
C15H32	-2.846	.1901	-.5876e-5	7.044e-9	4.419	.1412
C17H36	-3.336	.2155	-.6694e-5	8.076e-9	4.932	.1598
C18H38	-3.456	.2278	-.7071e-5	8.512e-9	5.280	.1690
C20H42	-5.346	.2573	-.8232e-5	1.036e-8	4.753	.1892

## Table 6: Vaporization Enthalpy for Selected Components

Vapourization enthalpies of these selected components can be estimated as a function of temperature, using data in this table. The correlation is

$$H_{vap}(T) = HVR * (T_c - T)^{0.38}$$

where T is temperature, Tc is critical temperature of the component and HVR is obtained from the table, below.

Example: Find the vapourization enthalpy of water at 100°C.

- a) Tc = 374.15°C (TCR from Table 3).
- b) HVR = 4820 J/gmol-C<sup>0.38</sup> from the table below.
- c) Hvap(100°C) = (4820 J/gmol-C<sup>0.38</sup>) \* (374.15°C - 100°C)<sup>0.38</sup> = 40,690 J/gmol.
- d) To check, convert to cal/gmol by multiplying by 4.1868 J/cal to get 9,718 cal/gmol. This is the Normal Boiling Enthalpy since T = 100°C is the Normal Boiling Point of water.

Coefficient HVR is given in the SI units (J/gmol-K<sup>0.38</sup>) and British units (Btu/lbmol-R<sup>0.38</sup>). The appropriate conversion factors must be applied if SI and British units are mixed.

Component	Normal Molecular Mass (gm/gmol)	Normal Boiling Point (deg K)	Normal Boiling Enthalpy (cal/gmol)	HVR (J, gmol, K or C)	HVR (Btu, lbmol, F or R)
H <sub>2</sub> O	18.015	373.2	9717.	4820.	1657.
H <sub>2</sub> S	34.080	212.8	4460.	2712.	932.
N <sub>2</sub>	28.013	77.4	1333.	1274.	438.
O <sub>2</sub>	31.999	90.2	1630.	1402.	482.
CO	28.010	81.7	1444.	1355.	466.
CO <sub>2</sub>	44.010	194.7	4100.	2882.	991.
CH <sub>4</sub>	16.043	111.7	1955.	1556.	535.
C <sub>2</sub> H <sub>6</sub>	30.070	184.5	3515.	2379.	818.
C <sub>3</sub> H <sub>8</sub>	44.097	231.1	4487.	2883.	991.
C <sub>4</sub> H <sub>10</sub>	58.124	272.7	5352.	3317.	1141.
C <sub>5</sub> H <sub>12</sub>	72.151	309.2	6160.	3745.	1288.
C <sub>6</sub> H <sub>14</sub>	86.178	341.9	6896.	4143.	1425.
C <sub>7</sub> H <sub>16</sub>	100.205	371.6	7576.	4520.	1554.
C <sub>8</sub> H <sub>18</sub>	114.232	398.8	8225.	4892.	1682.
C <sub>9</sub> H <sub>20</sub>	128.259	424.0	8823.	5240.	1802.
C <sub>10</sub> H <sub>22</sub>	142.286	447.3	9388.	5579.	1919.
C <sub>12</sub> H <sub>26</sub>	170.340	489.5	10430.	6220.	2139.
C <sub>15</sub> H <sub>32</sub>	212.421	543.8	11820.	7139.	2455.
C <sub>17</sub> H <sub>36</sub>	240.475	575.2	12640.	7733.	2659.
C <sub>18</sub> H <sub>38</sub>	254.502	589.5	13020.	8010.	2754.
C <sub>20</sub> H <sub>42</sub>	282.556	617.0	13740.	8569.	2947.

---

## Table 7: Selected Unit Conversions

Pressure:	1 atm = 101.3250 kPa 1 bar = 100 kPa (exact) 1 psi = 6.894757 kPa 1 kg/cm <sup>2</sup> (kg force per cm <sup>2</sup> ) = 98.0665 kPa
Temperature:	deg F = (F-32)/1.8 deg C deg F = (F+459.67)/1.8 K
Length:	1 ft = 0.3048 m (exact) 1 in = 2.54 cm (exact) 1 mile = 1609.344 m (exact)
Volume:	1 acre = 4046.873 m <sup>2</sup> 1 acre-ft = 1233.489 m <sup>3</sup> = 43,560 ft <sup>3</sup> 1 bbl = 0.1589873 m <sup>3</sup> = 5.61457 ft <sup>3</sup> = 42 US gal 1 US gal = 0.003785412 m <sup>3</sup> 1 litre = 0.001 m <sup>3</sup> (exact) 1 Mscf = 28.3168 m <sup>3</sup>
Viscosity:	1 cp = 1 mPa-s = 11.574·10 <sup>-12</sup> kPa-day 1 stokes = 0.0001 m <sup>2</sup> /s (exact)
Permeability:	1 darcy = 0.986923·10 <sup>-12</sup> m <sup>2</sup> = 0.986923 μm <sup>2</sup>
Mass:	1 lbm = 0.4535924 kg 1 ton = 0.9071847 Mg 1 tonne = 1 Mg 1 lb mole = 453.592 gm mole Specific gravity (1 atm, 60°F) = 141.5/(131.5+API)
Energy:	1 Btu = 1055.056 J 1 Btu/ft-day-F = 6230.65 J/m-day-K 1 Btu/ft-hr-F = 149,520 J/m-day-K 1 calorie = 4.1868 J 1 Btu/lbmole = 2.32544 J/gmole 1 cal/gm-K = 1 Btu/lbm-F (exact)
Transmissibility to Flow Volume:	1 md-ft = 1.127166·10 <sup>-3</sup> bbl-cp/psi-day 1 md-m = 8.527395·10 <sup>-5</sup> m <sup>3</sup> -cp/kPa-day 1 md-cm = 5.921538·10 <sup>-4</sup> cm <sup>3</sup> -cp/kPa-min 1 μm <sup>2</sup> -cm = 0.6 cm <sup>3</sup> -cp/kPa-min (exact)
Gas Constant R:	8.31431 J/K-gmole (*INUNIT *SI & *LAB) 1.986 Btu/R-lbmole (*INUNIT *FIELD) 1.98717 cal/K-gmole 0.0820552 liter-atm/K-gmole 0.7302 ft <sup>3</sup> -atm/R-lbmole

# Appendix A: Well Model Details

## Overview

This appendix contains detailed descriptions of portions of the well models used by STARS, divided into the following sections.

- A.1 Radial Inflow Well Model
- A.2 Well Indices
- A.3 Anisotropic Permeability
- A.4 Backflow
- A.5 Surface Flash
- A.6 Calculation of Geometrical Factor CC
- A.7 Notes on Discretized Wellbore Model Usage

---

## A.1 Radial Inflow Well Model

The radial inflow well model couples the pressure in the wellbore to the average grid block pressure. The grid block is deemed to have an effective radius, and the radial flow equation is solved between the effective block radius and the wellbore radius. An equivalent proportionality factor (well index) is developed for the appropriate pressure drop, in the form of the general equation

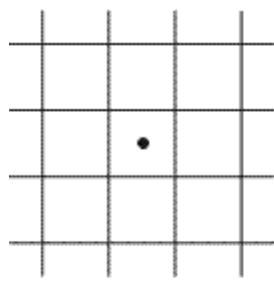
$$q_{jk} = I'_k \lambda_{jk} (p_{wfk} - p_k) \quad (\text{A1.1a})$$

$$I'_k = \frac{2\pi h k f_h f}{\ln(r_e/r_w) + S} \quad (\text{A1.1b})$$

$q_{jk}$	downhole flow rate of phase j, layer k (normally positive for injection and negative for production)
$I'_k$	well index, layer k (portion independent of fluid conditions)
$\lambda_{jk}$	relative mobility of phase j, layer k (See Appendix A.2)
$p_{wfk}$	flowing wellbore pressure, layer k
$p_k$	grid block pressure, layer k
$h$	layer thickness in the <i>well direction</i> which is from *GEOMETRY
$k$	absolute permeability around wellbore, normal to <i>well direction</i>
$f_h$	layer thickness factor ( $= ff$ in *PERF). Layer completion length is $f_h \bullet h$ .
$f$	well fraction ( $=wfrac$ in *GEOMETRY and Figure A.1). This will be 1 for a well going approximately through the centre of a grid block, $\frac{1}{2}$ for a half well on a grid block boundary, and $\frac{1}{4}$ for a quarter well at the corner of a grid block.
$r_w$	wellbore radius (from *GEOMETRY)
$S$	skin factor (from *GEOMETRY)
$r_e$	effective block radius; for isotropic reservoir given by

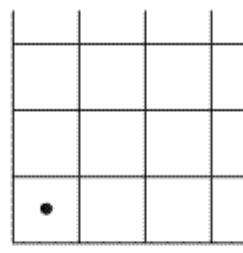
$$r_e = CC \sqrt{\frac{\Delta x^2 + \Delta y^2}{f\pi}} \quad (\text{A1.2})$$

where CC (=geofac in \*GEOMETRY and Figure A.1) is a factor depending on the geometry of the situation, and  $\Delta x$  and  $\Delta y$  are block sizes in the plane approximately perpendicular to the well (Peaceman<sup>1</sup>). See also Appendix A.6.



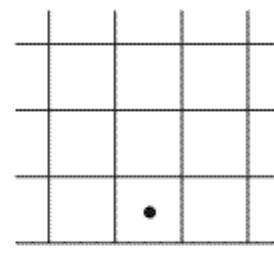
wfrac = 1  
geofac = 0.249

(a) Centre of Grid Block



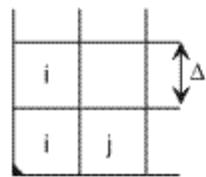
wfrac = 1  
geofac = 0.229

(b) Center of Corner Block



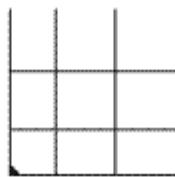
wfrac = 1  
geofac = 0.235

(c) Centre of Edge Block



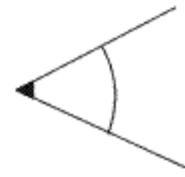
wfrac = 1/4  
geofac = 0.432

(d) Well in Corner



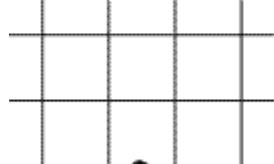
wfrac = 1/4  
geofac = 0.377

(e) Edge Blocks Half Size



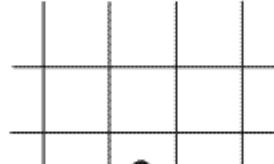
wfrac = segment  
geofac = 0.5

(f) Radial Segment



wfrac = 1/2  
geofac = 0.363

(g) Well on Edge



wfrac = 1/2  
geofac = 0.284

(h) Edge Blocks Half Size

**Figure A.1: Well Fraction and Geometrical Factor for Various Common Geometries Used in Keyword \*GEOMETRY**

Note: If a well is located on the block node in a repeating pattern, case (a) should be used. Symmetry element boundaries are handled via keywords \*VAMOD and \*WELL \*FRAC.

---

## A.2 Well Indices

### Production

For a production well the radial inflow model can be used directly. The parameters  $r_w$ , CC, f, S, and  $f_h$  are required data; h and k are obtained from the grid data. The constant part  $I_k'$  is evaluated and then reported. The phase flow rate is then

$$q_{ik} = I_k' (p_{wfk} - p_k) \lambda_j \quad (\text{A2.1})$$

where

$$\lambda_j = \frac{k_{rj}}{\mu_j} \quad (\text{A2.2})$$

Because this is production, the upstream mobility is taken from the grid block conditions.

There is also an option in STARS for entering the quantity  $I_k'$  directly as data. In each case, volumes are at reservoir conditions. The surface flash may be used to convert from reservoir rates to surface rates.

### Injection

The same equation applies for injection, but the well index may be specified in two different ways.

1. Mobility Weighting Option

For option \*MOBWEIGHT,  $\lambda_j$  is the total downstream (grid block) mobility and is applied internally, so that the well index entered by the user is only the geometry factor  $I_k'$ . This option recognizes that injection is achieved by the displacement of fluid in the completion block. STARS would calculate the geometry factor  $I_k'$  from the geometry information if requested by \*PERF GEO.

2. Mobility Unweighting Option

For option \*UNWEIGHT,  $\lambda_j$  is the mobility of the injected fluid and must be included in the well index entered by the user. The included phase mobility is assumed to be constant. The constant part may be calculated from the radial inflow model. In fact, the value  $I_k'$  reported for the production well can be used directly, if appropriate. The phase mobility can be estimated from representative values for  $k_{rj}$  (=1, perhaps) and  $\mu_j$  at the injection temperature. For steam injection processes, a cold water equivalent (CWE) quality option is available, which essentially expresses a two phase water/steam injection in terms of an equivalent amount (moles or mass) of injected water phase volumes. In this case, an equivalent single-phase water injectivity index is required.

$$I_{wj}^{\text{ST(CWE)}} \equiv f_w^{\text{RC}} I_{wk}^{\text{RC}} + f_s^{\text{RC}} I_{gk}^{\text{RC}} \quad (\text{A2.3})$$

where the volume fractions of injected water and steam, expressed at reservoir conditions,  $f_w^{RC}$ , are obtained from the injected quality Q by conservation of mass

$$Q = \frac{\text{masssteam}}{\text{masssteam} + \text{masswater}} \quad (\text{A2.4})$$

$$= \frac{\rho_s^{RC} + f_s^{RC}}{\rho_s^{RC} f_s^{RC} + \rho_w^{RC} f_w^{RC}} \quad (\text{A2.5})$$

which yields

$$\begin{aligned} f_w^{RC} &= \left\{ 1 + \frac{Q}{1-Q} \frac{\rho_w^{RC}}{\rho_s^{RC}} \right\}^{-1} \\ f_s^{RC} &= 1 - f_w^{RC} \end{aligned} \quad (\text{A2.6})$$

It is worth emphasizing again, that with the CWE approach, the volume of equivalent water injected is not the same as that obtained by the separate phase (water/steam) approach, although the moles or mass (and energy) injected are.

If \*PERF GEO is used then STARS would calculate the geometry factor  $I_k'$  as well as the appropriate injection fluid mobility.

### Surface Rate

The surface rate is obtained by multiplying the downhole rate by the density ratio  $\rho_{jr}/\rho_{js}$ , which is calculated and applied internally and must not be included in the well index entered by the user.

### A.3 Anisotropic Permeability

The following is from Peaceman<sup>1</sup>, and is used for the \*GEOA and \*KHA options.

Consider a vertical well located in a reservoir in which the principal axes of the permeability tensor are parallel to the X- and Y-axes. Then the differential equation for steady state pressure is:

$$k_x \frac{\partial^2 p}{\partial x^2} + k_y \frac{\partial^2 p}{\partial y^2} = 0 \quad \left. \begin{array}{l} k_x = \text{perm. in } x - \text{direction} \\ k_y = \text{perm. in } y - \text{direction} \end{array} \right\} \quad (\text{A3.1})$$

By making the change of variables

$$u = (k_y / k_x)^{1/4} x \text{ and } v = (k_x / k_y)^{1/4} y \quad (\text{A3.2})$$

the pressure will satisfy the equation

$$P - P_{wf} = \frac{qu}{2\pi(k_x / k_y)^{1/2} h} \ln \frac{r^{uv}}{\hat{r}_w} \quad (\text{A3.3})$$

where

$$r^{uv} = (u^2 + v^2)^{1/2} \quad (\text{A3.4})$$

and

$$\hat{r}_w = 1/2 r_w [(k_y / k_x)^{1/4} + (k_x / k_y)^{1/4}] \quad (\text{A3.5})$$

In the u-v plane, the difference equation is identical to the isotropic problem.

$$r_e^{uv} = 0.14(\Delta u^2 + \Delta v^2)^{1/2} \quad (\text{A3.6})$$

and

$$r_e = (r_w / \hat{r}_w) r_e^{uv} \quad (\text{A3.7})$$

So,

$$r_e = 0.28 \frac{[(k_y / k_x)^{1/2} \Delta x^2 + (k_x / k_y)^{1/2} \Delta y^2]^{1/2}}{(k_y / k_x)^{1/4} + (k_x / k_y)^{1/4}} \quad (\text{A3.8})$$

The \*GEO and \*KH options use an expression for  $r_e$  that is the same as in equation A3.8 except that constant 0.28 is replaced with  $\text{geofac} \cdot 2 / \sqrt{\pi}$ . The two expressions are the same for the default “centre of block” case in which  $\text{geofac} = 0.249$  (Figure A.1 (a)). This means that \*GEO with  $\text{geofac} = 0.249$  gives the same result as \*GEOA, and \*KH with  $\text{geofac} = 0.249$  gives the same result as \*KHA.

The above derivation is written for a vertical well (\*GEOMETRY \*K). A similar calculation involves  $k_y$ ,  $k_z$ ,  $\Delta y$  and  $\Delta z$  for the \*I direction and  $k_x$ ,  $k_z$ ,  $\Delta x$  and  $\Delta z$  for the \*J direction.

---

## A.4 Backflow

A well completed in only one grid block is relatively simple to control, since the well equation

$$q_{jl} = I_{jl} \cdot (p_{wf} - p) \quad (\text{A4.1})$$

is linear with respect to the two primary variables  $p_{wf}$  and  $p$ . A specified rate  $q_{jl}$  will be honored. A specified bottom hole pressure  $p_{wf}$  will result in backflow under exactly one condition, i.e.,  $p > p_{wf}$  for injection, or  $p < p_{wf}$  for production. These conditions are easy to detect.

A well completed in several blocks is more complicated and can be difficult to control, since

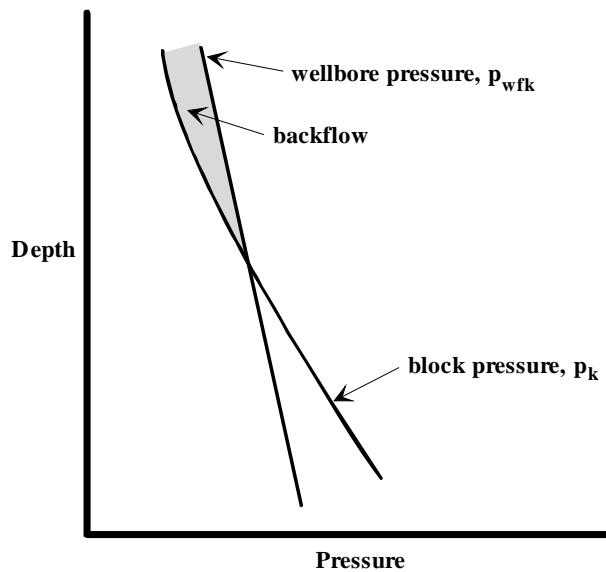
$$q_j = \sum_{k=1}^{n_{\text{lay}}} I_{jk} \cdot (p_{wfk} - p_k) \quad (\text{A4.2})$$

depends upon many variables. Under many conditions it is possible for some layers to experience backflow. For an injector, enough layers may backflow ( $p_{wfk} < p_k$ ) that  $q_j$ , which should be positive for injection, may be negative.

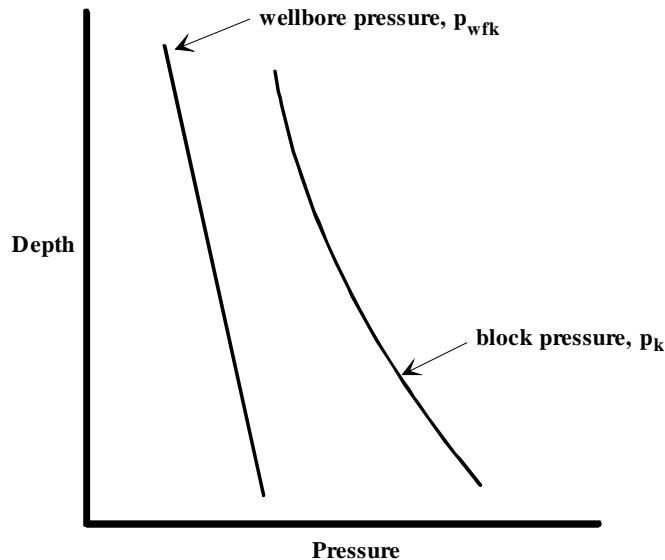
Backflow in any layer  $k$  is controlled by the relationship between  $p_{wfk}$  and  $p_k$ . The block pressure  $p_k$  is involved intimately with the component flow equations, which carry the fluid to or from the wellbore. The vertical pressure gradient in the reservoir is determined by the fluid head between nodes, which depends upon the saturations in those blocks.

On the other hand, the wellbore pressure  $p_{wfk}$  is determined by the fluid head in the wellbore. This head or pressure gradient depends upon the saturations in the wellbore. These saturations are not the same as in the grid block, but are related to what flowed out of the grid block into the wellbore. Thus, the wellbore fluid depends on the fluid mobilities in the block. This discrepancy between the two pressure gradients is illustrated in Figure A.2, where  $p_k > p_{wfk}$  denotes normal production. This condition can occur even for a constant rate well, and is common in problems containing long completion lengths (>3m).

This condition can be avoided by decreasing the well index  $I_{jk}$  such that the pressure drop ( $p_{wfk} - p_k$ ) increase. The result is shown in Figure A.3. Some situations may require the backflow to be modelled, but the well model does not account for the change in upstream direction and conditions. Also, partial backflow may affect simulator performance, especially when a constant-surface-rate constraint is controlling the well.



*Figure A.2: Partial backflow during production*



*Figure A.3: No backflow during production*

## A.5 Surface Flash

In order to convert reservoir rates and compositions to surface conditions, the flow stream is passed through a surface flash. The following steps are followed.

1. Calculate ncf (the number of fluid components) mole rates  $q_i$ , total mole rate  $q$ , and global mole fractions  $z_i$ .

$$q_i = \rho_w q_{wk} w_i + \rho_o q_{ok} x_i + \rho_g q_{gk} y_i \quad i = 1 \text{ to } n_{cf} \quad (\text{A5.1})$$

$$q = \sum_{i=1}^{n_{cf}} q_i \quad (\text{A5.2})$$

$$z_i = q_i / q \quad i = 1 \text{ to } n_{cf} \quad (\text{A5.3})$$

Since the  $q_{jk}$  have the same sign, the  $q_i$  will all have the same sign and the  $z_i$  will satisfy  $0 \leq z_i \leq 1$ , whether  $q$  is positive or negative.

2. Calculate the phase splits  $f_w$ ,  $f_o$  and  $f_g$  when this global composition is flashed to surface  $p$  and  $T$ . Two options are available: segregated phases or rigorous phase equilibrium flash.
3. The surface compositions are  $x_i$  and  $y_i$  from the flash. Water and gas standard densities  $\rho_w^{\text{ST}}$  and  $\rho_g^{\text{ST}}$  are constant. Calculate the oil density  $\rho_o^{\text{ST}}$  from composition  $x_i$  and component liquid densities  $\rho_{oi}^{\text{ST}}$

$$\frac{1}{\rho_o^{\text{ST}}} = \sum_{i=1}^{n_c} \frac{x_i}{\rho_{oi}^{\text{ST}}} \quad (\text{A5.4})$$

The surface volume rates are  $q \bullet f_j / \rho_j^{\text{ST}}$ .

## A.6 Calculation of Geometrical Factor CC

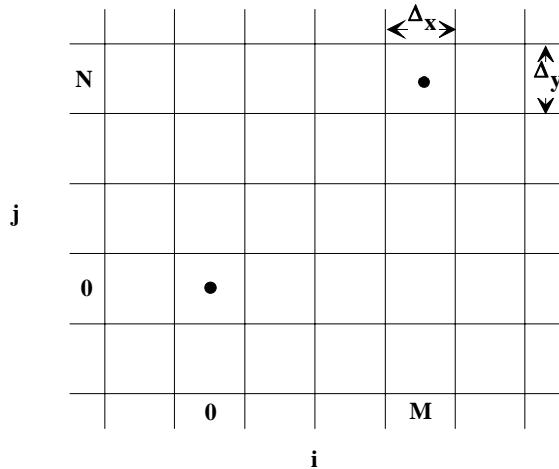


Figure A.4: Effective well radius

Assume incompressible, single-phase, steady state flow, with no skin effects.

For all blocks,  $0 \leq i \leq M$ ,  $0 \leq j \leq N$  (see Figure A.4), the finite difference equation for the steady state pressure distribution is

$$q_{i,j} = \frac{kh\Delta y}{\mu\Delta x}(P_{i+1,j} - 2P_{i,j} + P_{i-1,j}) + \frac{kh\Delta x}{\mu\Delta y}(P_{i,j+1} - 2P_{i,j} + P_{i,j-1}) \quad (\text{A6.1})$$

Assume production at rate  $q$  at the lower left-hand corner, and injection at rate  $q$  at the upper right-hand corner. Thus:

$$\begin{aligned} q_{0,0} &= q \\ q_{M,N} &= -q \\ q_{i,j} &= 0 \text{ for } (i,j) \neq (0,0) \text{ or } (M,N) \end{aligned} \quad (\text{A6.2})$$

Defining  $P_D = (kh / q\mu)p$  allows (A6.2) to be simplified to

$$\alpha(P_D)_{i-1,j} + (P_D)_{i+1,j} + (1/\alpha)(P_D)_{i,j-1} + (P_D)_{i,j+1} - (2\alpha + 2/\alpha)(P_D)_{i,j} = \delta_{i,j} \quad (\text{A6.3})$$

where

$$\left. \begin{aligned} \delta_{i,j} &= 0 \text{ for } i,j \neq (0,0) \text{ or } (M,N) \\ \delta_{0,0} &= 1 \\ \delta_{M,N} &= -1 \\ \alpha &= M/N \end{aligned} \right\} \quad (\text{A6.4})$$

Changing variables and taking the limit we obtain the integral solution for an infinite grid.

$$(P_D)_{i,j} = \frac{-1}{\pi^2} \int_0^{\pi/2} \int_0^{\pi/2} \frac{\cos(2iu)\cos(2jv)}{\alpha \sin^2 u + (1/\alpha) \sin^2 v} du dv \quad (\text{A6.5})$$

For large  $i$ , the solution on the infinite grid satisfies the exact radial solution. Along the horizontal axis ( $j=0$ ), the exact radial solution is

$$(P_D)_{i,0} = (P_D)_{wf} + (1/2\pi) \ln(r_e/r_w) \quad (A6.6)$$

By the definition of well-block equivalent radius,

$$(P_D)_{0,0} = (P_D)_{wf} + (1/2\pi) \ln(r_e/r_w) \quad (A6.7)$$

Subtracting these two equations gives

$$\ln(r_e/i\Delta x) = 2\pi(P_D)_{0,0} - (P_D)_{i,0} \quad (A6.8)$$

Combine this with (A6.5) to get

$$\ln(r_e/i\Delta x) = \int_0^{\pi/2} \frac{\cos(2iu)-1}{\sin u (1+\alpha^2 \sin^2 u)^{1/2}} du \quad (A6.9)$$

Expand in Taylor series and take the limit,

$$\lim_{1 \rightarrow \infty} \ln(r_e/\Delta x) = -\gamma - 2\ln 2 + 1/2 \ln(1+\alpha^2) \quad (A6.10)$$

$$\lim_{1 \rightarrow \infty} \frac{r_e}{(\Delta x^2 + \Delta y^2)^{1/2}} = \frac{e^{-\gamma}}{4} = 0.1403649 \quad (A6.11)$$

For a square grid,  $\Delta x = \Delta y$  and (A6.10) becomes

$$\lim_{1 \rightarrow \infty} \frac{r_e}{\Delta x} = \frac{\sqrt{2} e^{-\gamma}}{4} = 0.198506 \quad (A6.12)$$

The geometry factor CC is defined as

$$r_e = CC \sqrt{\frac{\Delta x^2 + \Delta y^2}{f\pi}} \quad (A6.13)$$

For the case with  $f = 1.0$  and  $\Delta x = \Delta y$ , equation (A6.13) becomes

$$\frac{r_e}{\Delta x} = CC \sqrt{\frac{2}{\pi}}$$

Compare this with (A6.12). The result is  $CC \approx 0.249$ .

The appropriate CC factor can be derived for other geometries using a similar procedure and the method of images. The results for some common cases are presented in Figure A.1. These results include the effects of anisotropic permeability.

---

## A.7 Notes on Discretized Wellbore Model Usage

### Theory and Implementation

The Discretized Wellbore (DW) Model is a fully coupled mechanistic wellbore model. It models fluid and heat flow in the wellbore and between a wellbore and a reservoir / overburden. Wellbore mass and energy conservation equations are solved together with reservoir equations for each wellbore section (perforation).

#### Wellbore Flow

To be able to solve the wellbore and reservoir equations together, some steps had to be taken to translate the pipe flow equations into Darcy's law equations. Darcy's law equations are used in reservoir simulation for flow in porous media. It means, that properties such as porosity, permeability, etc. must be assigned to the wellbore. For example, permeability may be evaluated by equating pipe flow and porous media velocity:

Velocity equation in porous media in x- direction is:

$$v = - \frac{kk_r}{\mu} \frac{\partial \Phi}{\partial x} \quad (\text{A7.1})$$

$k$  = permeability  
 $k_r$  = relative permeability  
 $\frac{\partial \Phi}{\partial x}$  = potential gradient  
 $\mu$  = viscosity

Velocity equation for homogeneous flow in a pipe is:

$$v^2 = \frac{r_w}{f\rho} \frac{\partial \Phi}{\partial x} \quad (\text{A7.2})$$

$r_w$  = wellbore radius  
 $f$  = Fanning friction factor  
 $\rho$  = mass density

When we assume that the relative permeability curves in a pipe are straight lines going from zero to one then for homogeneous fluid  $k_r = 1$  and for multiphase flow  $k_r$  will equate to saturation. For laminar flow  $f = 16/Re$  and

$$Re = \frac{2v\rho r_w}{\mu} \quad (\text{A7.3})$$

Substituting these values into equation (A7.2) will give permeability in a laminar mode as

$$\frac{r_w^2}{8} \quad (\text{A7.4})$$

Permeability expression for a turbulent flow is more complex and depends also on friction factor, fluid viscosity and density. It is also evaluated from equations (A7.1) and (A7.2) as

$$k = \mu \left[ \frac{r_w}{\rho f} \frac{\partial x}{\partial \Phi} \right]^{1/2} \frac{\partial \Phi}{\partial x} \quad (\text{A7.5})$$

Permeability is updated at each timestep and its value would depend on the flow pattern and fluid composition. Potential gradient  $\partial\Phi/\partial x$  is the sum of frictional, gravity and viscous forces. Friction factor for turbulent, single-phase flow is calculated from Colebrook's equation as:

$$\frac{1}{\sqrt{f}} = 4 \ln \frac{1}{2\epsilon} + 3.48 - 4 \ln \left[ 1 + \frac{9.35}{2\epsilon \operatorname{Re} \sqrt{f}} \right] \quad (\text{A7.6})$$

$\epsilon$  = relative roughness

When two phase fluid (liquid-gas) is present in the wellbore then liquid hold-up must be also considered in the friction pressure drop calculation. Liquid hold-up represents a slip between gas and liquid phase. Its magnitude depends on the flow regime i.e. the amount of each phase present as well as phase velocities. Liquid hold-up  $R_g$  is predicted from Bankoff's correlation as:

$$\frac{1}{Y} = 1 - \frac{\rho_l}{\rho_g} \left[ 1 - \frac{K}{R_g} \right] \quad (\text{A7.7})$$

The correlation parameter K is a function of Reynolds number, Froude number and a flowing mass void fraction Y. It may attain values from 0.185 to one. Gas phase mobility is altered to account for the difference in liquid and gas phase velocities i.e. gas relative permeability is augmented by the ratio of gas saturation and void fraction  $R_g$ . This operation relates the liquid hold-up calculated from pipe flow equations to saturation needed in flow equations in porous media.

Wellbore hydraulics may be used in wells with co-current upward or horizontal flow due to the chosen correlation for the liquid hold-up.

#### *Annulus Flow*

In dual stream wells flow through tubing and annulus must be considered. Tubing flow is handled similarly as wellbore flow. For laminar flow the annulus permeability is calculated as:

$$k_a = \frac{1}{8} \left[ r_a^2 + r_t^2 - \frac{r_a^2 - r_t^2}{\ln \frac{r_a}{r_t}} \right] \quad (\text{A7.8})$$

$r_a$  - annulus radius

$r_t$  - tubing radius

Velocity and permeability for turbulent flow is calculated with the proper hydraulic diameter for annulus. The same correlations as mentioned above are used to calculate the friction pressure drop and slip between gas and liquid phase. Correct area and hydraulic diameter is applied wherever necessary.

#### *Tubing - Annulus Flow*

Only conductive heat transfer is allowed between tubing and annulus along the tubing length. Fluid is allowed to flow to the annulus at the end of the tubing. The same equations as mentioned above are used, but the equivalent drainage radius is calculated as

$$r_T = r_t \exp \left[ \frac{\alpha^2}{\alpha^2 - 1} \ln \alpha - \frac{1}{2} \right] \quad (A7.9)$$

$$\alpha = \frac{r_a}{r_t} \quad (A7.10)$$

#### *Wellbore-Reservoir Flow*

Fluid and energy flow between a wellbore section and a reservoir grid block is handled the same way as between individual reservoir grid blocks. Peaceman's equation is used to calculate transmissibilities (well index) between wellbore sections and the reservoir.

$$T_j = \frac{2\pi \Delta x \bar{k}}{\ln \frac{r_o}{r_k}} \left( \frac{k_{rj}}{\mu_j r_j} \right) \quad (A7.11)$$

$r_k$  - wellbore or annulus radius

$$\bar{k} = \sqrt{k_x k_y} \quad (A7.12)$$

The equivalent drainage radius  $r$  is obtained from

$$r_o = 0.28 \frac{\left[ \left( k_z / k_y \right)^{1/2} \Delta y^2 + \left( k_y / k_z \right)^{1/2} \Delta z^2 \right]^{1/2}}{\left( k_z / k_y \right)^{1/4} + \left( k_y / k_z \right)^{1/4}} \quad (A7.13)$$

These internally calculated values may be changed by the user with the keyword \*TRANSWB if necessary.

The flow term of energy consists of convective and conductive flow. Convective heat transfer uses the same phase transmissibilities  $T_j$  as the component flow equations.

Conductive transmissibility is expressed as

$$K_o = \frac{2\pi \Delta x \kappa}{\ln \frac{r_T}{r_k}} \quad (A7.14)$$

with equivalent drainage radius

$$r_T = 0.14 \sqrt{\Delta y^2 + \Delta z^2} \quad (A7.15)$$

The inflow and outflow of fluid through each perforation changes properties in each wellbore section because all wellbore conservation equations are solved implicitly. Therefore, the DW model is able to handle backflow (crossflow) between reservoir and a wellbore correctly.

#### *Wellbore Initialization and Transient Behavior*

The initial conditions in the wellbore will determine the short behavior of a reservoir in the vicinity of a well and dictate the length of a transient state. When initial pressure, temperature and composition differ considerably from conditions at which fluid is injected or produced, the period of transient behavior may be extended to several days. Simulation of the transient behavior does not affect long term physical results for most of the processes used in EOR simulation. However, transients may be important in cyclic processes where the cycle duration is the same order of magnitude as the transient period. The transient period is

generally longer for injectors. It will increase when low mobility fluid is injected or when low mobility fluid is originally in the wellbore. Simulation of wellbore transients is necessary in well test analysis.

The effect of wellbore transients on numerical performance is larger in heavy oils or bitumen reservoirs than in conventional oil reservoirs due to very low oil mobility. It also seems to be more pronounced in injectors than producers. In addition, attempts to simulate the transient period will change the overall numerical performance in comparison with the sink/source approach where pseudo-steady state is assumed. High pressure, temperature or saturation changes occur due to small wellbore volume. Even in an implicit simulator the timestep size will be fairly small (10e-3 to 10e-4 days, probably smaller for high rates). For example, the worst scenario is to inject steam into a wellbore containing cold oil, which may be the case after primary production. Thus, the well type may be changed instantaneously, but the condition in the discretized part of the well will take time to change.

If one is not interested in the wellbore's transient behavior, the initial conditions should be a pseudo-steady state to avoid a lengthy equilibration period. This is achieved by omitting the keyword \*TRANSIENT or by using \*TRANSIENT OFF. This keyword may also be used in recurrent data. When transient behavior is not requested then STARS will do automatic pseudo-steady state initialization in the discretized wellbore at the beginning of simulation and at each time when operating conditions are changed. Operating conditions such as pressure, rate composition etc. are taken into consideration during pseudo-steady state initialization.

### **Usage of DW Model**

The question may arise, when should one use the DW model? The answer is not simple and straightforward but the following points may be used as guidelines in the decision making process. A simpler Sink/Source well model may be adequate:

1. For reservoirs with reasonable injectivity where the effect of heat conduction between a wellbore and a reservoir is negligible. Injectivity is very low in heavy oil or tar sands reservoirs without bottom water and therefore oil may be initially mobilized only by heat conduction which is not possible with a Sink/Source model.
2. For processes with small flow rate or big pipe diameters where frictional pressure drop is almost nonexistent.
3. For short horizontal wells with a possibility of homogeneous fluid along a wellbore.
4. For homogeneous reservoirs where wellbore-reservoir communication is uniform.
5. For vertical wells where fluid segregation is minimal.
6. For reservoirs which have much higher draw-down than the expected friction pressure drop. One has to keep in mind that the absolute value of the frictional pressure drop is not so important as is the ratio of frictional pressure drop in the wellbore and pressure drop in the reservoir. It means that low frictional pressure drop may affect results when SAGD is used in very permeable thin reservoirs, but may not have a significant effect on thicker reservoirs with lower permeability.

For any other case the DW Model should be used. However, one has to be aware of possible numerical difficulties due to drastic PVT behavior and increased nonlinearities. A wellbore does not contain rock to buffer the effect of temperature. When pressure and temperature are close to saturated conditions then every small change in them will cause phases to appear or disappear. In reservoir, rock will absorb the marginal fluctuation in energy and therefore transition between phases is smoother. Sometimes it helps when tubing wall is also specified in the data and its heat capacity is entered through keyword \*ROCKCP.

Numerical difficulties also arise when:

- There is not a distinct upstream direction either in the wellbore or between a reservoir and wellbore grid blocks. This situation occurs mostly when the well is shut in at the sandface but the wellbore still communicates with the reservoir. This problem may be overcome 90% of the times by disconnecting the wellbore from a reservoir using TRANSWB WELBORE ... CON 0.0 when the well is shut in. TRANSWB must be reset to original values when the well is open. Wellbore properties are not correct when wellbore is disconnected from the reservoir and therefore this numerical problem treatment MAY NOT be used with \*TRANSIENT ON. However, the incorrect wellbore properties will be overwritten with pseudo-steady state values when \*TRANSIENT OFF is used and the well is open.
- Gas percolates to the top of a wellbore. It is more pronounced in producers. In this case saturation and possibly also mole fraction changes are very high and do not allow a timestep to increase. Most of the time it happens either during steam (gas) break through or around a well change time when the timestep is small. Possible cure - check the \*NORM values for mole fraction and saturation. These values are used in calculation of timestep size and may be too low for the large changes (dsmax, dymax, etc in the output), and do not allow the timestep size to increase. STARS uses double of specified \*NORM values for discretized wellbore blocks. When changing \*NORM does not help, then it is possible to reduce the wellbore velocity with a keyword \*TRANSI or \*TRANSJ or \*TRANSK depending on the wellbore direction. One has to be very careful and try different values of the transmissibility multiplier because production will be also affected. The goal is to overcome the numerical difficulties with the least possible change of physical results.

## Reference

1. 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.

# Appendix B: Data Sets

## B.1 Summary of Test Bed Data Sets

Test Bed data sets are used to illustrate and verify operation of the various features and options in STARS. These data sets may be found in directory "testbed" in the STARS release template area. Note that these data sets are designed to run without modifications. For example, commenting out \*MAXSTEPS 1 in a run designed to test an input or space allocation option may not result in a successful run, despite what recurrent data there is.

Data File	Description
sstst01.dat	Dry Combustion Tube
sstst02.dat	Wet Combustion Tube
sstst03.dat	Coats Lab Scale Steam Flood
sstst04.dat	Cartesian Grid as Corner-point (*COORD & *ZCORN)
sstst05.dat	Vapex with 2D Corner-point Grid (*DI, *DJ & *ZCORN)
sstst06.dat	Three Cycles of Huff'nPuff, SPE4 #1
sstst07.dat	Dead Oil Pattern Steam Flood, SPE4, #2
sstst08.dat	Live Oil Pattern Steam Flood, SPE4 #3
sstst09.dat	Variable Depth and Thickness Example
sstst10.dat	Dry Combustion Tube, with Combustion Water Component
sstst11.dat	Lab Scale Isothermal Emulsion Flood
sstst12.dat	Lab Scale Caustic-polymer Flood
sstst13.dat	Field Scale Surfactant Slug in North Sea Reservoir
sstst14.dat	Dead Oil Steam Flood with Corner-point (*COORD & *ZCORN)
sstst15.dat	Coats Lab Scale Steam Flood, ZH Solution Method
sstst16.dat	Cartesian Grid as *XCORN, *YCORN & *ZCORN
sstst17.dat	Lab Scale Steam Flood with Additives
sstst18.dat	Insitu Gelation
sstst19.dat	Lab Scale Pre-generated Foam Propagation, Chevron #1
sstst20.dat	Steam Injection with Wellbore Heatloss
sstst21.dat	Cartesian Grid as *DI, *DJ & *ZCORN
sstst22.dat	Coats Lab Scale Steam Flood; LGR Illustration
sstst23.dat	Field Scale Steam History Match and Foam Forecast
sstst24.dat	Testbed #6 with Lateral Aquifer
sstst25.dat	Lab Scale Steam Flood with Bottom Aquifer
sstst26.dat	Multi-layer Variable Depth and Thickness

No.	Description
sstt27.dat	Lab Scale Insitu-generated Foam Propagation, Chevron #2
sstt28.dat	Modified Kazimi's Dual Porosity Problem (*DUALPOR option)
sstt29.dat	Pruess Geothermal Problem on 1/8 5-spot (*MINC option)
sstt30.dat	Gravity Drainage Problem (*DUALPERM option)
sstt31.dat	Chen Problem (*SUBDOMAIN option)
sstt32.dat	Lab Scale Foam Run with Lamella Model, Chevron #3
sstt33.dat	SAGD with Discretized Circulating Injector and Producer
sstt34.dat	Modelling Near-well Phenomena with Hybrid Grid
sstt43.dat	Smaller Version of Test Bed No. 31
sstt46.dat	Test Bed #7 with 31x31x10 Full Pattern Grid
sstt47.dat	Test Zero-porosity Blocks and Full Printout
sstt48.dat	Dilation-Recompaction Option
sstt49.dat	Intermediate-Wet Relative Permeability Option *INTMED1
sstt50.dat	Oil-Wet Relative Permeability Option *OILWET
sstt54.dat	Model Metal Tube Wall with Zero Porosity and Radial Grid
sstt55.dat	Dilation and Thermal Conductivity Nine-Point Options
sstt56.dat	Foam with Horizontal Wells, Troll Field
sstt57.dat	Multiple Contact Miscible with STARS
sstt58.dat	Switch Between Mixed Surface/Downhole Constraints
sstt59.dat	Upper/lower Trans. Mult. For LGR Cartesian, redefined with time
sstt61.dat	Water/Oil Vertical Equilibrium Illustration
sstt62.dat	Water/Oil/Gas Vertical Equilibrium with Two Transition Zones
sstt63.dat	Steam Trap Well Control Option Illustration
sstt64.dat	Horizontal Discretized Wellbore Inside Hybrid Grid
sstt65.dat	Illustrate Foamy Oil Process – Bubbly Oil Approach
sstt66.dat	Illustrate Foamy Oil Process – Oily Foam Approach
sstt67.dat	Steam Trap Alternate Location Option
sstt68.dat	SAM Wellbore option with Source/Sink Wells
sstt69.dat	SAM Wellbore option with DW Tubing/Annulus
sstt70.dat	SAM Wellbore option with DW & Wellhead Constraints
sstt71.dat	Dead Oil Steam Flood with *FAULT
sstt72.dat	Vapex with 2D Corner-point Grid
sstt73.dat	Thermal Conductivity Options, Slaved Heater and Variable-Permeability
sstt74.dat	Elastic-Plastic Compaction-Rebound
sstt75.dat	Test/Illustrate BBM Kr Hysteresis (Water Wet)
sstt76.dat	Test/Illustrate Carlson Method for Kro, Krg and Pcoc Hysteresis

Data sets in directory "verify" in the STARS template release area are used to verify operation of some specific options and are not particularly useful for template purposes. Directory "restart" contains data sets that verify multi-segment restart option.

In directory "output" in the STARS template release area are summaries of the results of running all the data sets from "testbed", "restart" and "verify" on all the supported platforms. Included are numbers of timestep, iterations and timestep cuts, along with CPU time for the specific computer model indicated.

---

## B.2 Template Sample Data Sets

Another set of template data files can be found in the same template area, organized under the following directories which correspond to the following functional categories:

Drm	- Drive Mechanisms
Flu	- Fluid Types
Frr	- Fractured Reservoirs
Geo	- Geomechanics
Gro	- Grid Options
Hrw	- Horizontal Wells
Smo	- Simulator Options
Spe	- SPE Problems
Wwm	- Wells and Well Management

Each directory contains a text file which documents the data files in that directory. The main template directory “tpl” contains file “template.txt” which contains brief descriptions of all the data sets in these template directories. The following are the data files in these template directories that are not copies of the test bed data sets.

### Fluid Types

stflu018.dat	Inject Foam to Correct Early Gas Breakthrough for WAG
stflu019.dat	GLISP 3-D Foam History Match Study
stflu020.dat	Micellar-Polymer Drive in a Stochastic Reservoir (Mobilize Residual Oil with Capillary Number Dependent Kr)
stflu021.dat	Asphaltene Dropout with Primary Production (Plugging and Rate-Dependent Deposition)
stflu022.dat	Mole-based Version of stflu009
stflu023.dat	SI Unit Version of stflu022
stflu024.dat	Wolf Lake Pad Symmetry Element (Cyclic Steaming Phase)
stflu025.dat	Wolf Lake Pad Symmetry Element (Combustion Phase)
stflu026.dat	Steam Cycling in Initially Frozen Formation
stflu027.dat	Steam Flood Encounters Sub-Zero Cooling Well, *PERMTAB Case
stflu028.dat	Steam Flood Encounters Sub-Zero Cooling Well, *PERMCK Case
stflu029.dat	Steam Cycling with Outer Cooling Ring
stflu030.dat	Steam Cycling with Outer Cooling Ring & *AUTOCOOLER
stflu031.dat	Series of Interacting Cooling Wells
stflu032.dat	Test/Illustrate *WATPENTH
stflu033.dat	Test/Illustrate *INIT_FROM_RESTART, based on STFLU025
stflu034.dat	Test/Illustrate Shear Thinning with *SHEARTHIN
stflu035.dat	Test/Illustrate Shear Thickening with *SHEARTHICK
stflu036.dat	Test/Illustrate *SHEARTHIN & *SHEARTHICK together

stflu037.dat	Test/Illustrate Shear Thinning with *SHEARTAB
stflu038.dat	Test/Illustrate *VISCTABLE *ATPRES with *VISCTYPE, *WATPHASE, *OILPHASE
stflu039.dat	Test/Illustrate *TORTIKE_VG Steam Viscosity

### Fractured Reservoirs

stfrr006.dat	Illustrate/Verify SUBDOMAIN-DK Option: *TRANSD, *SD_REINF, *SD_HCMULT
--------------	--

### Geomechanics

stgeo002.dat	Elasto-plasticity with 2-D Radial Grid
stgeo003.dat	Sand Failure due to Pressure Drawdown
stgeo004.dat	Axi-symmetric Cyclic Steam Injection (4 Rock Types, Rigid Top)
stgeo005.dat	Single-Well Cold Flow with Sand Failure (2D Radial Model)
stgeo006.dat	Overburden Loads and Reservoir Body Force
stgeo007.dat	Geomechanics Domains and AIMSOL Matrix Solution
stgeo008.dat	Corner Point Grid Deformation with Displacement Vectors
stgeo009.dat	Corner Point Grid Deformation with Displacement Vectors (Two Geomechanics Rock Types)
stgeo010.dat	Axi-symmetric External Loads (Two Rock Types)
stgeo013.dat	Steam Cycling with Elastoplasticity
stgeo014.dat	3D Cylindrical Grid, Linear & Nonlinear 1 Elastic Constitutive Models
stgeo015.dat	3D Distributed Loads, Plot Grid Deformation, AIMSOL
stgeo016.dat	3D, Nonlinear 2 Elastic, Plot Stresses & Deformation
stgeo017.dat	3D, Initial Stresses, Dynamic Loading
stgeo018.dat	3D, Cap Model
stgeo019.dat	3D, Nonlinear elasticity, Hyperelastic model (compare with stgeo018)
stgeo020.dat	3D, Elasto-Viscoplastic Model, Flow Function 1
stgeo021.dat	3D, Elasto-Viscoplastic Model, Flow Function 2
stgeo022.dat	Test/Illustrate variable permeability options with *GEOMECH
stgeo023.dat	150k Grid Blocks, 4 Components, *GEOM3D, based on STGRO014
stgeo024.dat	Dilation Model, one way coupling without *PGDILA
stgeo025.dat	Dilation Model, one way coupling with *PGDILA
stgeo026.dat	Barton-Bandis Fracture Model, 5-Spot Pattern
stgeo027.dat	Geomechanics-Dependent Permeability
stgeo028.dat	Two Rock Types, without Porosity Calibration
stgeo029.dat	Two Rock Types, with Porosity Calibration
stgeo030.dat	Barton-Bandis Fracture Model, Horizontal Well with Overburden

stgeo031.dat	3D Radial Grid, Cap Model
stgeo032.dat	3D Radial Grid, Empirical dilation model, One-way Coupling
stgeo033.dat	3D Elasto-Plastic Druck-Prager Cap, Overburden, Grid Deformation
stgeo034.dat	Fluid/Heat Reservoir Embedded in Larger Geomechanics Grid
stgeo035.dat	Reservoir Not Embedded in Larger Geomechanics Grid
stgeo036.dat	Distributed Loads on Boundary Surface with *DLOADBC3D
stgeo037.dat	Distributed Loads on Boundary Surface with *DLOADBCIJK
stgeo038.dat	Infinite Mohr-Coulomb Medium, Dilation Angle
stgeo039.dat	Skempton and Mandel-Cryer Effects
stgeo040.dat	STGEO023 without *GEOM3D, fits Win32
stgeo041.dat	3D Single Surface Constitutive Model
stgeo042.dat	Infinite Mohr-Coulomb Medium, Associated SS Model
stgeo043.dat	Infinite Mohr-Coulomb Medium, Non-associated SS Model
stgeo044.dat	Porosity-Dependent Material Properties with *GRPORTAB
stgeo045.dat	Solid-Component Material Properties with *GEOSOLID
stgeo046.dat	Geomechanics Domain Pressure-Boundary with *BCDOMAIN
stgeo047.dat	Unload Stress for 3D Cylindrical Grid with *UNLOADSTR
stgeo048.dat	Generalized Plastic Model *GENPLAST, plus *DLOADBC3D
stgeo049.dat	Test/Illustrate *GRTEMTAB - T-Dependent Geomaterial Properties
stgeo050.dat	Test/Illustrate *GEOGRID *GCORNER with *GRID *CORNER
stgeo051.dat	Test/Illustrate *GEOGRID *CART with *GRID *RADIAL
stgeo052.dat	Test/Illustrate *GEOGRID *RADIAL with *GRID *CART
stgeo053.dat	Dean Problem #4, Water Flood - Single *CART Grid
stgeo054.dat	Dean Problem #4, Water Flood - Dual *CART Grids
stgeo055.dat	Dean Problem #1, Reservoir Embedded in Geomechanics Region
stgeo056.dat	Infinite Mohr-Coulomb Medium, Dilation Angle, Over/Underburden
stgeo057.dat	Generalized Plastic Model *GENPLAST with *COHESHARD and *FRANGSOFT
stgeo058.dat	Skempton and Mandel-Cryer Effects with Modified Cam Clay Model

### Grid Options

stgro007.dat	3x2 Patterns of 41x41x24 Each, ~1.5 Gb Total Storage; runs on 32-bit workstation with 1 Gb RAM; ~1/4 million blocks
stgro008.dat	4x3 Patterns of 41x41x24 Each, ~3 Gb Total Storage; runs on 32-bit server with 3 Gb process space and 2 Gb RAM; ~1/2 million blocks
stgro009.dat	Validate Operation of MRF Larger Than 2 GB
stgro010.dat	Restart from MRF Larger Than 2 GB

stgro013.dat	7x4 Patterns of 41x41x24 Each, 1 Million Blocks, ~7 Gb Total Storage; runs on 64-bit system with 4 Gb RAM
stgro014.dat	Single 41x41x24 grid with 40,344 blocks
stgro015.dat	9x6 Patterns of 41x41x24 Each, 2 Million Blocks, ~14 Gb Total Storage
stgro016.dat	28x28 Patterns of 41x41x24 Each, 30 Million Blocks
stgro017.dat	8-level LGR blocks with perfs and Special Histories
stgro018.dat	Recurrent Refinement and Derefinement
stgro019.dat	Recurrent Grid Amalgamation
stgro020.dat	Dynamic Refinement and Derefinement
stgro021.dat	Dynamic Amalgamation and Deamalgamation
stgro022.dat	Test/Illustrate Fault Options
stgro023.dat	5x4 Patterns of 41x41x24, 776k Blocks, 3 Gb Allocation, 2.4 Gb Usage
stgro024.dat	Test/Illustrate *NINEPOINT and *NINEPTH with *GRID *CORNER *TRANSF for a 3D Fault Surface
stgro025.dat	23x23=529 Patterns of 41x41x24, 20e6 Blocks, 120 Gb Alloc, 79 Gb Use
stgro026.dat	507x1 Patterns of 41x41x24, 20e6 Blocks, 120 Gb Alloc, 56 Gb Use
stgro027.dat	*TRANSF with *DUALPOR
stgro028.dat	*TRANSF with *MINC
stgro029.dat	*TRANSF with *DUALPERM
stgro030.dat	*TRANSF for a 3D Fault Surface with *SUBDOMAIN
stgro031.dat	250x1 STGRO014 Patterns, 10 Million Blocks
stgro032.dat	13x10 STGRO014 Patterns, 5 Million Blocks
stgro033.dat	Test/Illustrate Heatloss Boundary Searching, Without LGR
stgro034.dat	Grid Extension with *FLUIDHEAT, based on STFLU024
stgro035.dat	Test/Illustrate Corner-Point Grid With Discretized Wellbore
stgro036.dat	Test/Illustrate Corner-Point Grid Without Discretized Wellbore
stgro037.dat	Test/Illustrate Shale Properties - Base Case
stgro038.dat	Test/Illustrate Shale Properties
stgro039.dat	Test/Illustrate *DYNAGRID with Recurrent New Wells and Perforations
stgro040.dat	Test/Illustrate *SHAPE *K-HARMONIC with *DUALPOR
stgro041.dat	Test/Illustrate *SHAPE *K-HARMONIC with *MINC
stgro042.dat	Test/Illustrate *SHAPE *K-HARMONIC with *SUBDOMAIN
stgro043.dat	Test/Illustrate *TRANSMF with *MINC
stgro044.dat	Test/Illustrate *DUALPOR with *REFINE and *AIMSOL
stgro045.dat	Test/Illustrate *DUALPOR with *REFINE and *PARASOL
stgro046.dat	Test/Illustrate *DYNAGRID with Static Discretized Wellbore
stgro047.dat	

stgro048.dat	Test/Illustrate Static Followed by Dynamic Gridding
stgro049.dat	Test/Illustrate Heatloss Boundary Searching, With LGR
stgro050.dat	Test/Illustrate *SHAPE *PI2
stgro051.dat	Test/Illustrate Partial Inheritance for *FRFRAC and *FORMINFRAC
stgro052.dat	Diagonal 25x13 Grid for 1/8 9-Spot Pattern
stgro053.dat	Parallel 18x18 Grid for 1/8 9-Spot Pattern

### Horizontal Wells

sthrw007.dat	Steam Injection with 26 Wells Including 3 Horizontal Producers Variable Top, 2 Regional Aquifers, Actual 12-Year History
sthrw008.dat	LIAOHE Phase 3 – Dual Well SAGD in Dipping Reservoir
sthrw009.dat	Single-Well SAGD – 800 m Discretized Wellbore in Hybrid Grid

### Simulator Options

stsмо010.dat	Water/Oil/Gas Ver. Eq. with Oil on Bottom, $P_c = 0$
stsмо011.dat	Oil/Water/Gas Ver. Eq. with Oil on Bottom, Non-Zero $P_{cow}$
stsмо012.dat	Water/Gas Vertical Equilibrium Initialization
stsмо013.dat	G-L and L-L Surface K Values
stsмо014.dat	*AQFRCOMP and Orig/Aqfr/Injt Water Components
stsмо015.dat	Test/Illustrate Killough Kr Hysteresis (Water Wet)
stsмо016.dat	Test/Illustrate *KRTYPE and *KRTYPE_VERT
stsмо017.dat	Test/Illustrate *KRSWITCH and Associated Keywords
stsмо018.dat	Base Case for Testing Array-Reading Option *BINARY_DATA
stsмо019.dat	Test/Illustrate Array-Reading Option *BINARY_DATA
stsмо020.dat	Base Case for Testing *BINARY_DATA for Natural Fracture
stsмо021.dat	Test/Illustrate *BINARY_DATA with Natural Fracture
stsмо022.dat	*LININTERP with Rel Perm Hysteresis of Killough
stsмо023.dat	Illustrate keyword equivalent of ‘-doms –parasol’
stsмо024.dat	Illustrate *SOLVER *PARASOL with *PPATTERN
stsмо025.dat	Illustrate *SOLVER *PARASOL with *PPATTERN *AUTOPSLAB
stsмо026.dat	Illustrate *SOLVER *PARASOL with *PPATTERN *PARTITION
stsмо027.dat	Illustrate *SOLVER *PARASOL with *PPATTERN *PPARTITION
stsмо028.dat	Illustrate *SOLVER *PARASOL with *PPATTERN *GPARTITION
stsмо029.dat	Illustrate *SOLVER *PARASOL with *PPATTERN *APARTITION
stsмо030.dat	Illustrate *SOLVER *PARASOL with *DTYPE
stsмо031.dat	Illustrate *SOLVER *PARASOL with *DPLANES
stsмо032.dat	Illustrate *SOLVER *PARASOL with *CHECKRB

stsмо033.dat	Illustrate *SOLVER *PARASOL with *PDEGAA and *PDEGAB
stsмо034.dat	Illustrate *SOLVER *PARASOL with *PNPROSL
stsмо035.dat	Test/Illustrate *OILWET with Vert. Eq. WO Trans. Zone and *WOC_SW
stsмо036.dat	Water-Gas System Vert. Eq. with *TRANZONE and *WOC_SW
stsмо037.dat	Test/Illustrate *KRTYPE_CTRWAT/OIL/GAS with 2-D SAGD
stsмо038.dat	Test/Illustrate *PORINTERP *REF - Static Grid
stsмо039.dat	Test/Illustrate *PORINTERP *INIT - Static Grid
stsмо040.dat	Test/Illustrate *PORINTERP *INIT - Dynamic Grid
stsмо041.dat	Test/Illustrate Over/Underburden Heatloss Through Null Blocks
stsмо042.dat	Heater Well - *HTWELL *HTWRATE replacing *HEATR
stsмо043.dat	Heater Well - Heat-only Multi-block Well with *HTWRATEPL
stsмо044.dat	Test/Illustrate *HTWELL with convective/max heat rate
stsмо045.dat	Test/Illustrate Heater Well in Cooling Mode with *PERFV *WI
stsмо046.dat	Test/Illustrate Heater Well Through Pinched-out, Null and Zero-Porosity
stsмо047.dat	Test/Illustrate *LAYERXYZ, Anisotropic/Nonuniform Thermal Cond. and Heat Rate Cut-off
stsмо048.dat	Multiple Heaters; Completion Length Modifiers; Shut-in Layers
stsмо049.dat	Oil-Wet Pc Hysteresis with Non-Wetting Water Kr Hysteresis (BBM)
stsмо050.dat	Oil-Wet Pc Hysteresis with Non-Wetting Water Kr Hysteresis (Killough)
stsмо051.dat	Test/Illustrate *AIM *STAB *BACK in Recurrent Data Only
stsмо052.dat	Test/Illustrate *DATUMDEPTH *INITIAL
stsмо053.dat	Test/Illustrate *DILATION with Thermal Effects
stsмо054.dat	Test/Illustrate *EPCOMPACT with Thermal Effects
stsмо055.dat	Test/Illustrate *COMPACT_VAR Rock Model
stsмо056.dat	Test/Illustrate *PORFORM *EXP Porosity Formula
stsмо057.dat	Test/Illustrate *VOLCONST *BULK Volume Constraint Type
stsмо058.dat	Illustrate/Verify *INIT_FROM_IMEX

## Wells and Well Management

stwwm009.dat	4-Well Waterflood (Group Control, Voidage Replacement, Work-Over)
stwwm010.dat	Combined Gas Cap-Solution Gas-Aquifer Drive, Gas Cycling, Well Workover, Group Well Control
stwwm011.dat	Gas Recycling with Additional Makeup Gas
stwwm012.dat	Miscible Flood with Gas Cycling and Group Control
stwwm013.dat	Miscible Flood with Nested Group Control and Autodrill
stwwm014.dat	Miscible Flood with Group Control and Autodrill
stwwm015.dat	Miscible Flood with Group Control, Autodrill, Gas Re-injection

stwwm016.dat	sttst08 with *LAYERXYZ
stwwm017.dat	stflu018 with Multilateral Well
stwwm018.dat	stwwm017 with Null Layers
stwwm019.dat	schrw007 with Source/Sink Wellbore Friction and Heat Loss
stwwm020.dat	stwwm008 with Source/Sink Wellbore Friction
stwwm021.dat	sttst70 with Source/Sink Friction/Heat Loss instead of Disc. Well
stwwm022.dat	Test/Illustrate Fully-Mixed Crossflow Option
stwwm023.dat	Test Completions Through Pinched-out, Null and Zero-Porosity Blocks
stwwm024.dat	Test/Illustrate *STO_COMP and *STG_COMP
stwwm025.dat	Test/Illustrate Multi-Phase Co-Injection Options
stwwm026.dat	Test/Illustrate Multi-Phase Co-Injection Options with S/S *SAMODEL
stwwm027.dat	Test/Illustrate Multi-Phase Co-Injection Options with DW *SAMODEL
stwwm028.dat	Steam Injection with S/S Limited Entry Perforations (LEP)
stwwm029.dat	Well before type defined, *WLISTSHUT, Inactive layers in RESULTS
stwwm030.dat	Semi-Analytical Wellbore Heatloss with PUMP option
stwwm031.dat	Steam Injection with DW Limited Entry Perforations (LEP)
stwwm032.dat	Steam Injection with Discretized Wellbore in Corner-Point Grid
stwwm033.dat	Verify/Illustrate *WELBORE-REC – Change DW to S/S
stwwm034.dat	Verify/Illustrate *WELBORE-REC – Add New DW
stwwm035.dat	Verify/Illustrate *WELBORE-REC – Shorten Tubing, Add New DW
stwwm036.dat	Group/Well On-time Fraction & Group Target Apportionment
stwwm037.dat	Matrix/Fracture Qualifiers in Perforation References
stwwm038.dat	Fluid Redistribution Using *MODELSHUT and *DWA
stwwm039.dat	Fluid Redistribution in a Communicating Well with *DWB
stwwm040.dat	4-Well Waterflood with Group Control and Sector Pressure Maintenance
stwwm041.dat	Group Control Steam Cycling *GCONCYC_START/END
stwwm042.dat	Verify/Illustrate Unlimited Number of Group Levels
stwwm043.dat	Verify/Illustrate Compositional Water Recycling with Group Control
stwwm044.dat	Trigger Options *ON_ELAPSED_TIME and *ON_SECTOR
stwwm045.dat	Trigger Options *ON_ELAPSED_TIME, *ON_SECTOR, *ON_WELL
stwwm046.dat	Test/Illustrate Isothermal *PTUBE1 with WHP Constraint
stwwm047.dat	Miscible Flood with Nested Group Control and Autodrill
stwwm048.dat	Verify/Illustrate *TARGET *STEAM Feature – Base Case
stwwm049.dat	Verify/Illustrate *TARGET *STEAM Feature

stwwm050.dat	Test/Illustrate Gas Injection with *MANIFOLD, *ITUBE1 and *PTUBE1 with ALQ Values
stwwm051.dat	Illustrate/Verify *MONITOR *WHP *AUTOWELL
stwwm054.dat	Group Control with STO Group Target
stwwm055.dat	Group Control with STF Group Target
stwwm059.dat	FlexWell with T/A, SAGD Production- Annulus Only
stwwm060.dat	FlexWell SAGD Production in Tubing & Annulus
stwwm061.dat	FlexWell with T/A, Variable Tubulars, SAGD Production in T/A
stwwm062.dat	FlexWell with T/A, SAGD Production, SAM - Annulus Only
stwwm063.dat	Replace S/S *GRAV-FRIC-HLOS Well with FlexWell - Cyclic steam injection
stwwm064.dat	FlexWell, Multi-Phase Co-Injection, SAGD Annulus Production Only
stwwm065.dat	FlexWell, Multi-Phase Co-Inj, SAGD T/A Production, Operating Constraint
stwwm066.dat	FlexWell, Multi-Phase Co-Injection, SAGD T/A Production
stwwm067.dat	Miscible Flood with Reference Volume Replacement
stwwm068.dat	Flexible Wellbore with *PACKER and *FCD-ORIF for Annulus
stwwm069.dat	Flexible Wellbore with *PACKER and *FCD-ORIF for Tubing
stwwm070.dat	Flexible Wellbore with *REPLACE
stwwm071.dat	SAGD with Flexible Wellbore and Instrumentation Tubing
stwwm072.dat	Steam Cycling with Flexible Wellbore and Instrumentation Tubing

#### **Electrical Heating (directory /electric)**

elec1.dat	Test/Illustrate *ELWCOMPTAB and *ELSCOMPTAB
elec2.dat	Test in 3D Cartesian Mode
elec3.dat	Test/Illustrate Metal Electrode and Constraint Switching
elec4.dat	Test/Illustrate the *NOFLASH option
elec5.dat	Test/Illustrate Multi-Well Hybrid Grid
elec6.dat	Electrical Heating Data #3 with Multiple Phases
elec7.dat	Test/Illustrate Isolated Three-Phase Triangular Configuration

# Appendix C: Advanced Processes

## Overview

This appendix contains detailed descriptions of advanced oil recovery processes that may be modelled by STARS, divided into the following sections.

- C.1 Hot Water Flooding Process
- C.2 Steam Flooding Process
- C.3 Steam Cycling Process
- C.4 Fire Flood Process
- C.5 Additives Overview
- C.6 Gas, Water and Oil Phase Tracers
- C.7 Gas Additives
- C.8 Water-Rock Chemical Interactions
- C.9 Polymers and Gels
- C.10 Surfactant and Caustic
- C.11 Fines and Emulsions
- C.12 Oil Additives and Partitioning Inversion
- C.13 Foam

---

## C.1 Hot Water Flooding Process

This process and its mechanisms can be found taking place in each of the other processes, and so it is the most basic. The following are mechanisms key to the hot water flooding process. See chapters 5 and 6 of Reference 1 for a more complete description.

Convective and conductive heat transfer in the formation will control the placement of heat, the driving force of thermal EOR processes. Heatloss to the adjacent formation can dominate the energy balance equation, so accurate estimates of heatloss are crucial<sup>2</sup>. Oil viscosity reduction with increasing temperature may be the dominating recovery mechanism. Heavy oil viscosities may decrease several orders of magnitude with a 200°C rise in temperature, decreasing WOR by the same ratio. Banking and plugging of oil can be beneficial or harmful. Banking occurs because oil viscosity decreases with increasing temperature. A plug is a low-mobility oil bank which forces the water saturation down to the immobile value, causing a large decrease in total fluid mobility. Water/oil relative permeabilities, together with viscosities, determine the in situ flowing and producing water-oil ratio WOR. Relative permeability dependence on temperature has been attributed to changing interfacial tension between the water and oil phases<sup>3,4,5,6</sup>.

---

## C.2 Steam Flooding Process

The steam flooding mechanisms will also occur as part of the cyclic steam and fire flood processes. In addition to the hot water process, the following mechanisms are important. See chapter 7 of Reference 1 for a more complete description.

Steam fluid properties such as vapor pressure, density and enthalpy will dominate. In particular, the latent heat of steam provides a significant excess source of energy over that of hot water at the same temperature leading to faster heat propagation<sup>7</sup>. But because conductive heat losses at steam temperature are supplied by the latent heat, the steam propagation rate slows down after a period of steam injection<sup>8</sup>. As well, this leads to the formation of a hot water condensate zone ahead of the steam front.

Oil relative permeability as a middle phase (assuming water wet rock) is estimated from two-phase data via Stone's formula<sup>9</sup>. This affects oil production directly. Gas/liquid relative permeabilities will help determine the in situ flowing and producing gas-liquid ratio, (e.g. GOR), which affects delivery of steam and gaseous additives. Gravity override and drainage become important for thicker formations<sup>10</sup>. The presence of gas (steam) with its very low density is the cause. Fluid banks possible include a steam bank, a hot water bank, a cold water bank, and an oil bank. Steam distillation of volatile hydrocarbons may create a solvent bank which can help mobilize native oil<sup>11</sup>. Heat scavenging applied to a more mature steam flood would involve injecting cooler steam and then cold water. Heat lost to the adjacent formation will flow back into the cooled reservoir.

---

### C.3 Steam Cycling Process

Also known as Huff 'n' Puff, as well as steam stimulation, steam cycling involves alternately injecting and producing in a single well. This technique is popular in fields whose oil mobility is too low to begin steam flooding immediately. In addition to the hot water and steam flood mechanisms, the following items are important. See chapter 9 of Reference 1 for a more complete description.

Low interwell communication results from heavy oils whose mobility is so low that excessive injection pressure is required to achieve any production. Steam cycling applied to each well will improve interwell communication to the point where steam flooding can be done. Fracturing may be required to inject steam, in light of possible oil plugging<sup>12</sup>.

Once cycle consists of injection, an optional soak or shut-in period, and production of the well. Optimum slug size can be considered when designing the cycles. A slug smaller than optimal does not mobilize (i.e. heat) as much incremental oil. A slug larger than optimal heats more oil than can be produced by drainage or by drive energy during that cycle. Compaction drive can provide drive energy due to deformation of the heated zone<sup>13,14,15</sup>, especially in early cycles. Cycle to drive conversion may occur when the interwell communication is sufficient, or when it is optimal.

---

## C.4 Fire Flood Process

Also known as in situ combustion, this process involves the injection of air or enriched air, which burns fuel in the formation. The technique may be viable in formations which are so thin that excessive heat losses make steam flood unfeasible. In addition to the hot water and steam flood mechanisms, the following points are important. See chapter 8 of Reference 1 for a more complete description.

In situ steam generation by burning part of the native hydrocarbon with oxygen is the key mechanisms. Fluid banks possible in fire flooding are air zone or burned zone, burning zone, coke or fuel bank, distillation zone, steam bank, distilled oil bank, hot water bank, native oil zone and combustion gases. Distillation of lighter hydrocarbons forms light oil and solvent banks. Immobile residual hydrocarbon, or fuel, is burned with injected oxygen. Fuel or coke is the results of cracking chemical reactions. Oxidation reactions provide the heat that drives the process. High-temperature<sup>16</sup> and low-temperature oxidation<sup>17</sup> chemical reactions can take place. Injected water can control temperatures and heat distribution in the formation.

Process status can be monitored by examining reaction products such as fresh water and ratios of CO to CO<sub>2</sub>. Combustion tubes in the laboratory can provide fine tuning of process parameters, as well as a go/no-go indication.

A field pilot project in Alberta<sup>18</sup> demonstrates many practical aspects of steam cycling, conversion to steam drive and conversion to in situ combustion processes.

## C.5 Additives Overview

Injection of steam is, in itself, an efficient method for mobilizing and producing heavy oil or bitumen, due to the large oil viscosity reduction observed at higher temperatures. However this effectiveness is often lost if steam does not contact large portions of the oil reservoir or if no flow paths exist for the mobilized oil to the producing wells.

Additives are injected with steam to either enhance the basic efficiency of the process or to improve conformance control. In the first category, gaseous additives such as CO<sub>2</sub> or CH<sub>4</sub> are employed to increase the size of the steam zone and to further reduce oil viscosity due to solubilization effects. Liquid based additives (naphtha, caustic, surfactants) can also reduce oil viscosity and/or residual oil saturation. Injected emulsion, foams and gels are being considered as conformance control agents either to block high permeability (thief) zones or to attempt to reduce the gravity override of steam.

The success of these additive processes is often dominated by water-rock chemistry. High temperature, high pH, and salinity changes can induce mobilization and plugging by fines; adsorption and thermal degradation of surfactants and polymers can limit the amount of additives that the reservoir actually sees. As illustrated in later sections, water chemistry can affect the performance of not only aqueous additives but those added to oil and gas phases as well.

Enhanced oil recovery process in the North Sea yields examples of thermal additive processes where steam does not need to be considered. Here injection of cold sea water (~5°C) into warmer reservoirs (~90°C) results in an "inverse" temperature gradient. Temperature effects, although not extreme, must still be considered on additive properties and additive thermal stability. Thus, temperature distributions occurring in the reservoir after several years of cold water injection must be accounted for.

While the previous chapter viewed temperature changes as the dominant feature to be modelled, in this section temperature is mostly secondary in the sense that it modifies processes which can occur at constant temperature as well. As a consequence, much useful information can be gained by isothermal simulations of these additive processes and have application in strictly compositional EOR processes such as CO<sub>2</sub> flooding.

---

## C.6 Gas, Water and Oil Phase Tracers

The use of chemical or radioactive tracers is an important experimental means of establishing the separate, tortuous flow paths of gas, water and oil phases as they flow through a porous medium<sup>19</sup>. The most important idea is that this information is scale dependent<sup>20,21</sup>, and useful data emerges at any scale, from small cores to field wide levels. Tracer profiles are normally analyzed in terms of effective dispersion coefficients for unit mobility displacement. The influence of permeability variation (as measured, for example by Dykstra-Parsons coefficient) and auto correlation lengths on these effective coefficients has been the subject of renewed research interest.

Tracers are normally chosen such that they partition into one phase only and which don't interact in any way with the other phases present (e.g. don't adsorb on the rock or don't plug flow pathways, etc.). In specific cases, tracers partitioning into another phase can prove useful - e.g. radioactive water injected as steam can also be analyzed in produced water and steam phases. As referenced above, tracer analysis has been primarily employed to yield well to well communication information. However, single well tracer studies are also used to determine residual oil saturations. Finally, it is noted that temperature is a "tracer" for energy flow, and thus thermal conductivity can also be scale dependent.

---

## C.7 Gas Additives

The addition of light hydrocarbons (methane, ethane) and/or other gases (e.g. CO<sub>2</sub> or N<sub>2</sub>) to improve the efficiency of both cycle steam or steam drive processes has been investigated for some time<sup>22,23</sup>. The general conclusion is that small concentrations of gaseous additives can alter and improve both early and ultimate recovery of heavy oil, on the order of 10-20% in favorable situations. However the addition of too large an amount of additive again reduces the efficiency.

The basic mechanisms are thought to be swelling of the oil, viscosity reduction and solution gas drive. It is observed that the additives tend to exist primarily at the boundary of the steam chamber where they expand the size of the steam chamber but lower the steam temperature (because of partial pressure effects). It is also in this region where significant partitioning into the oil and water phases occur. The detrimental effects of too much additive are caused by either the reduction of the temperature of the gas phase to inadequate levels and/or the hindrance of the flow of movable oil caused by a too high gas saturation. If the heavy oil already contains a significant amount of dissolved gas, the beneficial effects of gaseous additives to steam can be severely retarded.

The above mechanisms can be modelled via appropriately chosen p,T (and possibly composition) dependent K values describing solubility, as well as compositionally dependent densities and viscosities. The addition of CO<sub>2</sub> at high temperature involves further problems because of its solubility in and reaction with chemicals in the water and the rock. This reactivity is described in more detail below.

In the context of (isothermal) compositional simulation, the above mechanisms indicate that the process is operating far from miscibility conditions (where vaporizing or condensing gas drive mechanisms would come into play). This is primarily due to the nature (composition) of the heavy oil. Application of steam recovery processes to lighter oil systems requires that these more involved compositional effects be evaluated<sup>24</sup>. In particular, steam distillation of the oil plays a dominant role - a thermal vaporizing gas drive mechanism.

---

## C.8 Water-Rock Chemical Interactions

This subsection presents a brief overview of rock matrix and clay reactivity which can dramatically influence the effectiveness of any additive process. In general terms, rock reactivity can change the ionic environment in which specific additives must operate and produce changes in porosity and permeability. Most reactions are accelerated at higher temperatures such as those occurring under steam flood or combustion conditions.

Application of these ideas to specific additive processes are dealt with in later subsections.

To a first approximation, reservoirs can be classified as primarily sandstone or limestone (carbonates and dolomite) types, whose primary components - quartz (silica) and CaCO<sub>3</sub> or MgCO<sub>3</sub> respectively - determine their reaction chemistry. More particularly, for sandstone the main dissolution/precipitation reaction can be modelled as<sup>25</sup>



which is strongly temperature dependent. Thus, at stream temperatures, the amount of dissolved silica can be quite significant<sup>26</sup>. This has important implications for porosity and permeability changes, as well as silica fines transport.

Silica chemistry is quite complex<sup>27</sup>, with the type and amount of aqueous silica species present varying widely as pH and temperature conditions change. In particular, injection of pH altering additives such as caustic or even CO<sub>2</sub> can be expected to affect the observed silica species (see the section on caustic flooding). For carbonate reservoirs, the important reaction is the dissolution of CaCO<sub>3</sub> (or MgCO<sub>3</sub>).

In addition to the fundamental rock matrix constituents mentioned above, most reservoirs have nonuniformly interspaced regions of various clays and shales (i.e. consolidated clay) which can also affect rock reactivity and rock transmissibility. The low ion exchange clays (kaolinite) are most mobile and are sensitive to release via changes in ionic environment (brine salinity). The high ion exchange capacity clay montmorillonite is a swelling clay which can be treated with potassium hydroxide (KOH)<sup>28</sup>. In either situation (mobile or swelling), the net result is permeability damage when fresh water is injected into a saline reservoir. In thermal processes, another source of fresh water is condensed stream.

Another phenomena associated with clays is ion exchange in which injected ionic compositions can change as cations such as Na<sup>+</sup> and Ca<sup>++</sup> are exchanged with their bound counterparts on the clays<sup>29</sup>. Limestone dissolution can also affect ion exchange behavior.

The matrix and clay reactivities outlined here (dissolution, precipitation, mineral conversions, and ion exchange) have been shown to be strongly temperature dependent. Equally important are the effects of water pH, and a discussion of this is given in the section of caustic injection processes.

## C.9 Polymers and Gels

Polymers are mobility control agents employed to improve microscopic (1d) displacement efficiency and 2d and 3d sweep efficiency<sup>30</sup>. They improve mobility ratio by a combination of increased water phase viscosity and decreased effective permeability by blocking of pathways. Their propagation characteristics are further affected by dispersion, adsorption and inaccessible pore volume (early polymer breakthrough)<sup>31,32</sup>. Recent work has emphasized the important role of viscous cross flow in the analysis of polymer flow patterns in heterogeneous reservoirs, including its effect on polymer slug breakdown<sup>30,33</sup>.

There are two general categories of polymer: Synthetic ionic (e.g. partially hydrolyzed polyacrylamides (PHPA) such as Dow Pusher 700) and biopolymers (e.g. xanthan gum) which differ in their molecular and flowing properties. In particular, xanthan is a rigid rod type molecule, almost helical in structure, which has very good viscosity modification properties. In contrast, PHPA is a flexible polymer whose viscosity is sensitive to salinity levels<sup>30</sup>, decreasing as salinity increases because the ionic side chains become electrically screened and the molecule folds back on itself. The synthetic polymer operates primarily as a pore blockage agent and is more sensitive to degradation. Stability questions are especially important at the high temperatures occurring in a steam flood.

Polymer viscosity is non-Newtonian (shear rate dependent) although the underlying physical mechanisms causing this behavior can differ. Normally this shear dependence is of importance only in the high flow regimes around wells.

Polymer flooding has been applied with success in many field projects. Newer applications include its use in conjunction with surfactant or caustic flooding, as discussed below.

In recent years, considerable interest in more complex means of remedying macroscopic reservoir heterogeneity has arisen, involving in-situ gel formation<sup>33</sup>. Gellation involves the injection of low viscosity fluids which react to form essentially immobile networks to block high permeability channels. Various gel systems have been investigated including silicate gels, polymer gels and phenylformaldehydes.

To date the most extensive work has been done on polymeric gel systems involving either synthetic (PHPA) or biopolymer (xanthan) and some cross-linking agent such as CrCl<sub>3</sub> or AlCl<sub>3</sub>. The kinetics of gelation itself are quite complex<sup>34</sup>, and the further complications associated with the availability of cross-linking agent include additional oxidation-reduction reactions to delay gel formation, ion exchange reactions of Cr+3 with the rock and possible precipitation of cross-linker at basic pH levels.

Recent work on gels suitable for steam flood applications indicate that phenoformaldehyde gels are much more promising<sup>35</sup>.

## C.10 Surfactant and Caustic

Surfactants are species which tend to exist at the interface between water and oil phases, lowering interfacial tension and promoting mixing between the two phases<sup>36</sup>. Surfactants can be ionic (usually negatively charged, e.g. Witro 7RS-10-80) or nonionic (e.g. Triton X-100). Surfactants are sensitive to both ionic composition and temperature, such that partitioning can change from predominately water based to oil based as salinity or temperature (for nonionics) are increased. (Ionic surfactants become more water soluble with temperature). Interfacial tension lowering capabilities change with partitioning - the lowest interfacial tension normally occurs around equal water/oil partitioning. Surfactants also strongly adsorb on rock with adsorption decreasing as salinity or temperature increase<sup>37</sup>. Most often, surfactant mixtures are injected and chromatographic separation of components can occur as the injected slug propagates due to differing component levels of partitioning and adsorption<sup>38</sup>. The chromatographic separation of salt and pH alternating components which may be injected with or ahead of the surfactant mixture are discussed in other sections.

The reduction of interfacial tension by surfactants is usually correlated with decreased residual oil (and connate water) through the calculation of a dimensionless capillary number describing the balance between viscous and interfacial forces<sup>39</sup>. In addition, the presence of surfactants can alter rock wettability, usually interpreted as a change in the curvature of water and oil relative permeability curves. The net result is a change in the flow characteristics of the phases as surfactant concentration increases to almost miscible type (straight line) relative permeability curves<sup>40,41</sup>.

The mechanism of caustic injection is thought to occur through the formation at high pH of an in situ surfactant, resulting from a reaction with naturally occurring (acid) species in the oil<sup>42</sup>. Various caustic species have been investigated<sup>43</sup> including sodium hydroxide (NaOH), sodium orthosilicate (Na<sub>4</sub>SiO<sub>4</sub>), and sodium carbonate (Na<sub>2</sub>CO<sub>3</sub>). Propagation of an alkali slug is affected by sodium/hydrogen and calcium/hydrogen ion exchange, rock dissolution reactions, and precipitation<sup>44</sup>.

A modified system that shows extreme promise, including encouraging field pilot results, is polymer-augmented caustic flooding<sup>43,45</sup>, where a synergistic effect of combining the two additives has been noted.

The effect of elevated temperature on caustic-sandstone interaction is primarily detrimental in the sense that both ion exchange and silica dissolution reaction increase, leading to an overall increase in caustic consumption<sup>46</sup>. High temperature caustic flooding or the use of caustic as a steam additive have also been considered<sup>47,48,49</sup>, even after the detrimental effect of temperature on caustic consumption was realized. Basically, because of the ease of emulsification at higher temperature and the effect of temperature on oil viscosity, this processes remains attractive.

## C.11 Fines and Emulsions

The propagation of small solid particles (fines) through a porous media is described mathematically as filtration<sup>50,51,52</sup> theory. These particles can have many sizes, shapes and chemical origin. However, the major particle types are mobile clays (especially kaolinite), silica fines (from the dissolution of quartz), and scale formation precipitates (e.g. CaCO<sub>3</sub>, Ca(OH)<sub>2</sub>). In a multiphase situation, these particles can propagate primarily either in the water or oil phases, depending on their wettability.

The basic laws of fines propagation and capture can be expressed in terms of a filtration coefficient (inverse of average propagation distance) and plugging coefficient (describing the permeability decrease due to particle capture). There are many postulated modes of particle capture, depending on particle size, flow velocity, and salinity which result in differing theoretical expressions for these coefficients. The sensitivity to water salinity should be particularly emphasized<sup>53</sup>.

Although other descriptions have been proposed, the flow of dilute, stable emulsions can be most satisfactorily modelled employing a variant of filtration theory<sup>54</sup>. Here, the oil droplets in an (e.g. oil-in-water) emulsion system act as (deformable) particles which propagate and are captured similar to fines particles. Again, porosity and permeability changes must also be considered. Differences however do exist in the details of the capture process. Basically, experiments indicate an upper limit to emulsion capture while solids can capture on top of each other (dendrite formation). This leads to a different dependence of filtration coefficient on captured droplet concentration and an observed steady state concentration profile for produced emulsion. Furthermore, the permeability alterations are much less severe in the emulsion case, for equally sized emulsion and fines particles.

Emulsions are unstable dispersions and break over a suitable time frame<sup>55</sup>. This can be especially important as they flow through a porous medium because of the strong interaction between the liquid and solid phase. In situ emulsion coalescence has been studied by several workers<sup>56</sup>, while inset formation is usually thought to occur via a snap off mechanism<sup>57</sup>. At higher emulsion concentrations, non-Newtonian viscosity effects become important. These ideas also pertain to foams since in the limit of high concentrations, emulsions and foam behave similarly.

Some recent laboratory studies have investigated the use of emulsion blocking agents for steam flooding applications<sup>58</sup>. Chung and Butler<sup>59</sup> have noted the large amount of water-in-oil emulsification obtained above a rising steam chamber and have verified the role of steam condensation of gas droplets in their formation.

---

## C.12 Oil Additives and Partitioning Inversion

Many of the topics treated under the general heading of aqueous phase additives have their counterparts as oil additives. Indeed, changing conditions of pH, salinity or temperature can often result in partitioning inversion, which chemicals (especially surfactants) can go from being strongly water soluble to strongly oil soluble. This is equally true for both injected surfactants and naturally occurring surfactants (originally dissolved in the oil) and can dramatically alter surfactant, caustic, emulsion and foam processes. Studies have demonstrated the connection between inversion<sup>60</sup> and the possibility of the formation of a third liquid ("microemulsion") phase as pH, salinity, or temperature are changed. In fact one EOR process (termed microemulsion flooding - see above) is designed to operate as much as possible in this three liquid phase environment. Normally, however, this extreme situation occurs over a very limited range of conditions, and when surfactant concentrations are low, this is very hard to detect. Still this three phase region represents the thermodynamic driving force for inversion. In addition to the essentially mechanical aspects of emulsion instability, emulsions are also unstable to inversion - interconverting from oil-in-water to water-in-oil as conditions of temperature, salinity, pH, or volume fractions of oil and water change.

Except under the above special, near miscible conditions, water displaces oil immiscibly. Thus, mobility control is achieved either by adding thickeners to the water phase (e.g. polymers and gels) or diluents to the oil phase (e.g. naptha and/or condensable gases). In this latter instance, oil additives are almost always hydrocarbon based, and concepts from oil/gas compositional modelling (e.g. through an equation of state) apply. For the heavy oils considered in most thermal projects, the compositional mixing effects are operating very far from oil/gas miscibility conditions. However, in more recent steam flooding applications of light reservoirs, this may not be the case.

## C.13 Foam

Foam is a gas phase mobility control process which is gaining increasing popularity for both steam flooding<sup>61,62,69</sup> and compositional (e.g. CO<sub>2</sub>)<sup>63</sup> processes. Although very complex, recent research into the process mechanisms has made simulation feasible<sup>64</sup>. In favorable circumstances, mobility reduction factors of 100 have been reported.

In developing a quantitative understanding of foam flow, several aspects must be considered. The first is surfactant stability<sup>65</sup> and propagation/retention<sup>66</sup>. Three general classes of surfactants have been considered-aromatic (alkylaryl sulfonates (AAS) such as Sun Tech IV), alphaolefin sulfonates (AOS) such as Chevron Chaser SD1000), and nonionics and ethoxysulfonates. The primary consideration in lower temperature regimes is surfactant adsorption while at higher temperatures (arising in steam flood application), thermal stability questions are dominant. Results show that AAS type surfactants adsorb more than AOS and are more strongly affected by salinity. For both surfactants, adsorption decreases with temperature and increases as the clay content of the sand increases. Conversely, AAS are superior to AOS in terms of thermal stability (although the AOS dimer Chevron SD1000 does appear to be thermally stable). Decomposition of surfactants is acid catalyzed and is therefore strongly dependent on pH - in basic solutions this hydrolysis reaction is normally first order in surfactant concentration. Nonionics and ethoxylated sulfonates have very poor temperature stability and can therefore only be considered for CO<sub>2</sub> type foam applications. Surfactant propagation can also be (adversely) affected by cation exchange between surfactant solution and formation clays. High salinity preflushes can improve this behavior. Normally, foam forming surfactants are chosen to be strongly water soluble, so that partitioning into the oil phase is minimized. However, both temperature and ionic concentration can affect this partitioning.

Of particular importance is the (normally detrimental) effect of the presence of oil on foam formation and propagation. Indeed only one system has been reported in which the presence of an oil phase has had a positive effect on foamability. Oil-foam interactions are complex, and the presence of oil can be detrimental to foams in several ways<sup>67,68</sup>.

Field applications of foam appear promising, especially in thermal projects, where several successful pilot trials<sup>61,62,69</sup> have been reported. As an alternative to water-alternating with gas (WAG) processes for mobility control for gas flooding projects, the situation is less clear. Obviously, foam application to plug near well, isolated thief zones have a higher probability of success than treatments designed to counteract gravity override deep into the reservoir. Evolving criteria<sup>61</sup> to ensure successful operation include addition of salt to the brine to counteract clay ion exchange and the effect of Ca<sup>++</sup> ions. This allows faster surfactant propagation into the reservoir. Major explanations for failed foam tests include insufficient length of injected surfactant slugs and poor reservoir characterization - especially in the size of the region to be treated.

## References

1. Prats, M., Thermal Recovery, SPE Monograph, Vol. 7, SPE of AIME, Dallas, Texas, 1982.
2. Lauwerier, H.A., "The Transport of Heat in an Oil Layer Caused by the Injection of Hot Fluids," *Appl. Sci. Res.*, Martinus Nijhoff, Publisher, The Hague (1955), Vol. 5, Sec. A, No. 2-3, pp. 145-150.
3. Dietrich, J.K., "Relative Permeability During Cyclic Steam Stimulation of Heavy-Oil Reservoirs," *JPT*, October 1981, pp. 1987.
4. Bennion, D.W., and Moore, R.G., "Effect of Relative Permeability on the Numerical Simulation of the Steam Stimulation Process," *JCPT*, March-April 1985, pp. 40.
5. Kisman, K.E., and Acteson, W.H., "Application of Basal Water Sands to Enhance Thermal Recovery," paper 102, presented at the Fourth UNITAR/UNDP Conference on Heavy Crude and Tar Sands, Edmonton, Alberta, August 1988.
6. Maini, B.B., and Okazawa, T., "Effects of Temperature on Heavy Oil-Water Relative Permeability of Sand," *JCPT*, Vol. 26, No. 3, June 1987, pp. 33-41.
7. Marx, J.W., and Langeheim, R.N., "Reservoir Heating by Hot Fluid Injection," *Trans. AIME*, 216, 1959, pp. 312.
8. Hearn, C.L., "Effect of Latent Heat Content of Injected Steam in a Steam Drive," *JPT*, April 1969, pp. 374.
9. Aziz, K., and Settari, A., Petroleum Reservoir Simulation, Applied Science Publishers, 1979.
10. Butler, R.M., "A New Approach to the Modelling of Steam-Assisted Gravity Drainage," *JCPT*, Vol. 24, No. 3, May-June 1985, pp. 42-51.
11. Coats, K.H., "Simulation of Steamflooding with Distillation and Solution Gas," *SPEJ*, October 1976, pp. 235-247.
12. Dietrich, J.K., "Steam Cycling of Tar Sands Through Hydraulically Induced Fractures," *SPE Res. Eng.*, May 1986, pp. 217-229.
13. Denbina, E.S., Boberg, T.C., and Rotter, M.B., "Evaluation of Key Reservoir Drive Mechanisms in the Early Cycles of Steam Simulation at Cold Lake," paper SPE 16737, presented at the 62nd Annual Technical Conference, Dallas, Texas, September 1987.
14. Espinoza, C.E., "A New Formulation for Numerical Simulation of Compaction, Sensitivity Studies for Steam Injection," paper SPE 12246, presented at the Reservoir Simulation Symposium, San Francisco, California, November 1983.
15. Beattie, C.I., Boberg, T.C., and McNab, G.S., "Reservoir Simulation of Cyclic Steam Stimulation in the Cold Lake Oil Sands," paper SPE 18752, presented at the 1989 California Regional Meeting, Bakersfield, California, April 5-7, 1989.
16. Burger, J.G., and Sahuquet, B.C., "Chemical Aspects of In-Situ Combustion - Heat of Combustion and Kinetics," *SPEJ*, October 1972, pp. 410.

17. Millour, J.P., Moore, R.G., Bennion, D.W., Ursenbach, M.G., and Gie, D.N., "An Expanded Compositional Model for Low-Temperature Oxidation of Athabasca Bitumen," JCPT, Vol. 26, No. 3, June 1987, pp. 24-32.
18. Hallam, R.J., Hajdo, L.E., and Donnelly, J.K., "Thermal Recovery of Bitumen at Wolf Lake," SPERE, May 1989, pp. 178.
19. Brigham, W.E., and Abbaszadeh-Dehghani, M., "Tracer Testing for Reservoir Description," JPT, May 1987, pp. 519.
20. Arya, A., Hewett, T.A., Larson, R.G., and Lake, L.W., "Dispersion and Reservoir Heterogeneity," SPE Res. Eng., Feb. 1988, pp. 139.
21. Mishra, S., Brigham, W.E., and Orr, F.M., "Tracer and Pressure Test Analysis for Characterization of Areally Heterogeneous Reservoirs," paper SPE/DOE 17365, presented at the EOR Symposium, Tulsa, Oklahoma, April 1988.
22. Redford, D.A., "The Use of Solvents and Gases with Steam in the Recovery of Bitumen from Oil Sands," JCPT, January-February 1982, pp. 48.
23. Stone, T., and Ivory, J., "An Examination of Steam-CO<sub>2</sub> Processes," JCPT, May-June 1987, pp. 54.
24. Blunschi, J.H., "Simulation of a 12-Pattern Field Trial of Light Oil Steam Flooding," SPE paper 16735, presented at the 62nd Annual Technical Conference, Dallas, Texas, September 1987.
25. Stone, T., Boon, J., and Bird, G.W., "Modelling Silica Transport in Large Scale Laboratory Experiments," JCPT, Vol. 25, 1986, pp. 76.
26. Reed, M.G., "Gravel Pack and Formation Sandstone Dissolution During Steam Injection," JPT, June 1980, pp. 941.
27. Iler, R.K., "The Chemistry of Silica: Solubility, Polymerization, Colloid and Surface Properties and Biochemistry," John Wiley and Sons, New York, 1979.
28. Sydansk, R.D., "Stabilizing Clays with KOH," paper SPE 11721, presented at the 1983 California Regional Meeting, Ventura, California, March 1983.
29. Bryant, S.L., Schechter, R.S., and Lake, L.W., "Interactions of Precipitation/Dissolution Waves and Ion Exchange," AIChE Journal, Vol. 32, 1986, pp. 751.
30. Sorbie, K.S., Roberts, L.J., and Foulser, R.W.S., "Polymer Flooding Calculations for Highly Stratified Brent Sands in the North Sea," Proc. 2nd European Symposium on EOR, Paris, France, November 1982.
31. Bondor, P.L., Hirasaki, G.J., and Tham, M.J., "Mathematical Simulation of Polymer Flooding in Complex Reservoirs," SPEJ, October 1972, pp. 369.
32. Hirasaki, G.J., and Pope, G.A., "Analysis of Factors Influencing Mobility and Adsorption in the Flow of Polymer Solution Through Porous Media," SPEJ, August 1974, pp. 337.
33. Scott, T., Roberts, L.J., Sharpe, S.R., Clifford, P.J., and Sorbie, K.S., "In-situ Gel Calculations in Complex Reservoir Systems Using a New Chemical Flood Simulator," SPE Res. Eng., November 1987, pp. 634.

34. Hubbard, S., Roberts, L.J., and Sorbie, K.S., "Experimental and Theoretical Investigation of Time-Setting Polymer Gels in Porous Media," paper SPE/DOE 14959, presented at the 5th EOR Symposium, Tulsa, Oklahoma, April 1986.
35. Nagra, S.S., Batycky, J.P., Nieman, R.E., and Bodeux, J.B., "Stability of Waterflood Diverting Agents at Elevated Temperatures in Reservoir Brines," paper SPE 15548, presented at the 61st Annual Technical Conference, New Orleans, Louisiana, October 1986.
36. Healy, R.N., and Reed, R.L., "Physiochemical Aspect of Microemulsion Flooding," SPEJ, October 1974, pp. 941.
37. Novosad, J., "Surfactant Retention in Berea Sandstone-Effects of Phase Behavior and Temperature," SPEJ, December 1982, pp. 962.
38. Hirasaki, G.J., "Interpretation of the Change in Optimal Salinity with Overall Surfactant Concentration," SPEJ, December 1982, pp. 971.
39. Gupta, S.P., and Trushenski, "Micellar Flooding - Compositional Effects on Oil Displacement," SPEJ, April 1979, pp. 116.
40. Camilleri, D., Engelsen, S., Lake, L.W., Lin, E.C., Ohno, T., Pope, G., and Sepehrnoori, K., "Description of an Improved Compositional Micellar/Polymer Simulator," SPE Res. Eng., November 1987, pp. 427.
41. Camilleri, D., Fil, A., Pope, G.A., Rouse, B.A., and Sepehrnoori, K., "Comparison of an Improved Compositional Micellar/Polymer Simulator with Laboratory Core Floods," SPE Res. Eng., November 1987, pp. 441.
42. deZabala, E.F., Vislocky, J.M., Rubin, E., and Radke, C.L., "A Chemical Theory for Linear Alkaline Flooding," SPEJ, April 1982, pp. 245.
43. Burk, J.H., "Comparison of Sodium Carbonate, Sodium Hydroxide, and Sodium Orthosilicate for EOR," SPE Res. Eng., February 1987, pp. 9.
44. Bunge, A.L., and Radke, C.J., "Reversible Hydrogen Uptake on Reservoir Rock," SPEJ, October 1985, pp. 711.
45. Doll, T.E., "An Update of the Polymer-Augmented Alkaline Flood at the Isenhour Unit," Sublette County, Wyoming, SPE Res. Eng., May 1988, pp. 604.
46. Sydansk, R.D., "Elevated Temperature Caustic-Sandstone Interactions-Implications for Improving Oil Recovery," SPEJ, August 1982, pp. 453.
47. Mehdizadeh, A., and Handy, L.L., "Further Investigation of High Temperature Alkaline Floods," paper SPE 13072, presented at the 59th Annual Technical Conference, Houston, Texas, September 1984.
48. Janssen-van Rosmalen, R., and Messelink, F.T., "Hot Caustic Flooding," Proceedings of the First European Symposium on EOR, Bournemouth, England, September 1981, pp. 573.
49. Nasr, T., and McKay, A.S., "Thermal Low Temperature Bitumen Recovery Processes Using Hot Water Additives," Alberta Research Council Technical Report 8788-16, November 1987.

50. Herzig, J.P., Leclerc, D.M., and LeGoff, P., "Flow of Suspensions Through Porous Media - Applications to Deep Filtration," *Ind. Eng. Chem.*, Vol. 63, 1979, pp. 8.
51. Tienard, C., and Payatakes, A.C., "Advances in Deep Bed Filtration," *AIChE J.*, Vol. 25, 1979, pp. 737.
52. Gruesbeck, C., and Collins, R.E., "Entrainment and Deposition of Fines Particles in Porous Media," *SPEJ*, Vol. 22, 1982, pp. 847.
53. Khilar, K.C., and Folger, H.S., "Water Sensitivity of Sandstones," *SPEJ*, February 1983, pp. 55.
54. Schmidt, D.P., Soo, H., and Radke, C.J., "Linear Oil Displacement by the Emulsion Entrapment Process," *SPEJ*, June 1984, pp. 351; Soo, H., and Radke, C.J., "Flow of Dilute Stable Liquid and Solid Dispersions in Underground Porous Media," *AIChE J.*, Vol. 31, 1985, pp. 1926.
55. Becher, P., "Emulsions: Theory and Practice," 2nd Ed., Reinhold, New York, 1965.
56. Spielman, L.A., and Su, Y.P., "Coalescence of Oil-in-Water Suspensions by Flow Through Porous Media," *Ind. Eng. Chem. Fund.*, Vol. 16, 1977, pp. 272.
57. Roof, J.G., "Snapoff of Oil Droplets in Water Wet Pores," *SPEJ*, March 1970, pp. 85.
58. French, T.R., Bruz, J.S., Lorenz, P.B., and Bertus, K.M., "Use of Emulsions for Mobility Control During Steam Flooding," paper SPE 15052, presented at the 56th California Regional Meeting, Oakland, California, April 1986.
59. Chung, K.H., and Butler, R.M., "In-situ Emulsification by the Condensation of Steam in Contact with Bitumen," CIM paper 88-39-18, presented at the 39th Annual Technical Conference, Calgary, Alberta, June 1988.
60. Salager, J.L., Loaiza Maldonado, I., Minana-Perez, M., and Silva, F., "Surfactant-Oil-Water Systems Near the Affinity Inversion Part I: Relationship Between Equilibrium Phase Behavior and Emulsion Type and Stability," *J. Disp. Sc. Tech.*, Vol. 3, 1982, pp. 279.
61. Dilgren, R.E., Deemer, A.R., and Owens, K.B., "The Laboratory Development and Field Testing of Steam/Noncondensable Gas Foams for Mobility Control in Heavy Oil Recovery," paper SPE 10774, presented at the 1982 California Regional Meeting, San Francisco, California, March 1982.
62. Ploeg, J.F., and Duerksen, J.H., "Two Successful Steam/Foam Field Tests, Midway-Sunset Field," paper SPE 13609, presented at the 1985 California Regional Meeting, Bakersfield, California, March 1985.
63. Holm, L.W., and Garrison, W.H., :"CO<sub>2</sub> Diversion With Foam in an Immiscible CO<sub>2</sub> Field Project," *SPE Res. Eng.*, February 1988, pp. 112.
64. Friedmann, F., Chen, W.M., and Gauglitz, P.A., "Experimental and Simulation Study of High Temperature Foam Displacement in Porous Media," paper SPE/DOE 17357, presented at the Enhanced Oil Recovery Symposium, Tulsa, Oklahoma, April 1988.

65. Handy, L.L., Amaefule, J.O., Ziegler, V.M., and Ershagi, I., "Thermal Stability of Surfactants for Reservoir Application," SPEJ, October 1982, pp. 722.
66. Lau, H.C., and O'Brien, S.M., "Surfactant Transport Through Porous Media in Stream Foam Processes," paper SPE 14391, presented at the 60th Annual Technical Conference, Las Vegas, Nevada, September 1985.
67. Lau, H.C., and O'Brien, S.M., "Effects of Spreading and Nonspreading Oils on Foam Propagation Through Porous Media," SPE Res. Eng., August 1988, pp. 893.
68. Jensen, J.A., and Friedman, F., "Physical and Chemical Effects of an Oil Phase on the Propagation of Foam in a Porous Media," paper SPE 16375, presented at the California Regional Meeting, Ventura, California, April 1987.
69. Mohammadi, S.S., Van Slyke, D.C., and Ganong, B.L., "Steam-Foam Pilot Project in Dome-Tumbador Midway-Sunset Field," SPERE, February 1989, pp. 7.



# Appendix D: Fluid and Rock Properties

---

## Overview

This appendix contains detailed descriptions of the fluid component models as well as rock-fluid model used by STARS, divided into the following sections.

- D.1 Components and Phases
- D.2 Component Design Concepts
- D.3 Fluid Phase Equilibrium
- D.4 Fluid Densities
- D.5 Viscosity
- D.6 Rock Fluid Properties
- D.7 Component Adsorption and Blockage
- D.8 A Simple Foam Model
- D.9 Phase Enthalpies
- D.10 Thermal Conductivity
- D.11 Overburden Heat Loss
- D.12 Thermal Aquifer
- D.13 Chemical Reactions
- D.14 Basic Concepts for Nonequilibrium Mass Transfer
- D.15 Stable Emulsion Flow and In Situ Generation Concepts
- D.16 A Lamella Density Model of Foam
- D.17 Oil Banking Theory
- D.18 Converting Black-Oil PVT to STARS
- D.19 Other Aquifer Models
- D.20 Velocity-Dependent Viscosity

## D.1 Components and Phases

### Roles of Components and Phases

An important aspect of thermal EOR processes is the interaction between chemical components. A chemical component is a single substance that can be identified as chemically uniform for the current purpose or task at hand. A component may be

- a single molecular species, such as water or methane
- a mixture of a range of functionally similar molecular species, such as all hydrocarbons with molecular weight between 200 and 500, with a fixed distribution.

A phase is a physical manifestation of (and is composed of) one component or more components. For example, the chemical component water commonly may be found in the liquid, gaseous and solid phases. It is the phase that possesses tangible properties such as density and viscosity. Therefore, all physical properties are assigned to a chemical component in terms of the phases in which that component may be found. Some sources of information about properties of common reservoir components may be found in handbooks (Ref. 2 & 3) and texts (Ref. 1).

In modelling transient multiphase flow on an average (reservoir) scale, it can sometimes prove extremely useful to relax the above traditional distinction between components and phases. This is especially true when the phases contain components (such as surfactants) which tend to stabilize small dispersions (droplets, bubbles, and lamella) of one phase in another - for example in emulsions and foam. In these cases, a third meaning can be employed for the term "component" - a dispersion of small amounts of one phase carried along in another, continuous, phase. Usually this "component" will have the same equilibrium properties (molecular weight, density) as the true component of which it consists, but will differ substantially in its effects on flow (viscosity, resistance factor).

One of the first simulation design decisions to make concerns the number and type of components, and the phases in which each is found.

### Black-Oil System

A conventional black-oil simulation involves water, oil and a solution gas. The solution gas may dissolve into the oil. The oil component is the combination of all other hydrocarbons, which do not vaporize at the conditions of interest. This information can be organized into a component-phase chart, such as Figure D.1.

COMPONENT	PHASE		
	Aqueous	Oleic	Gaseous
Water	X		
Dead Oil		X	
Solution Gas		X	X

Figure D.1: Black-Oil system

## Thermal Black-Oil System

A steam flood or cycling process without additives might be adequately described with components shown in Figure D.2. This has the same components as the black-oil system, but in addition accounts for steam (gaseous water). This is known as a black-oil-steam or thermal black-oil model. Many "steam" models existing today<sup>4</sup> are of this type, and commonly were developed by adding an energy equation to an available isothermal black-oil model.

COMPONENT	PHASE		
	Aqueous	Oleic	Gaseous
Water	X		X
Dead Oil		X	
Solution Gas		X	X

Figure D.2: Thermal black-oil system

## Steam-Additive System

When differential vaporization of oil is part of the EOR process, the oil must be represented as more than one component, as shown in Figure D.3. Gaseous additives such as naphtha and water-soluble carbon dioxide also might be used, as shown in Figure D.3.

COMPONENT	PHASE		
	Aqueous	Oleic	Gaseous
Water	X		X
Heavy Oil		X	
Light Oil		X	X
Naphtha		X	X
Carbon Dioxide	X	X	X

Figure D.3: Steam-additive system

## Combustion System

In combustion, vaporization and chemical reactions often affect different parts of the oil phase. These processes may involve non-soluble air and reaction products and a solid coke fuel, as shown in Figure D.4.

COMPONENT	PHASE			
	Aqueous	Oleic	Gaseous	Solid
Water	X		X	
Asphaltenes		X		
Maltenes		X	X	
Light Oil		X	X	
CO <sub>2</sub>	X	X	X	
N <sub>2</sub> / CO			X	
Oxygen			X	
Coke Fuel				X

Figure D.4: Combustion system

## In Situ Gellation System

Two aqueous based additives (an adsorbing polymer and a non-adsorbing cross-linking agent) are injected into an oil containing reservoir to block preferential water pathways by reacting to form a pure blocking (re adsorbing) gel, as shown in Figure D.5.

COMPONENT	PHASE
Water	Aqueous X
Polymer	Oleic X
Cross-linker	Adsorbed X
Gel	X
Oil	X

Figure D.5: Gel system

## Emulsion Diverting System

A dilute, stable oil-in-water emulsion system is injected to plug a high permeability zone via droplet capture, to divert flow and to improve water displacement efficiency of a continuous oil phase.

COMPONENT	PHASE
Water	Aqueous X
Oil Droplet	Oleic X
Continuous Oil	X
Trapped Droplet	X

Figure D.6: Black-Oil system

## Simple Foam System

A simple phenomenological model of foam can be constructed focusing on the local concentration of surfactant by assuming that foam generation/coalescence mechanisms are fast relative to flow (quasi equilibrium assumption). In this approach the mobility of the gas phase (gas phase relative permeability) is reduced based primarily on surfactant concentration (velocity, oil saturation effects can also be considered).

COMPONENT	PHASE	Aqueous	Oleic	Gaseous	Adsorbed
Water	X				
Oil			X		
Surfactant	X		X		X
N <sub>2</sub>				X	

Figure D.7: Simple foam system

## Mechanistic Foam System

A more mechanistic model of foam propagation, creation and coalescence can be constructed employing a dispersed lamella component in the gas phase. Foam mobility reduction is then attributed to the concentration of lamella in the gas phase. Local lamella concentration is affected by explicit generation and coalescence rates which are in turn affected by local surfactant concentration.

COMPONENT	PHASE			
	Aqueous	Oleic	Gaseous	Adsorbed
Water	X			
Oil		X		
Surfactant	X	X		X
N <sub>2</sub>			X	
Lamella			X	X

*Figure D.8: Mechanistic foam system*

---

## D.2 Component Design Concepts

### Water Components

Water is a standard component, and internal defaults are available for vapor-liquid K-values, density, viscosity and heat capacity. The Stone's model for oil relative permeability assumes that the aqueous phase is the wetting phase, that is, the phase that contacts the rock at the pore level.

The STARS thermal model allows for other component besides water in the aqueous phase. Carbon dioxide solubility in water can be significant, and the aqueous phase can transport CO<sub>2</sub> to oil which otherwise might not be contacted. A labeled water tracer may be used to monitor formation versus fresh water, or salt or ion components. Other likely aqueous components are caustic, surfactants and polymers, as well as dispersed phase components like oil emulsion droplets and mobile fines. Often the presence of these components affects water phase propagation characteristics - viscosity, relative permeability, and water phase resistance factors.

### Oil Components

Oil components are of two types: A **pure component** consists of a single molecular species, such as a hydrocarbon C<sub>n</sub>H<sub>m</sub>, or a soluble hydrocarbon gas such as CH<sub>4</sub>.

A **pseudo-component** is a group of molecular species, lumped together and with a fixed distribution. An example is the dead oil component in a black-oil model, which usually includes all hydrocarbon species except CH<sub>4</sub>. A pseudo-component could consist of a limited group of molecular species, such as C<sub>7</sub> to C<sub>9</sub>. Native oil may be split into pseudo-components using the following techniques.

1. Molecular Weight. For example, one pseudo-component may be all hydrocarbons with molecular weight between 200 and 400, with a fixed average value.
2. Distillation Analysis. For examples, one pseudo-component could be the fraction of the oil which vaporizes between 15°C and 300°C. This fraction may be separated physically, and its properties either measured in the lab or estimated from a characteristic boiling temperature or molecular weight.
3. Solubility Analysis. A series of solvents is used to separate soluble parts of the oil from insoluble parts. The properties of the resulting fractions are measured in the lab or estimated using a characteristic boiling temperature or molecular weight.
4. Equation of State. An oil can be characterized accurately with many (10 to 30) components by matching PVT and VLE lab data with an equation of state package, such as CMGPROP6. The result of lumping these components into a few (2 to 5) pseudo-components can be tested with the same software. Properties estimated for these lumped components may be used directly.

An oil component may be volatile to some degree, in which case K-values and gas phase properties must be supplied, in addition to oil phase properties. A dead oil component will have zero K-value.

## Gas Components

Gas components are those that normally make up the gas phase at standard conditions, such as solution gas, CO<sub>2</sub> and air, and normally will have K-values greater than one.

A **condensable gas** is a gas component which is soluble in a liquid phase. Properties in both the liquid and gas phases must be supplied, along with K-values. An example is the solution gas component in black-oil models. Liquid phase properties may be only apparent, that is, cannot be measured directly because that component does not form a liquid phase by itself at the conditions of interest. In this case, indirect estimates are made.

A **non-condensable gas** is one for which the solubility in liquid is small enough to be ignored. Such a component exists only in the gas phase, theoretically corresponding to an infinitely large K-value. Instead, these components are treated differently from the condensable components: K-values and liquid property data are not required, and the corresponding properties and equation terms are skipped.

Typical examples are oxygen and nitrogen (air), and carbon monoxide, used in combustion. Carbon dioxide and methane may be condensable or non-condensable, depending on which processes are of interest. Often the non-condensable gases nitrogen and carbon oxides are lumped into one functional component, called “inert gas”.

## Adsorbed or Trapped Components

An adsorbed component is a non-mobile component which is in equilibrium with its fluid phase counterparts. This equilibrium relationship is expressed as a (temperature dependent) adsorption isotherm describing amount adsorbed as a function of a specified fluid phase composition. In essence, the adsorption isotherm acts as a fluid-solid phase K-value.

Typical adsorbed species include polymers and surfactants as well as inorganic cations (sodium, calcium, and hydrogen) which can ion exchange with the rock. Since the actual capture mechanism is not specified via phenomenological adsorption isotherms, the list of adsorbed species can also be extended to capture “dispersed component” particles such as fines, emulsions, and foam lamellae, when the capture process is fast relative to the flow (i.e. the equilibrium assumption holds).

## Solid or Trapped Components

Traditionally, a solid component is usually the solid coke fuel deposited on the rock when the heavy parts of the oil are cracked at higher temperature. It is characterized by its H to C ratio, and is commonly assumed to be HC. In addition, some simple types of formation dissolution can be modelled by calling V<sub>r</sub> the volume of insoluble rock and V<sub>s</sub> the volume of soluble rock, such as when carbonate is dissolved in the presence of CO<sub>2</sub>. In each case, the porosity change is accounted for.

Again, because the rate expressions for creating or destroying solid components are general phenomenological interphase mass transfer equations, the list of solid species can be extended to include “dispersed component” particles such as fines, emulsions, and foam lamellae. However, in this instance, the rate of capture (and/or release) is of importance (i.e. a nonequilibrium process).

## D.3 Fluid Phase Equilibrium

### Definitions of K Values

In compositional models, the phase equilibrium is specified via phase equilibrium ratios (K-values). These may be obtained from some thermodynamic model e.g. Peng-Robinson equation of state or input directly as functions of pressure, temperature and composition.

STARS requires the latter approach. For three phases (aqueous, oleic, and gaseous), three K-values can be defined for each component i,

$$K_i^{gw} = \frac{y_i}{w_i}, K_i^{go} = \frac{y_i}{x_i}, K_i^{ow} = \frac{x_i}{w_i} \quad (D3.1)$$

but only two of these are independent (the third is a ratio of the other two). For ease of input and to reduce the number of nonzero K-values required, STARS groups components as water-like, oil-like, and non-condensable gases depending on the choice of reference phase - i.e. which K-value is viewed as dependent and is not required. For the water-like components, water is the reference phase and

$$K_i^{gl} \equiv \frac{y_i}{w_i}, K_i^{ll} \equiv \frac{x_i}{w_i} \quad 1, \dots, \text{NUMW} \quad (D3.2)$$

are input. For the oil-like components, oil is the reference phase and

$$K_i^{gl} \equiv \frac{y_i}{x_i}, K_i^{ll} \equiv \frac{w_i}{x_i} \quad \text{NUMW}+1, \dots, \text{NUMX} \quad (D3.3)$$

are input. for the non-condensable gases, gas is the reference phase, and in addition the K-values

$$K_i^{wg} \equiv \frac{w_i}{y_i} \equiv 0, K_i^{og} \equiv \frac{x_i}{y_i} \equiv 0 \quad \text{NUMX}+1, \dots, \text{NUMY} \quad (D3.4)$$

are assumed zero so no K-values input is required. The simpler partitioning assumptions of a thermal black-oil model follow directly by setting NUMW=1 (one water-like component) and setting all liquid-liquid K-values equal to zero (no partitioning of water in oil and vice versa).

### Vapor Pressure Perspective

A black-oil model approaches phase equilibrium from a solubility angle, where the physical concepts are expressed in terms of bubble point pressure and gas-oil ratios. However, the wide range of processes involved in thermal EOR requires that a more general compositional approach be taken here, starting with a vapor pressure perspective of phase equilibrium.

Consider pure component i in its liquid state at temperature T. When this liquid is decompressed (pressure decreased) at constant temperature the vapor phase will appear at a certain pressure. In fact, the liquid and vapor phases will coexist at only one pressure, called the vapor pressure at that temperature. Similar measurements at other T will result in a vapor pressure curve, written as  $p = psat(T)$ . A steam table is an example of a vapor pressure curve. A pure-component vapor pressure is a function only of T.

Consider a two-phase mixture of one or more components in phase equilibrium, with mole fractions  $x_i$  and  $y_i$ . The following two key concepts are required:

1. Vapor pressures  $p_{vi}$  and pressure  $p$  are low enough that all fugacity coefficient corrections are negligible.
2. The liquid is assumed to be an ideal mixture, whereby pure-component properties can be used, that is, mixed proportional to  $x_i$ . This means that  $p_{vi}(T)$  is independent of  $x_i$ .

The combination of these concepts results in **Raoult's Law**:

The partial pressure exerted by each component in an ideal solution is the mole-weighted pure-component value (Ref. 7, p. 404).

If we denote the partial pressure in the gas phase as  $y_{ip}$ , then Raoult's Law can be written as

$$y_{ip} = x_i p_{vi}(T) \quad (\text{D3.5})$$

So the K-value may be thought of as vapor pressure divided by pressure

$$K_i = y_i / x_i = p_{vi}(T) / p \quad (\text{D3.6})$$

Knowing this, it is easy to calculate K-value data from vapor pressure data. Also note that  $K_i$  is naturally a function of  $p$  and  $T$ .

### Derivation of Simple Correlation

The simple K-value correlation mentioned previously can be derived from thermodynamic theory. Consider the Clausius - Clapeyron equation

$$\left. \frac{\partial p_v}{\partial T} \right|_{\text{sat}} = \frac{\Delta H}{T \Delta V} \quad (\text{D3.7})$$

where  $V$  is the molar volume and  $\Delta$  denotes the difference between the liquid and vapor values (Ref. 7, pp. 250-251).

If we assume that

1.  $p_v$  is low enough that the liquid volume is negligible compared to the gas volume, so that  $\Delta V = V_g$
2. the vapor phase is ideal, so that  $V_g = RT / p$ , and
3. the heat of vaporization  $\Delta H$  is constant,

then we can integrate

$$\frac{dp_v}{dT} = \frac{p_v \Delta H}{RT^2} \quad (\text{D3.8})$$

from temperature  $T_1$  to  $T$  to get

$$p_v(T) = p_v(T_1) \exp \left[ -\frac{\Delta H}{R} \left( \frac{1}{T} - \frac{1}{T_1} \right) \right] \quad (\text{D3.9})$$

If we let

$$a = p_v(T_1) \exp\left[+ b/T_1\right] \quad (D3.10)$$

$$b = \Delta H/R \quad (D3.11)$$

then

$$p_v(T) = a \exp(-b/T) \quad (D3.12)$$

Remembering that the K-value is  $p_v(T)/p$ , the correlation for K-value is

$$K(p, T) = (a/p) \cdot \exp(-b/T) \quad (D3.13)$$

This derivation gives us a thermodynamic basis for estimating the coefficients a and b.

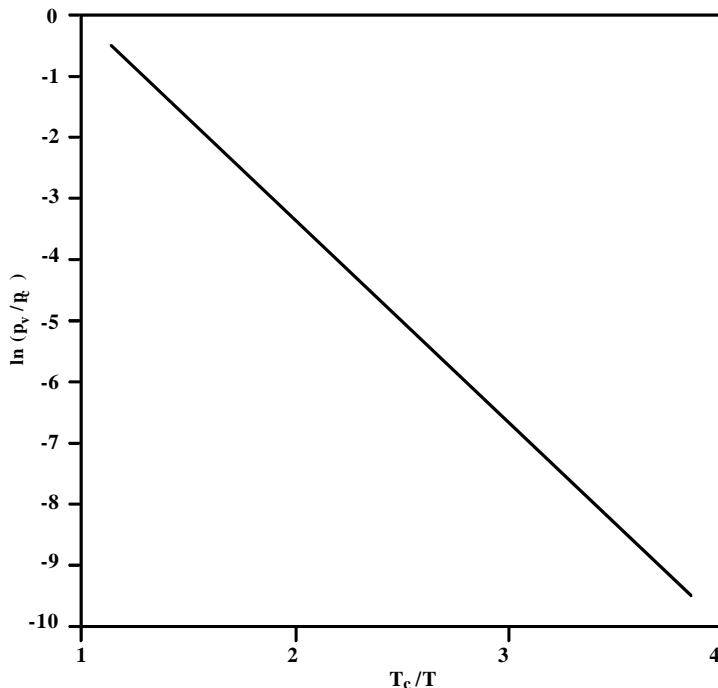


Figure D.9: Vapor pressure curve from Van der Waal's EOS

This correlation works surprisingly well over a wide range of p and T, even up to the component's critical point. To test the accuracy of the correlation, Van der Waal's equation of state was used to calculate vapor pressures which were plotted as  $\ln(p_v/p_c)$  versus  $T_c/T$ , in Figure D.9. The curve of the EOS result is nearly straight, and its slope changes significantly only near the critical point (Table D.1). The correlation's results lie exactly on a straight line. For example, the straight line fitted through the critical point and the last point in Table D.1 is described as follows:

Average slope is:

$$\frac{(0) - (-9.6161)}{(1) - (3.800)} = -3.434 \quad (D3.14)$$

Correlation is

$$\ln(p_v/p_c) = -3.434 (T_c/T - 1) \quad (D3.15)$$

or

$$p_v(T) = 31.00 p_c \cdot \exp(-3.434 T_c/T) \quad (D3.16)$$

$1/T_r$	$T_r$	$\ln(p_r)$	$p_r$	slope
1.100	0.9091	-0.3926	0.6753	-3.858
1.500	0.6667	-1.8608	0.1556	-3.532
2.000	0.5000	-3.5831	$2.7789 \cdot 10^{-2}$	-3.386
3.000	0.3333	-6.9360	$9.7217 \cdot 10^{-4}$	-3.343
3.800	0.2632	-9.6161	$6.6649 \cdot 10^{-5}$	-3.351

Slope is  $\frac{d\ln(p_r)}{d(1/T_r)} = -\frac{T_r \Delta h_r}{Z_c p_r \Delta V_r}$

where

- $p_r$  - reduced pressure  $p_v/p_c$
- $T_r$  - reduced temperature  $T/T_c$
- $h_r$  - reduced enthalpy  $(H - H^*)/RT_c$
- $V_r$  - reduced molar volume  $V/V_c$
- $Z_c$  - critical compressibility  $p_c V_c/RT_c (=0.375 \text{ for the Van der Waal EOS})$
- $\Delta$  - denotes difference between vapor and liquid phases
- \* - denotes ideal gas ( $p=0$ ) value

Table D.1: Vapor pressure for Van der Waal's EOS

### Hand's Tie-Line Interpolation Option

As described by van Quy et al<sup>8</sup> and Young and Stephensen<sup>9</sup>. Hand's rule is a tie-line relationship that assumes, for a ternary system, that all tie-lines intersect at a given point. As illustrated in Figure D.10 below, this allows a normalized composition variable  $u$  to be defined

$$u = \frac{z_3}{Az_2 + B} \quad (D3.17)$$

such that  $Au$  and  $Bu$  are the slope and intercept of the tie-line corresponding to the given global composition  $(z_1, z_2, Z_3)$ , since

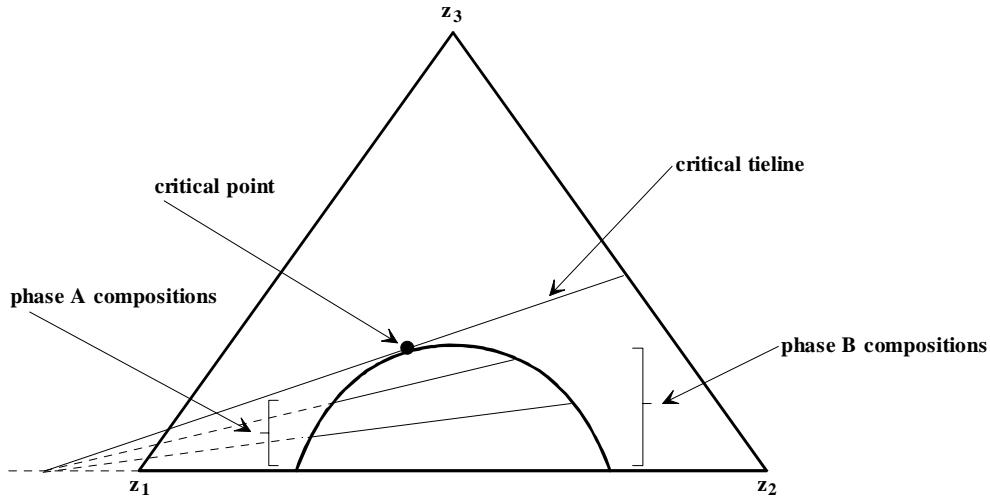
$$z_3 = (Au) z_2 + (Bu) \quad (D3.18)$$

With A, B defined as the slope and intercept of the critical tie line, the composition variable u ranges between [0, 1].

Defining the K-values  $K_i^{BA}$

$$K_i^{BA} = \frac{\text{composition of component } i \text{ in phase B}}{\text{composition of component } i \text{ in phase A}} \quad (\text{D3.19})$$

valid input for this system would be  $K_1^{BA}(u)$ ,  $K_2^{BA}(u)$ ,  $K_3^{BA}(u)$  for a set of u values [0,1].



**Figure D.10:** Hand's tie-line rule in a ternary system

This approach is useful for gas-oil systems, allowing multiple-contact miscible descriptions of vaporizing-gas and condensing-gas drive processes<sup>10</sup>, as well as describing water-oil mixing for surfactant injection processes<sup>11</sup>. In this latter case, it is often sufficient to assume surfactant partitions out totally in the water phase (lower phase microemulsion) or totally in the oil phase (upper phase microemulsion) depending on e.g. brine salinity. Here the above scheme reduces to B ≡ 0, and only one component has  $K_i(u)$  nonzero when the K-values are suitably defined. Thus for an upper phase microemulsion

$$\begin{aligned} K_i^{OW} = K_i^{OW}(u) &= 1 = \text{water component can partition in oil} \\ K_2^{WO} = 0 &= 1 = \text{oil component remains in oil} \\ K_3^{WO} = 0 &= 3 = \text{surfactant component remains in oil} \end{aligned} \quad (\text{D3.20})$$

## D.4 Fluid Densities

Water and Oil Phases

Liquid densities are obtained by ideal mixing of pure-component densities with phase composition

$$\frac{1}{\rho_w} = \sum_{i=1}^{n_c} \frac{w_i}{\rho_{wi}} \quad (D4.1)$$

$$\frac{1}{\rho_o} = \sum_{i=1}^{n_c} \frac{x_i}{\rho_{oi}} \quad (D4.2)$$

Densities  $\rho_w$  and  $\rho_o$  are inverses of phase molar volumes. Component densities  $\rho_{wi}$  and  $\rho_{oi}$  are inverses of the corresponding partial molar volume<sup>7</sup>, and should be regarded as the pure-component contribution to the phase volume.

For liquid oil components it is possible to measure densities directly in the lab, and to use estimations from the literature. But it is unlikely that a soluble gas component such as methane will exist alone in a liquid phase, at least at the p and T of interest. The following example illustrates the meaning of  $\rho_{oi}$  in this case, and how it is obtained.

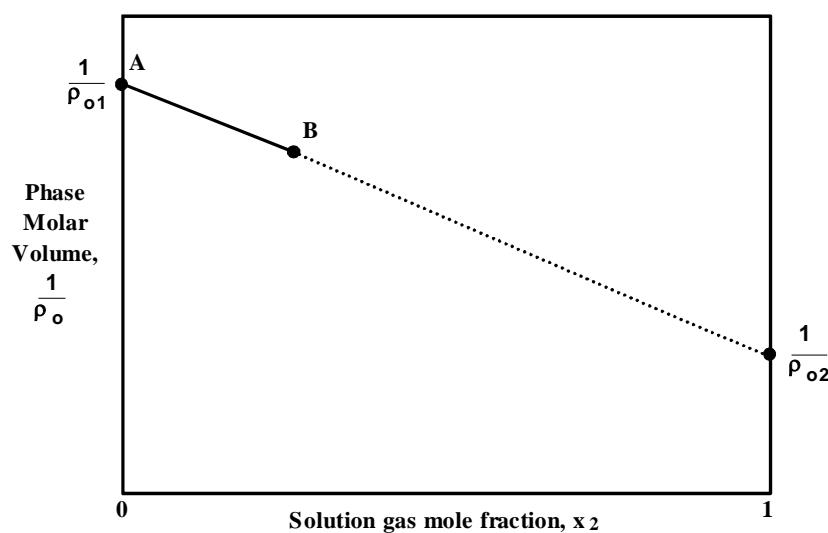
Example: Suppose 1m<sup>3</sup> of live oil contains 4.415 kg mole of dead oil and 0.5 kg mole of solution gas. Suppose also that 1m<sup>3</sup> of dead oil contains 4.5 kg mole.

The mixed phase density of the live oil is  $\rho_o = (4.415+0.5)/1m^3 = 4.915 \text{ kg mole/m}^3$ . The density  $\rho_{oi}$  is 4.500 kg mole/m<sup>3</sup>. The mole fraction of dead oil is  $x_1 = 4.415/4.915 = 0.8983$ , and  $x_2 = 1-x_1$ . The formula for  $\rho_{o2}$  is

$$\frac{1}{4.915} = \frac{0.8983}{4.500} + \frac{0.1017}{\rho_{o2}} \quad (D4.3)$$

$$\rho_{o2} = 26.47 \text{ kg mole m}^3.$$

For a solution gas with molecular weight of 18, this corresponds to a hypothetical liquid with a gravity of 0.48.



**Figure D.11: Meaning of liquid density of soluble gas**

One would not measure this density directly in the lab. Instead, a graph such as Figure D.11 is used. Molar volumes of samples with different gas content (points A and B) are plotted against mole fraction, and the slope is extrapolated to  $x_2 = 1$  where

$$\rho_0 = \rho_{02}.$$

The pure component densities  $\rho_{wi}$  and  $\rho_{oi}$  are simple correlation functions of T (**in absolute units**, see “Second Temperature Coefficient” in the EXPLANATION for keyword \*MOLDEN) and p:

$$\rho_{wi}(p, T) = \rho_{wi}^0 \cdot \exp \left[ a(p - p_r) - b(T - T_r) - \frac{1}{2} c(T^2 - T_r^2) \right] \quad (\text{D4.4})$$

where

a is the compressibility,

b+cT is the thermal expansion coefficient as a function of temperature, and

$\rho_{wi}^0$  is the density at reference conditions  $p_r$  and  $T_r$ .

A similar formula exists for  $\rho_{oi}(p, T)$ .

### Gas Phase

The mole density of the gas phase is calculated internally from

$$\rho_g = p / RTZ \quad (\text{D4.5}).$$

There are two ways to obtain Z.

The ideal gas method assumes that  $Z=1$ , so  $\rho_g$  depends only on p and T (**in absolute units**).

This method is cheap, but is inaccurate if any of the components are not far away from their critical point.

The second method uses the Redlich-Kwong EOS<sup>12</sup>, and typically predicts Z varying from 0.3 to 1.2.

1. Calculate critical p and T (in absolute units) from gas phase composition and Redlich-Kwong mixing rules, assuming zero interaction coefficients.

$$a = \sum_{i=1}^{n_c} y_i T_{ci} \sqrt{T_{ci}^{1/2}/p_{ci}} \quad (D4.6)$$

$$b = \sum_{i=1}^{n_c} y_i T_{ci} / p_{ci} \quad (D4.7)$$

$$T_c = \left( \frac{a^2}{b} \right)^{2/3} \quad (D4.8a)$$

$$p_c = T_c / b \quad (D4.8b)$$

2. Calculate Z factor from Redlich-Kwong EOS, assuming zero interaction coefficients. Z is the largest root of

$$Z^3 - Z^2 + (A - B^2 - B)Z - AB = 0 \quad (D4.9)$$

where

$$A = 0.427480 \bullet (p / p_c) \bullet (T_c / T)^{2.5} \quad (D4.10)$$

$$B = 0.086640 \bullet (p / p_c) \bullet (T_c / T) \quad (D4.11)$$

## D.5 Viscosity

### Water Phase

Water phase viscosity tends to be relatively constant at 1 cp, and decreases as far as 0.1 cp at 300°C. The main purpose of allowing entry of water viscosity data is to account for the various brine concentrations encountered in different reservoirs.

The STARS model has the following data entry options:

1. Use internal table of  $\mu_w$  versus T, with a possible dependence on salt concentration which can be significant.
2. Use the correlation  $\mu_w = a \cdot \exp(b/T)$ , where T is in absolute degrees.
3. Enter directly a table of  $\mu_w$  versus T.

### Gas Phase

Gas phase viscosities usually are much smaller than liquid phase values, and hence will tend to dominate the flow if gas phase is mobile. As a consequence, pressure gradients may be high when gas is immobile, but will certainly be low when gas is flowing. Gas phase viscosities have values around 0.01 cp.

The STARS model has the following gas viscosity options:

1. Correlation

$$\mu_g = 0.0136 + 3.8 \cdot 10^{-5} \cdot T \quad (D5.1)$$

T in deg C;  $\mu_g$  depends only on temperature, and not on composition or p. This gives  $\mu_g = 0.014$  cp at 20°C and  $\mu_g = 0.025$  at 300°C.

This option is very cheap to use and is very often quite sufficient, for the reason that the compositional effects on  $\mu_g$  are small.

2. Correlation

$$\mu_{gi} = a_i \cdot T^{bi} \quad (D5.2)$$

(T in absolute degrees) for each component. The phase viscosity is

$$\mu_g = \frac{\sum_{i=1}^{n_c} \omega_i \mu_{gi}}{\sum_{i=1}^{n_c} \omega_i} \quad (D5.3)$$

where

$$\omega_i = y_i \sqrt{M_i} \quad (D5.4)$$

because viscosities  $\mu_{gi}$  for different components are quite similar, and the T dependence is not strong, the first option mentioned above is often sufficient.

3. Same as option (2), but with an additional correction to  $\mu_g$  for high pressure.

## Oil Phase

In many steam-injection processes the  $\mu_o$ -versus-T function will contain much of the nonlinearity in the flow equations.

This is because  $\mu_o$  can decrease by several orders of magnitude over only a modest temperature increase. Therefore, it is crucial that the temperature dependence of  $\mu_o$  be represented adequately. In addition, the effect of soluble gas components on  $\mu_o$  can be significant, and so must be accounted for.

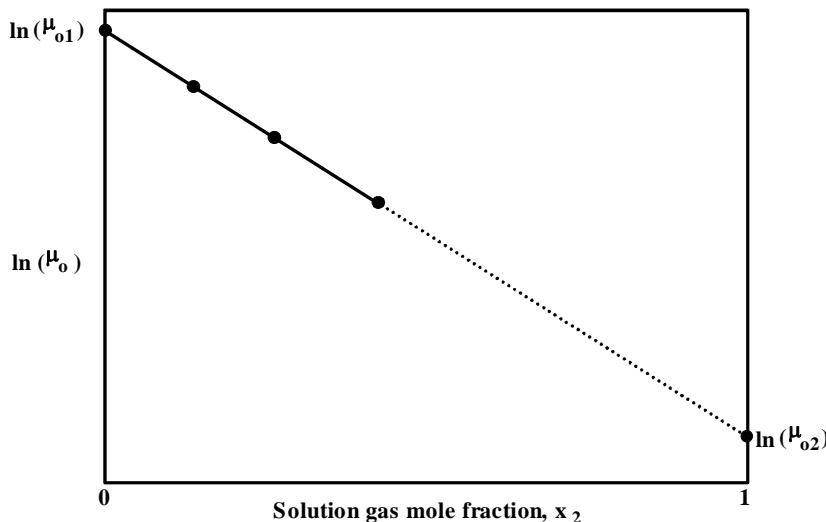
The oil phase viscosity is obtained by a logarithmic mixing rule:

$$\ln(\mu_o) = \sum_{i=1}^{n_c} x_i \ln(\mu_{oi}) \quad (D5.5)$$

For liquids the component value  $\mu_{oi}$  can be measured directly or estimated from correlations or tables. However, for a soluble gas such as methane, a measured value for the liquid phase may be difficult to find.

In this case, as was the case for oil phase density calculations,  $\mu_{oi}$  must be regarded as the contribution of the soluble gas toward the viscosity of the liquid mixture.

Figure D.12 shows that plotting  $\ln(\mu_o)$  versus mole fraction for various values of gas content will yield a value for the gas component  $\mu_{o2}$  by extrapolating  $x_2=1$ .



*Figure D.12: Calculation of liquid viscosity for solution gas component*

Example: Suppose a dead oil component has a viscosity of  $\mu_{o1}=1000$  cp. When some soluble gas is added and mixed thoroughly, the live oil viscosity is found to be  $\mu_o=300$  cp, and the mole fraction is calculated to be  $x_2=0.20$ . The equation for  $\mu_{o2}$  is

$$\ln(300) = 0.80 \ln(1000) + 0.20 \ln(\mu_{o2}) \quad (D5.6)$$

so

$$\ln(\mu_{o2}) = 0.1776 \text{ and } \mu_{o2} = 1.19 \text{ cp} \quad (\text{D5.7})$$

In the case of a soluble gas,  $\mu_{o2}$  is the viscosity of a hypothetical liquid composed of 100% solution gas. Note that  $\mu_{o2}$  is NOT the viscosity of solution gas in the gas phase. Note also that the above sample calculation must be done at other temperatures, in order to obtain the dependence of  $\mu_{o2}$  on T.

The component values  $\mu_{oi}$  may be specified using one of the following two options.

1. Correlation  $\mu_{oi} = a_i \cdot \exp(b_i/T)$ , where T is in absolute degrees.

Coefficients can be calculated from two points on the curve. For example, a heavy oil component may have

$$\mu_{oi} = 10^4 \text{ cp at } 20^\circ\text{C} (= 293 \text{ K}) \quad (\text{D5.8})$$

$$\mu_{oi} = 10 \text{ cp at } 300^\circ\text{C} (= 573 \text{ K}) \quad (\text{D5.9})$$

Just solve

$$10^4 = a_i \cdot \exp(b_i / 293) \quad (\text{D5.10})$$

$$10 = a_i \cdot \exp(b_i / 573) \quad (\text{D5.11})$$

for  $a_i$  and  $b_i$ , which are

$$b_i = \frac{\ln(10^4) - \ln(10)}{1/293 - 1/573} = 4141.9 \text{ K} \quad (\text{D5.12})$$

$$a_i = 10^4 / \exp(b_i / 293) = 7.256 \cdot 10^{-3} \text{ cp} \quad (\text{D5.13})$$

Check at the other point:

$$\text{At } T = 300^\circ\text{C}, \mu_{oi} = a_i \cdot \exp(b_i/573) = 1.0 \text{ cp.}$$

2. Enter directly a table of  $\mu_{oi}$  versus T. Interpolation between entries is done using the correlation described in option (1) above.

### Nonlinear Viscosity Mixing

For situations where the normal mixing rule for phase viscosities does not appear adequate, an approach based on a nonlinear mixing rule for a key component can be employed. Basically, the linear logarithmic mixing rule

$$\ln \mu = \sum_i x_i \ln \mu_i \quad (\text{D5.14})$$

is replaced by a nonlinear function

$$x_a \rightarrow f(x_a) \quad (\text{D5.15})$$

for one key component "a". The requirement that the pseudo-compositions still sum to 1 yields a condition on the normalizing factor N:

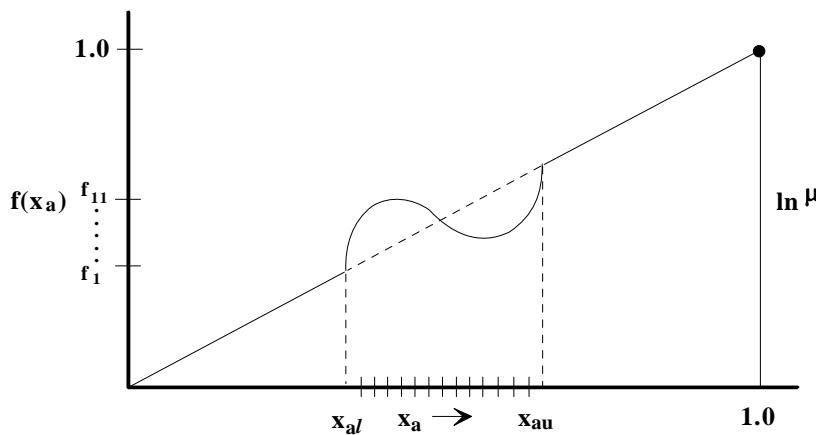
$$f(x_a) + N \sum_{i \neq a} x_i = 1 \quad (\text{D5.16})$$

$$\Rightarrow N = \frac{1 - f(x_a)}{1 - x_a} \quad (\text{D5.17})$$

so that the modified logarithmic mixing rule becomes

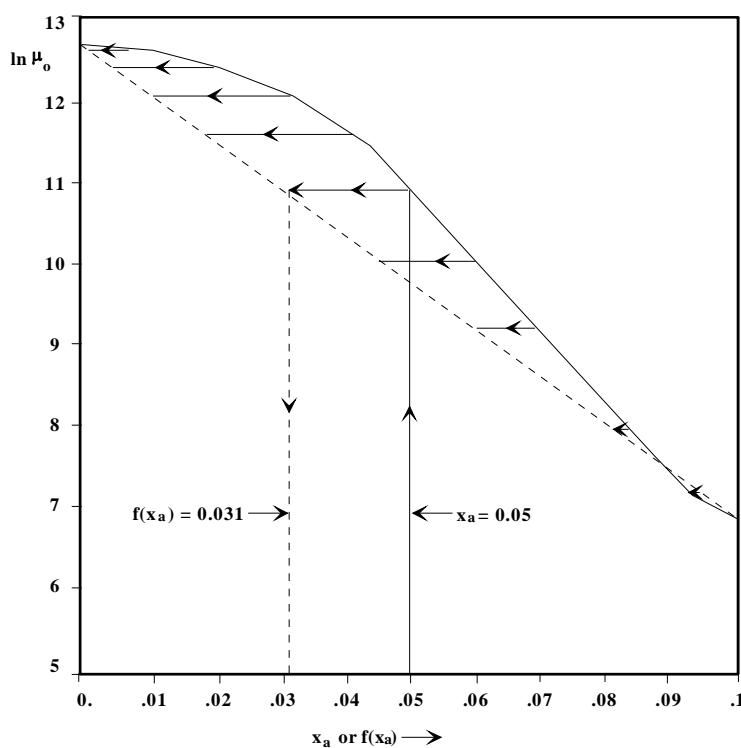
$$\ln \mu = f(x_a) \ln \mu_a + \frac{1 - f(x_a)}{1 - x_a} \sum_{i \neq a} x_i \ln \mu_i \quad (\text{D5.18})$$

If  $f(x_a)$  is linear then this form reduces to the original mixing rule. The function  $f(x_a)$  can in principle be quite general such that the nonlinear interval can be in any region between  $x_a = 0.0$  and  $x_a = 1.0$ .



**Figure D.13: Nonlinear mixing function for key component "a" illustrating the range of values (between  $x_{al}$  and  $x_{au}$ ) over which the nonlinear function is assumed to apply and the method by which the function is input into the simulator**

Figure D.13 shows the general form of this function and illustrates that outside the specified range, a linear function is assumed. This figure also illustrates the manner in which the nonlinear mixing function is entered - namely that the values of this function over 10 equally spaced intervals between  $x_{al}$  and  $x_{au}$ .



**Figure D.14:** Fit of nonlinear curve  $\mu_0 = \mu_b e^{-Bx_a^2} + \mu_a$  to the nonlinear viscosity mixing function illustrating how the 6th value ( $f(x_a = 0.05)$ ) can be obtained graphically (Parameter values  $\mu_b = 3 \times 10^5$  cp;  $\mu_a = 6 \times 10^2$  cp;  $B = 500$ ;  $x_a^{max} = 0.1$ )

In order to obtain this function, the recommended method is to plot values of the logarithm of the phase viscosity versus concentration of the key component, as illustrated in Figure D.14. A straight line joining the  $\ln \mu$  values at the two extremes of the region is also plotted. The first curve is  $\ln \mu$  as a nonlinear function of  $x_a$  while the second is the equivalent curve of  $\ln \mu$  as a linear function of  $f(x_a)$ . This equivalence is established for each of the 11 equally spaced values of  $x_a^i$  by connecting the appropriate  $\ln \mu(x_a^i)$  on the first curve to the same value on the straight line curve – the concentration value for this point is then  $f(x_a^i)$ , as illustrated explicitly in Figure D.14 for the 6<sup>th</sup> value,  $x_a^6$ . Table D.2 gives the complete results for the first example problem, illustrating viscosity reduction of bitumen by the addition of solvent.

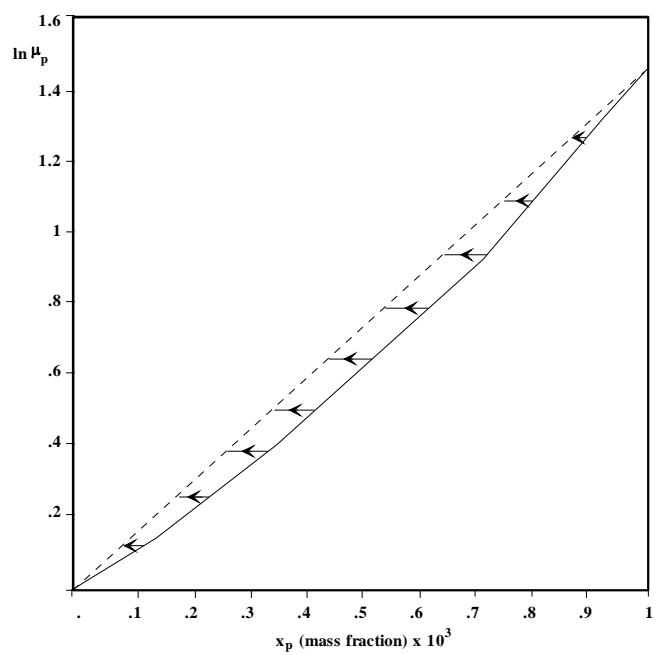
$x_a$	$\mu_o \ln \mu_o$	$f(x_a)$	
0.00	$3.006 \times 10^{-5}$	12.61	0.00
0.01	$2.790 \times 10^{-5}$	12.54	0.001
0.02	$2.229 \times 10^{-5}$	12.31	0.005
0.03	$1.533 \times 10^{-5}$	11.94	0.011
0.04	$9.09 \times 10^{-4}$	11.42	0.020
0.05	$4.65 \times 10^{-4}$	10.25	0.031
0.06	$2.07 \times 10^{-4}$	9.94	0.045
0.07	$8.19 \times 10^{-3}$	9.01	0.060
0.08	$3.07 \times 10^{-3}$	8.03	0.077
0.09	$1.29 \times 10^{-3}$	7.16	0.092
0.10	$7.66 \times 10^{-2}$	6.64	0.100

**Table D.2: Fit of nonlinear curve  $\mu_o = \mu_b e^{-Bx_a^2} + \mu_a$  to the nonlinear viscosity mixing function. The appropriate  $f(x_a)$  functions are obtained graphically employing the construction illustrated in Figure D.15**  
 (Parameter values :  $\mu_b = 3 \times 10^5$  cp;  $\mu_a = 6 \times 10^2$  cp;  $B = 500$ ;  $x_a^{\max} = 0.1$  )

A second example generalizes somewhat the above approach by normalizing  $f(x_a^{\max}) = 1.0$  instead of employing  $f(x_a^{\max}) = x_a^{\max}$ . Concurrently, the input value for the component viscosity  $\mu_a$  is set as  $\mu(x_a = x_a^{\max})$  instead of  $\mu(x_a = 1)$ . Such an approach is useful for situations when the expected range of compositions  $x_a \ll 1$  and pure component viscosities  $\mu(x_a = 1)$  are unknown or become meaningless for the problem to be simulated. A typical example of this situation is polymer viscosity modifications to the aqueous phase, see Figure D.15 and Table D.3.

$c_p$	$p$	$\ln p$	$x_p$ (mass frac)	$f(x_p)$	$f_N(x_p)$
0.0	1.000	0.0	0.0	0.0	0.0
0.01	1.111	0.105	$10^{-4}$	$7.5 \times 10^{-5}$	0.075
0.02	1.248	0.222	$2 \times 10^{-4}$	$1.6 \times 10^{-4}$	0.16
0.03	1.417	0.349	$3 \times 10^{-4}$	$2.5 \times 10^{-4}$	0.25
0.04	1.624	0.485	$4 \times 10^{-4}$	$3.5 \times 10^{-4}$	0.35
0.05	1.875	0.629	$5 \times 10^{-4}$	$4.5 \times 10^{-4}$	0.45
0.06	2.176	0.777	$6 \times 10^{-4}$	$5.6 \times 10^{-4}$	0.56
0.07	2.533	0.929	$7 \times 10^{-4}$	$6.7 \times 10^{-4}$	0.67
0.08	2.952	1.082	$8 \times 10^{-4}$	$7.8 \times 10^{-4}$	0.78
0.09	3.439	1.235	$9 \times 10^{-4}$	$8.9 \times 10^{-4}$	0.89
0.10	4.000	1.386	$1 \times 10^{-3}$	$1.0 \times 10^{-3}$	1.00

**Table D.3: Fit of nonlinear curve  $\mu_r = \mu_w (1 + A c_p + A_2 c_p^2 + A_3 c_p^3)$  to the nonlinear viscosity mixing function. The appropriate  $f(x_p)$  functions are obtained graphically (see Figure D.12) and then are normalized by dividing by  $f(x_p^{\max})$ .** (Parameter values :  $\mu_w = 1$  cp;  $A_1 = 10$ ;  $A_2 = 10^2$ ;  $A_3 = 10^3$ ). Concentration units in equivalent forms:  
 $c_p = 0.1$  wt% is equivalently  $10^3$  ppm;  $10^{-3}$  mass fraction;  $10^{-6}$  mole fraction if  $M_w = 18$  kg/gmole



**Figure D.15:** Fit of nonlinear curve  $\mu_p = \mu_w (1 + A c_p + A_2 c_p^2 + A_3 c_p^3)$  to the nonlinear viscosity mixing function.  
 (Parameter values :  $\mu_w = 1\text{cp}$ ;  $A_1 = 10$ ;  $A_2 = 10^2$ ;  $A_3 = 10^3$ ) and concentration values in weight percent converted to mass fractions)

## D.6 Rock Fluid Properties

### Relative Permeability and Capillary Pressure Interpolation Options

Under special circumstances (nearly miscible fluids, pH changes, surfactant concentration changes, large increases in applied flow velocities), the assumption that rock fluid properties are only functions of fluid saturations and fluid saturation history is not sufficient to accurately describe observed flow behavior.

In these cases, the ability to interpolate basic relative permeability and capillary pressure data as functions of concentration or capillary number can prove very useful. Because of the flexibility in the choice of interpolation parameter and the fact that essentially arbitrary tabular data for relative permeability and capillary pressure can be employed, a wide variety of phenomena can be handled - caustic flooding and wettability alteration, foam mobility reduction effects. Currently three interpolation options are employed:

1. Interpolation as a function of key component composition in a specified phase.
2. Capillary number interpolation requiring specification of interfacial tension as a function of a key component composition.
3. A simple foam interpolation (see Section D.8) as a function of a product of factors - key component composition; oil saturation; capillary number (i.e. flow velocity).

A capillary number is dimensionless velocity representing the ratio of viscous to interfacial tension forces. For typical values of velocity  $v = 1 \text{ ft/day}$  ( $0.3 \text{ m/day}$ ), viscosity  $\mu = 1 \text{ cp}$  and interfacial tension  $\sigma = 30 \text{ dynes/cm}$ ;  $N_c = \frac{\mu v}{\sigma}$  is approximately  $1 \times 10^{-7}$ . Note that the ratio

$\sigma/\mu$  is equivalent to a reference velocity of  $v_r = 1 \times 10^7 \text{ ft/day}$  ( $3 \times 10^6 \text{ m/day}$ ). Often not all the required relative permeability and capillary pressure data change as a function of interpolation (e.g. oil-water relative permeabilities change as a function of surfactant concentration but not the gas-oil curves). In these cases the original curves for the latter properties are duplicated for the second input set.

As an example of this option, consider the addition of surfactant to a water-oil system. Figure D.16 illustrates how the water-oil interfacial tension is affected by the addition of surfactant, while Figure D.17 shows how the residual saturations can be expected to change with changing capillary number. The overall effect of surfactant addition to the water-oil relative permeability curves is given in Figure D.18. These curves demonstrate that as interfacial tensions are reduced to ultra-low values, residual saturations decrease and relative permeability curves approach straight lines. Intermediate values of capillary number indicate flow behavior between the high tension and low tension curves.

Finally, a word of caution when employing this option. As always, it is the nonlinearities associated with the relative permeabilities which are a major source of convergence problems (when they arise) in reservoir simulation. This statement is doubly true here since now relative permeabilities are functions of both saturation and interpolation parameter. It is always a good idea to plot the relative permeability curves employed to check that the nonlinearities are not too unreasonable.

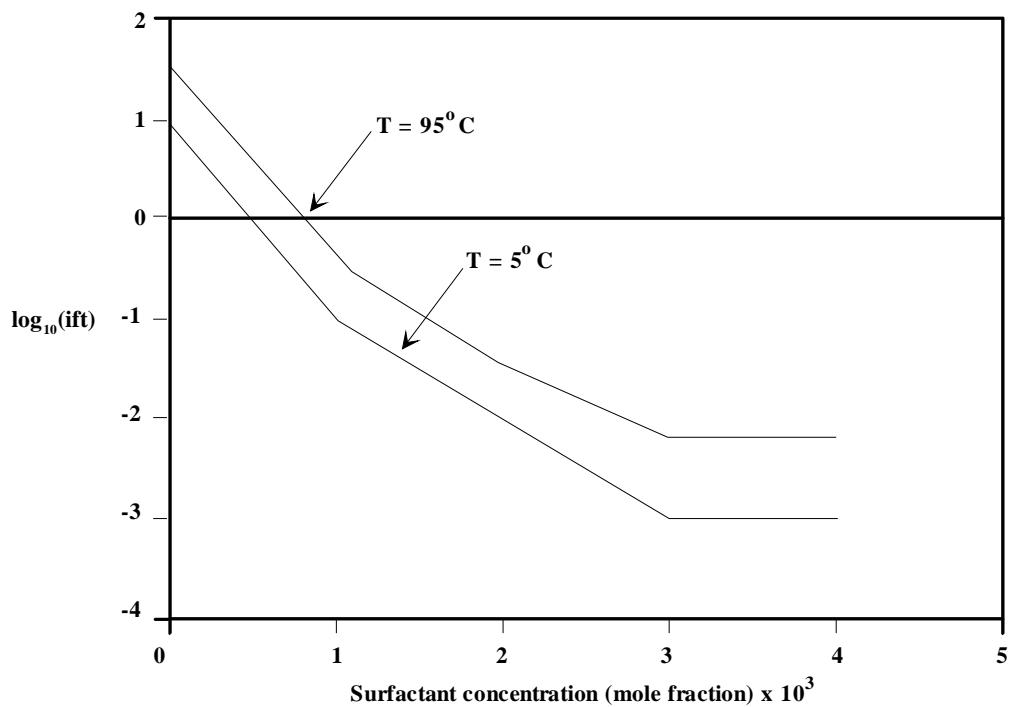


Figure D.16:  $\log_{10}$  (interfacial tension) versus surfactant concentration (mole fraction) at 2 temperatures

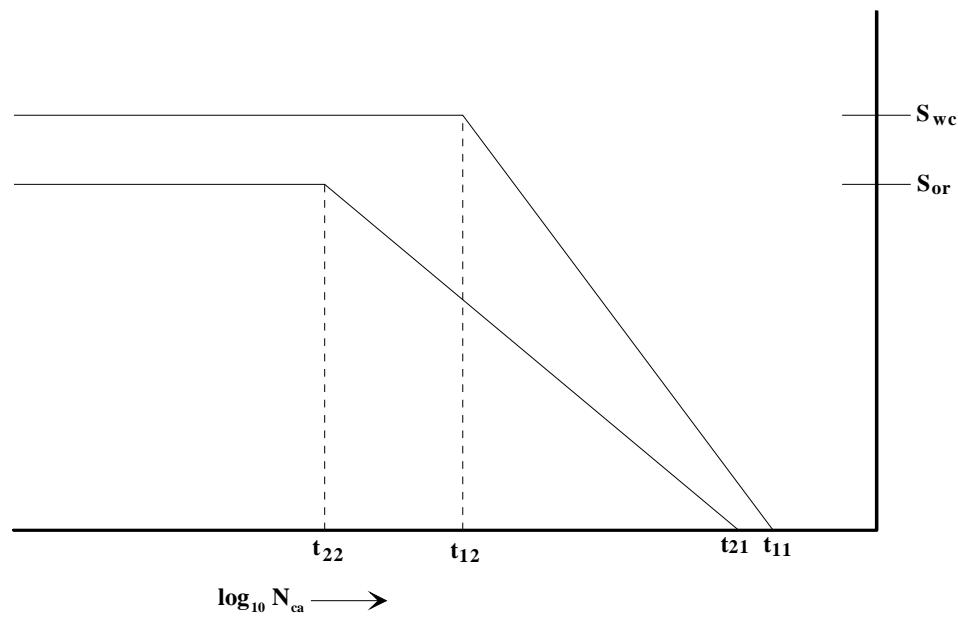
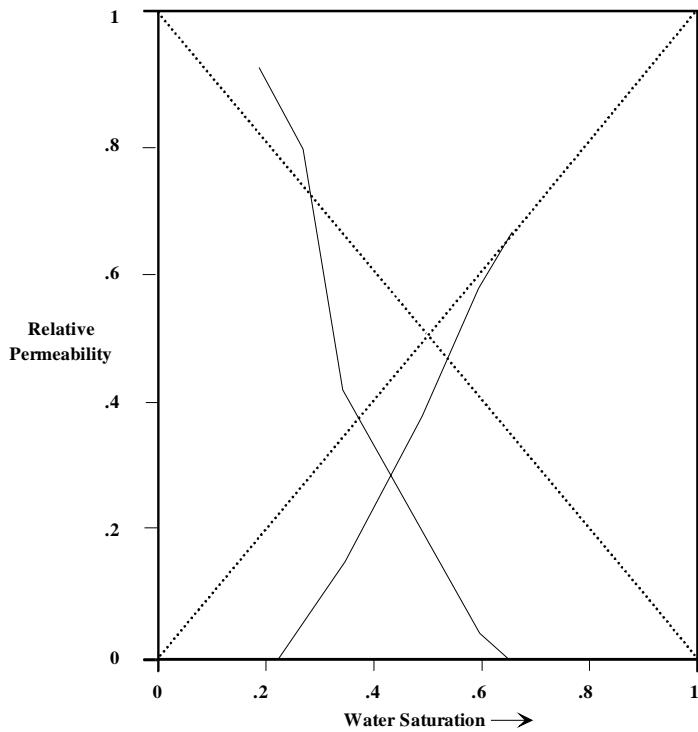


Figure D.17: Behavior of residual saturations as a function of the logarithm of capillary number. Example detrapping parameters ( $t_{22} = -4.22$ ;  $t_{12} = -4.00$ ;  $t_{21} = -2.22$ ;  $t_{11} = -1.68$ )



**Figure D.18:** High tension (— no surf) and low tension (---- surf) water-oil relative permeability curves

### Temperature Dependence

When relative permeability data is entered, critical saturations ( $S_{wr1}$ ,  $S_{wr2}$ , etc.) and end points ( $krwol$ ,  $krwo2$ , etc) at two temperatures  $T_1$  and  $T_2$  are obtained from the table entries or are specified explicitly; table entries are then scaled from 0 to 1. When a relative permeability at  $T$  is required, the critical saturations are evaluated (Table D.4) and then the dimensionless saturations  $S_{ew}$  and  $S_{el}$  are calculated from  $S_w$  and  $S_o$  (see the Stone's models, below). The four two-phase values are obtained via table lookup, and then adjusted to temperature  $T$  according to Table D.5.

Quantity	Slope	Interpolation
$S_{wc}$	$(S_{wr2}-S_{wr1})/(T_2-T_1)$	$S_{wr1} + \text{slope} \bullet (T-T_1)$
$S_{orw}$	$(S_{orw2}-S_{orw1})/(T_2-T_1)$	$S_{orw1} + \text{slope} \bullet (T-T_1)$
$S_{org}$	$(S_{org2}-S_{org1})/(T_2-T_1)$	$S_{org1} + \text{slope} \bullet (T-T_1)$
$S_{gc}$	$(S_{gr2}-S_{gr1})/(T_2-T_1)$	$S_{gr1} + \text{slope} \bullet (T-T_1)$

$T$  is constrained to fall in the range  $(T_1, T_2)$

**Table D.4:** Critical saturation temperature dependence

Quantity	Slope	Interpolation
$k_{rw}$	$(k_{rwro2}/k_{rwro1}-1)/(T_2-T_1)$	$k_{rw}(S_{ew}) \bullet [1 + \text{slope} \bullet (T - T_1)]$
$k_{row}$	$(k_{rocw2}/k_{rocw1}-1)/(T_2-T_1)$	$k_{row}(S_{ew}) \bullet [1 + \text{slope} \bullet (T - T_1)]$
$k_{rog}$	$(k_{rgcw2}/k_{rgc1}-1)/(T_2-T_1)$	$k_{rog}(S_{el}) \bullet [1 + \text{slope} \bullet (T - T_1)]$
$k_{rg}$	$(p_{cg2}/p_{cg1}-1)/(T_2-T_1)$	$k_{rg}(S_{el}) \bullet [1 + \text{slope} \bullet (T - T_1)]$
$p_{cow}$	$(p_{cw2}/p_{cw1}-1)(T_2-T_1)$	$p_{cow}(S_{ew}) \bullet [1 + \text{slope} \bullet (T - T_1)]$
$p_{cog}$	$(p_{cg2}/p_{cg1}-1)(T_2-T_1)$	$p_{cog}(S_{el}) \bullet [1 + \text{slope} \bullet (T - T_1)]$
T is constrained to fall in the range ( $T_1, T_2$ )		
$k_{rw}(S_{ew})$ etc., are table look-ups at $T_1$		

**Table D.5: Relative permeability temperature dependence**

Finally, a Stone's model (usually model II) is used to estimate the middle phase relative permeability,  $k_{ro}$ .

### Stone's Three-Phase Model II (modified)

The three-phase oil relative permeability calculation is<sup>4</sup>:

$$S_{ew} = (S_w - S_{wc}) / (1 - S_{orw} - S_{wc})$$

$$S_{el} = (S_w + S_o - S_{lc}) / (1 - S_{gc} - S_{lc})$$

$$k_{rw} = k_{rw} (S_{ew})$$

$$k_{rg} = k_{rg} (S_{el})$$

$$k_{ro} = k_{rocw} \bullet [(k_{row}/k_{rocw} + k_{rw}) \bullet (k_{rog}/k_{rocw} + k_{rg}) - k_{rw} - k_{rg}] \quad (\text{D6.1})$$

where  $k_{rocw} = k_{row}$  ( $S_w = S_{wc}$ ) =  $k_{rog}$  ( $S_g = 0$ ) to ensure that  $k_{ro} = k_{rw}$  when  $S_g = 0$  and that  $k_{ro} = k_{rg}$  when  $S_w = 0$  (or  $S_{wc}$ ). If a negative value is calculated for  $k_{ro}$ , then  $k_{ro} = 0$  is assumed. The critical liquid saturation may or may not contain critical water. Even if all the two-phase data is consistent and physical, Stone's model II can predict  $k_{ro} > 0$  when  $S_o = 0$ , which will cause nonconvergence (flow of nonexistent oil). The STARS model will perform a check for this condition when the three-phase table is printed.

### Stone's Three-Phase Model I (modified)

Stone's model I is constrained by the requirements that  $S_{gc} = 0$  and  $S_{om}$  varies between  $S_{orw}$  and  $S_{org}$ . This option is available only when  $S_{wc}$  is included in the liquid saturation. The three-phase oil relative permeability calculation is<sup>13</sup>

$$S_{eo} = (S_o - S_{om}) / (1 - S_{wc} - S_{om})$$

$$S_{ew} = (S_w - S_{wc}) / (1 - S_{wc} - S_{om})$$

$$S_{eg} = S_g / (1 - S_{wc} - S_{om})$$

$$S_{el} = 1 - S_{eg} \quad (\text{D6.2})$$

$$\beta_w = k_{row} (S_{ew}) / (k_{row} \bullet (1 - S_{ew}))$$

$$\beta_g = k_{rog} (S_{el}) / (k_{row} \bullet (1 - S_{eg}))$$

$$k_{rw} = k_{rw} (S_{ew})$$

$$k_{rg} = k_{rg} (S_{el})$$

$$k_{ro} = k_{row} \bullet S_{eo} \bullet \beta_w \bullet \beta_g \quad (D6.3)$$

where  $k_{row} = k_{row} (S_w = S_{wc}) = k_{rog} (S_g = 0)$  to ensure that  $k_{ro} = k_{row}$  when  $S_g = 0$  and that  $k_{ro} = k_{rog}$  when  $S_w = S_{wc}$ .

The "minimal" value of oil saturation  $S_{om}$  is the linear function of  $S_g$  proposed by Fayers and Matthews (SPEJ April 1984, pp. 224-232):

$$S_{om}(S_g) = (1 - a(S_g)) * S_{orw} + a(S_g) * S_{org},$$

where

$$a(S_g) = S_g / (1 - S_{wcrit} - S_{org}).$$

### Critical Water in Liquid

The liquid saturation  $S_l$  in the liquid-gas data from labs either contains irreducible water or not. Consider the following lab procedure.

1. Flood evacuated core with water to get  $S_l = 1$  and  $S_w = 1$ ,
2. Flood with oil until irreducible water is reached, resulting in  $S_l = 1$  and  $S_w = S_{wc}$ ,
3. Optionally, flood with water to get imbibition data, but flood with oil to return to  $S_l = 1$  and  $S_w = S_{wc}$ , and
4. Flood with gas until irreducible liquid is reached, whereby  $S_l = S_{lc}$  and  $S_w = S_{wc}$ .

This lab sequence will result in liquid-gas data which include critical water in  $S_l$ . However, some gas floods are done from an oil-saturated core. Make sure the data is assigned correctly.

Notes:

1. The curve of  $k_{rg} (S_g)$  can be the major source of nonlinearity in a steam-injection simulation.
2. The amount of mobile water ( $S_{wi} - S_{wc}$ ) is the major factor in determining initial injectivities in heavy oil problems.

## D.7 Component Adsorption and Blockage

The rate of propagation of many additives (surfactants, caustic, and polymers) and in situ created species (fines, emulsions) are strongly affected by their interaction with the rock matrix. These interactions can be chemical (e.g. ion exchange) or mechanical (e.g. blockage, straining capture) or some combination of mechanisms. The capture levels can depend on fluid concentrations, temperature and rock type (e.g. permeability).

STARS allows a phenomenological description of these phenomena, wherein a set of constant temperature adsorption isotherms (adsorption level as a function of fluid composition) are input. These isotherms can be either in tabular form or in terms of the well known Langmuir isotherm correlation

$$AD = \frac{Az}{1 + Bz} \quad (D7.1)$$

where  $z$  is some fluid component composition and where  $A$  and  $B$  are generally temperature dependent. The component and the fluid phase are specified by the user. Note the maximum adsorption level associated with this formula is  $A/B$ . Isotherms for up to four different temperatures can be supplied. Most often it is expected that adsorption decreases as temperature increases. Multiple components may adsorb, each with their individual isotherms, although it is assumed that individual species adsorb independently.

Some discussion of adsorption units is now given. Because of the form of the adsorption term in the flow equations,  $\frac{\partial}{\partial t} [\phi Ad_i]$ , simulator adsorption levels are described as moles (or mass) of component  $i$  adsorbed per unit volume. A variety of other measures of adsorption levels can be employed, and these must be appropriately converted for simulator input requirements.

Example: A surfactant adsorption maximum of 2.0 mg/g rock is reported at a water phase concentration of 0.4 wt%. Convert this information to appropriate Langmuir isotherm values.

The major task is to convert rock mass fraction information to pore volume. Porosity and rock density information are required. Assume porosity  $\phi = 0.3$  and rock density  $\rho_r = 2.7$  g/cm<sup>3</sup> (quartz). Then

$$\begin{aligned} Ad_i &= 2.0 \frac{mgsurf}{grock} \times \frac{(1 - \phi)\rho_r}{\phi} \frac{grock}{cm^3 PV} \times 10^{-3} \frac{gsurf}{mgsurf} \\ &= 1.25 \times 10^{-2} \frac{gsurf}{cm^3 PV} \end{aligned} \quad (D7.2)$$

Ideally, not only the adsorption maximum but the rate of increase of adsorption with fluid composition should be known in order to fit the two Langmuir parameters  $A$  and  $B$ . If this is not reported, as is often the case, one must use the fluid composition at the adsorption maximum to indirectly determine this second factor. The Langmuir isotherm equation can be rewritten as

$$Ad_i = \left( \frac{A}{B} \right) \times \frac{Bz_i}{1 + Bz_i} \quad (D7.3)$$

and the second factor is of order O(1) when  $B_{zi} \geq 10$ . In the above example, the maximum adsorption level occurs at 0.4 wt% which is equivalent to a mass fraction concentration of  $z_i = 0.004$ . Thus

$$B_{zi} = 10 \Rightarrow B = 2.5 \times 10^3 \quad (\text{D7.4})$$

With the value of B established, the adsorption maximum level can be used to determine A

$$\frac{A}{B} = 1.25 \times 10^{-2} \frac{\text{g surf}}{\text{cm}^3 \text{PV}} \Rightarrow A = 31.5 \frac{\text{g surf}}{\text{cm}^3 \text{PV}} \quad (\text{D7.5})$$

These are appropriate parameter values for laboratory mass fraction units. S.I. units are equivalently

$$A = 31.5 \times 10^{-3} \frac{\text{kg surf}}{\text{m}^3 \text{PV}} \quad (\text{mass fraction units}) \quad (\text{D7.6})$$

$$B = 2.5 \times 10^{-3}$$

In mole fraction units, the surfactant molecular weight is also required (e.g. MW = 400 g/mole) implying an adsorption maximum

$$\begin{aligned} Ad_i &= 1.25 \times 10^{-2} \frac{\text{g surf}}{\text{cm}^3 \text{PV}} \times \frac{1}{4 \times 10^2} \frac{\text{moles surf}}{\text{g surf}} \\ &= 3.15 \times 10^{-5} \frac{\text{moles surf}}{\text{cm}^3 \text{PV}} \end{aligned} \quad (\text{D7.7})$$

Similarly, the fluid mass fraction concentration is converted to mole fraction

$$\begin{aligned} \text{mole fraction surf} &= \text{mass fraction surf} \times \frac{\text{MW water}}{\text{MW surf}} \\ &= 0.004 \times \frac{18}{400} \\ &= 1.8 \times 10^{-4} \end{aligned} \quad (\text{D7.8})$$

Following a similar procedure to that outline above

$$B = 5.56 \times 10^4 \quad (\text{mole fraction units}) \quad (\text{D7.9})$$

$$A = 1.75 \frac{\text{moles surf}}{\text{cm}^3 \text{PV}}$$

or in SI units

$$A = 1.75 \times 10^6 \frac{\text{moles surf}}{\text{cm}^3 \text{PV}} \quad (\text{mole fraction units}) \quad (\text{D7.10})$$

$$B = 5.56 \times 10^4$$

Maximum adsorption levels ADMAXT and residual adsorption levels ADRT can be made region dependent, so that these properties can vary from grid block to grid block. Specification of residual adsorption levels allows the flexibility of modelling both reversible (i.e. chemical) adsorption  $ADRT = 0.0$  and irreversible (i.e. mechanical) adsorption  $ADRT = ADMAXT$ , as well as partially reversible process.

Permeability alteration often accompanies adsorption (especially if adsorption is of mechanical, blockage type). The simulator accounts for this via region dependent resistance factors RRF which allow correlation of local permeability with local adsorption levels - it is assumed that only single-phase flow paths are altered. Thus for example, the water phase permeability reduction factor is defined as

$$RKW = 1.0 + (RRF - 1.0) AD/ADMAXT \quad (D7.11)$$

which varies between 1.0 and a maximum of RRF as adsorption level increases. The mobility of the water phase is divided by RKW, thus accounting for blockage.

## D.8 A Simple Foam Model

A simple (quasi-equilibrium) approach to foam modelling can be employed utilizing several property options described in this chapter. The basic assumption utilized in this approach is that foam creation and coalescence mechanisms occur rapidly relative to flow such that whenever gas and aqueous surfactant coexists - foam exists. The advantages of this approach are

- Relatively limited experimental information is required
- One extra flow equation (for surfactant) is employed (implying limited increased simulation costs)
- Simplicity allows rapid screening tests and field pilot matching to be effectively utilized

Furthermore, this approach provides a useful background for more complex (i.e. mechanistic) studies of foam flow, as outlined in Section D.16.

The basic requirement of this model is a correct description of the flow of surfactant component. Surfactant data requirements include:

- Gas/oil K-values: Normally surfactant can be considered nonvolatile,  $K_{gw} \geq 0.0$
- Oil/water K-values: Normally foam forming surfactants have limited partitioning into the oil (for steam foams, K-values at initial and steam temperatures are minimum requirements)
- Adsorption maxima (or adsorption isotherms): Normally surfactant adsorption decreases with increasing temperature (for steam foams, maxima at initial and steam temperatures are minimum requirements)
- Surfactant half life (surfactant decomposition rate): Normally surfactant decomposition increases with temperature (for steam foams, rates at initial and steam temperatures are minimum requirements). First order decomposition rate constant  $k_d$  (at basic pH), or equivalently, surfactant half-life  $t_{1/2}$  (since  $k_d = 0.693/t_{1/2}$ ).

In addition, foam effects on gas mobility and flow pathways are handled via modified relative permeability curves. Because of the form of the simulation flow equations (gas mobility  $\lambda_g = k_g/\mu_g/r_g$ ) reduced gas relative permeability is equivalent to increased gas viscosity, increased gas resistance factor, or combinations of these factors. Utilizing relative permeability curves to describe combination of effects appears the most flexible method. Experimental observations indicate that foam mobility is sensitive to surfactant concentration; gas flow velocity (or capillary number); and the presence of oil. A simple interpolation scheme based on these parameters employs a dimensionless interpolation factor FM,

$$FM = \left\{ 1 + MRF \left( \frac{ws}{ws_{max}} \right)^{es} \left( \frac{S_o^{max} - S_o}{S_o^{max}} \right)^{eo} \left( \frac{N_c^{ref}}{N_c} \right)^{ev} \right\}^{-1} \quad (D8.1)$$

varying between  $FM = 1$  (no foam) and  $FM \sim (MRF)^{-1}$  (strongest foam).

Here the simplest choices for the exponents are  $es = 1.0$  to  $2.0$ ;  $eo = 1.0$  to  $2.0$ ; and  $ev = 0.3$  to  $0.7$ . Explicit consideration of foam rheology can detrimentally affect the numerical performance of the run so that it is often advisable to estimate foam mobility effects at the gas velocity of interest and set  $ev = 0.0$  (no rheology).

Based on these ideas, the additional foam mobility data requirements are

- $k_{rg}$ ,  $k_{rw}$ ,  $k_{ro}$  (no foam): Normally two phase gas/oil and water/oil curves plus Stone's model are employed. These curves define the basic flow pathways that the phase will take even in the presence of foam.
- Maximum mobility reduction factor MRF. Normally obtained from foam water flow experiments at maximum surfactant concentration  $w_s^{\max}$  a reference flow rate or capillary number  $N_c^{\text{ref}}$ , and no oil in the core;

$$\text{MRF} = \frac{(\Delta P)_{\text{foam}}}{(\Delta P)_{\text{nofoam}}}$$

This factor is used to scale (reduce) the gas relative permeability curve and can vary from 5.0 to 500.0.

- Interpolation parameters:

$w_s^{\max}$  – Maximum surfactant concentration needed to obtain a strong foam (normally about 1.0 weight percent)

$s_o^{\max}$  – Maximum oil saturation above which no foam will form (normally 10 to 30 percent depending on choice of surfactant)

$N_c^{\text{ref}}$  – Reference capillary number (equivalently, reference gas velocity) of the experiment

The simplest characterization of the effect of foam is to input similar gas relative permeability curves for the no foam and maximum foam cases - only the end point is rescaled downwards - see Figure D.19. A more complex characterization might be to include increased gas trapping - see Figure D.19. Even more complex foam behavior can be modelled with the input of intermediate curves as a function of FM. Finally, Figure D.20 illustrates the nature of the chosen interpolation method for the simplest example.

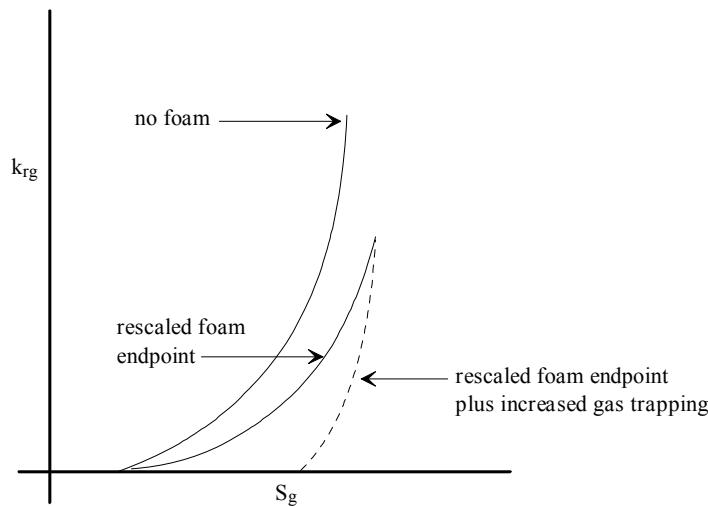
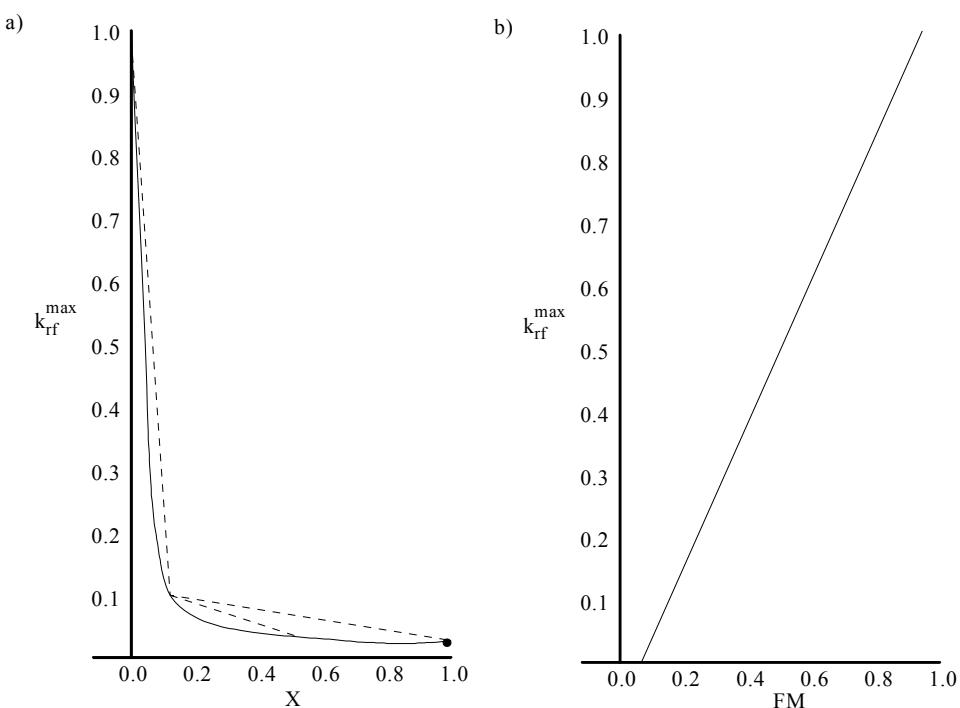


Figure D.19: Schematic of simple characterization of foam effects on gas relative permeabilities



**Figure D.20: Example of rescaling gas phase relative permeability end point – two views of how the interpolation scheme operates**

a)  $k_{rf}^{max}$  as a function of  $X = w_s / w_x^{max}$  and b)  $k_{rf}^{max}$  as a function of  $FM = (1 + MRFX)^{-1}$ .

Here  $k_{rf}^{max} \equiv / (1 + MRFX)$     $k_{rf}^{max} = 1.0$  and  $MRF = 100$

Some extra comments summarize this approach to foam modelling. Firstly, the wide variation in observed MRF factors probably reflects the type of foam created in the experiments. All foams are not equivalent! In this empirical approach, MRF should be viewed as a fitting parameter which can vary both with the type of laboratory experiment conducted, and scale (from laboratory to the field). Secondly, this quasi-equilibrium modelling should be viewed as an optimistic description of foam propagation, assuming as it does that the foam front moves with the same velocity as the surfactant front. More detailed analysis in the laboratory and in the field indicate that the foam (or pressure) front trails the surfactant front, moving at perhaps 2/3 the surfactant rate. This indicates that a certain fraction of created foam is non-mobile and the separation between surfactant and foam fronts increases as the relative proportion of trapped foam increases. Thus, if one attempts to fit observed pressure propagation to the surfactant flow equation as utilized in this simple model, an increased "pseudo-adsorption" of surfactant may be required to retard propagation. These concepts have been recently applied to the modelling of an Alberta steam foam pilot test<sup>14</sup>.

A more mechanistic approach to foam modelling, accounting for changing lamella densities, can be contrasted with the approach presented here - see section D.16 and a CMG foam report<sup>15</sup>. In particular, the form of the interpolation chosen for the simple model can be (at least partially) justified.

## D.9 Phase Enthalpies

Phase enthalpies and internal energies appear directly in the energy equation. They have units of energy per mole, resulting in an equation with units energy per time. Enthalpy fulfills the same function as does phase mole fraction (moles of component i per mole of phase) in the mass conservation equation. This is why the energy equation looks very similar to a component conservation equation.

The component enthalpies  $H_{Li}$  and  $H_{gi}$  are calculated from heat capacity and vaporization enthalpy correlations. Ideal mixing is assumed, so component enthalpies are independent of phase composition and pressure (except for water vapour enthalpy which depends on pressure).

### Water Phase

The phase value  $H_w$  is equal to the liquid value  $HL_1$  for the water component. The internal energy is

$$U_w = H_w - p_w / \rho_w \quad (D9.1)$$

Note that  $H_w$  includes the mechanical work  $p_w/\rho_w$ , which flows with the fluid and can be transformed into heat energy in another location in the reservoir. The formula for  $H_w$  is the same as  $H_o$  below, but uses  $w_i$  instead of  $x_i$ .

### Oil Phase

The phase value  $H_o$  is obtained by mixing the component liquid values

$$H_o = \sum_{i=1}^{n_c} x_i H_{Li} \quad (D9.2)$$
$$U_o = H_o - p_o / \rho_o$$

### Gas Phase

$$H_g = \sum_{i=1}^{n_c} y_i H_{gi} \quad (D9.3)$$
$$U_g = H_g - p_g / \rho_g$$

### Solid Phase

Heat capacity of the solid component is entered as data, and the internal energy is

$$U_s = C_{ps}(T - T_r) \quad (D9.4)$$

### Rock

The heat capacity of the rock is treated on a volumetric basis, since rock is handled by volume and not by moles. Since porosity is usually less than 40%, the rock can contain most of the injected heat. Therefore it is important to define the rock heat content accurately.

Rock heat content is described by

$$U_r = a(T - T_r) + \frac{1}{2} b (T^2 - T_r^2) \quad (D9.5)$$

where  $a$  is a constant volumetric heat capacity, and  $b$  is a temperature coefficient. A value of  $a = 35 \text{ Btu/ft}^3\text{-F}$  is used commonly.

## D.10 Thermal Conductivity

Thermal conductivity determines the flow term  $K\Delta T$  due to diffusion of energy from a region of high temperature to low temperature.

The only other way for energy to flow in situ is by convection. In field-scale steam problems convection usually dominates conduction, at least in the direction of flow. In field-scale combustion, the temperature profile at the fire front can be determined largely by conduction, but this temperature profile is almost never resolved because the grid blocks used are too large. For these reasons, conduction is rarely a major mechanism in field-scale problems. Conduction can play a significant role in both steam and combustion at the laboratory scale, since the length scale is much smaller than in the field.

The following are options for calculating an overall thermal conductivity from phase values. In each the porosity is fluid porosity  $\varphi_f$ .

### Linear Mixing

Thermal conductivities are weighted by volume

$$\kappa = \varphi \cdot [S_w \kappa_w + S_o \kappa_o + S_g \kappa_g] + (1-\varphi) \cdot \kappa_r \quad (\text{D10.1})$$

### Nonlinear Mixing

The thermal conductivities are weighted using the correlation of Anand et al<sup>16</sup>. The liquid-rock mixed value is

$$\kappa_{L-r} = \kappa_L \cdot a^b \quad (\text{D10.2})$$

where

$$\begin{aligned} \kappa_L &= (S_o \kappa_o + S_w \kappa_w) / (S_o + S_w) \\ a &= \kappa_r / \kappa_L \\ b &= 0.28 - 0.757 \cdot \log_{10} \varphi - 0.057 \cdot \log_{10} a \end{aligned} \quad (\text{D10.3})$$

The gas-rock mixed value is

$$\kappa_{g-r} = \kappa_g \cdot c^d \quad (\text{D10.4})$$

where

$$\begin{aligned} c &= \kappa_r / \kappa_g \\ d &= 0.28 - 0.757 \cdot \log_{10} \varphi - 0.057 \cdot \log_{10} c \end{aligned} \quad (\text{D10.5})$$

The gas-liquid-rock mixed value is

$$\kappa_{g-L-r} = (1-e) \cdot \kappa_{g-r} + e \cdot \kappa_{L-r} \quad (\text{D10.6})$$

where

$$e = \sqrt{S_w + S_o} \quad (\text{D10.7})$$

Example: Let values for rock and water be 44 and 8.6 Btu/ft-day-F, respectively, with  $\varphi = 0.3$ . The value of  $\kappa$  for water saturated rock ( $S_w = 1$ ) is given by

$$\kappa_L = \kappa_w = 8.6$$

$$a = 44 / 8.6 = 5.116 \quad (D10.8)$$

$$b = 0.6354$$

so

$$\kappa_{L-r} = 8.6 \cdot a^b = 24.3 \text{ Btu / ft - day - F.} \quad (D10.9)$$

Compare this with the linear option, which gives

$$\kappa_{L-r} = 0.3 \cdot 8.6 + 0.7 \cdot 44 = 33.4 \text{ Btu / ft - day - F.} \quad (D10.10)$$

Basically the linear option assumes that all phases (including solid) are randomly mixed in a porous medium while the nonlinear option implies some type of correlated distribution of phases (reflecting the fact that the liquids tend to wet the rock).

### Temperature Dependence

This modification of Somerton et al<sup>17</sup> accounts for the observed change in thermal conductivity as temperature is increased. In the STARS model this modification may be done after the mixed liquid-gas-rock value has been calculated. The unit of  $\kappa$  is J/m-day-K.

$$\kappa = a - 1.7524 \cdot 10^{-5} (T - T_r) \cdot (a - 119616) \cdot b \cdot c \quad (D10.11)$$

where

$$a = \kappa_{g-L-r}$$

$$b = a^{-0.64}$$

$$c = a \cdot d^e + 110644.8 \quad (D10.12)$$

$$d = 1.8 \cdot 10^{-3} \cdot T, \text{ where } T \text{ is in K, and}$$

$$e = -3.6784 \cdot 10^{-6} \cdot a$$

## D.11 Overburden Heat Loss

A semi-analytical model is used for heat transfer to or from an adjacent formation of infinite extent<sup>18</sup>. It assumes a temperature profile in the base or cap rock as a function of time and distance  $z$  from the reservoir interface, i.e.,

$$T(t, z) = (\theta + pz + qz^2) e^{-z/d} \quad (D11.1)$$

The diffusion length is  $d = \sqrt{\lambda t}/2$ . This profile satisfies both  $T(t, 0) = \theta$  (grid block temperature) and  $T(t, \infty) = 0$ . At the interface the profile must satisfy

$$\frac{\partial \theta}{\partial t} = \lambda \frac{\partial^2 T}{\partial z^2} \Big|_{z=0} \quad (D11.2)$$

or

$$\frac{\theta - \theta^N}{\Delta t} = \lambda \left( \frac{\theta}{d^2} - \frac{2p}{d} + 2q \right) \quad (D11.3)$$

when the time derivative is discretized. The conservation of energy is expressed as

$$\frac{\partial}{\partial t} \int_0^\infty T dz = \lambda \frac{\partial T}{\partial z} \Big|_\infty \quad (D11.4)$$

or

$$\frac{\theta d + pd^2 + 2qd^3 - I^N}{\Delta t} = \lambda \left( \frac{\theta}{d} - p \right) \quad (D11.5)$$

where  $I^N = \theta N dN + pN (dN)^2 + 2qN (dN)^3$ .

Isolating parameters  $p$  and  $q$  gives

$$p = \frac{\left[ \frac{\lambda \Delta t \theta}{d} + I^N - \frac{d^3(\theta - \theta^N)}{\lambda \Delta t} \right]}{\left[ 3d^2 + \lambda \Delta t \right]} \quad (D11.6)$$

$$q = \frac{\left[ 2pd - \theta + d^2 \frac{\theta - \theta^N}{\lambda \Delta t} \right]}{2d^2}. \quad (D11.7)$$

The total energy in the overburden, i.e. under the temperature profile, is

$$\frac{\kappa}{\lambda} \int_0^\infty T dz = \frac{\kappa}{\lambda} d \left[ \theta + pd + 2qd^2 \right] \quad (D11.8)$$

The rate at which heat is lost from the block to the overburden is

$$-\kappa \frac{\partial T}{\partial z} \Big|_{z=0} = \kappa \left( \frac{\theta}{d} - p \right) \quad (D11.9)$$

The heat loss rate and its derivative with respect to temperature can be calculated, and used directly in the energy conservation derivatives. The energy lost is used directly in the energy balance statistics.

One set of these calculations is performed for each grid block face which is adjacent to cap or base rock. The only data required are thermal conductivity and heat capacity of the base and cap rock, both of which are quite standard data. This model gives solutions quite close to exact answers to many heat transfer problems.

## D.12 Thermal Aquifer

A semi-analytical model is used to calculate the water and energy flow to/from an adjacent aquifer. It is based on a one-dimensional single-phase (water) and energy flow. This flow is perpendicular to the reservoir-aquifer boundary and may be either linear or radial. The model assumes temperature/pressure profile as

$$TP(t, x) = \left( TP_{ra} - TP_{ref} + px + qx^2 \right) e^{-x/d} + TP_{ref} \quad (D12.1)$$

which depends on the temperature/pressure value at a reservoir-aquifer interface  $TP_{ra}$ , the aquifer height as well as the fluid mobility and rock and fluid heat conductivity and capacity.

At a reservoir-aquifer boundary the temperature/pressure value is equal to the grid block temperature/pressure  $TP(t, 0) = TP_{ra} - TP_{ref}$ .

At an infinite boundary the value is the initial temperature/pressure  $TP(t, \infty) = TP_{ref}$ .

In an aquifer of finite size, the temperature/pressure will rise or fall according to the flow direction and the heat loss to adjacent formation.

The diffusion length

$$d = \frac{\sqrt{\alpha t}}{2}, \quad (D12.2)$$

where

$$\alpha = \left[ \frac{k}{\varphi \mu (c_r + c_w)} \right]_{aq} \text{ for a water flow} \quad (D12.3)$$

and

$$\alpha = \left[ \frac{\varphi \kappa_w + (1-\varphi) \kappa_r}{\varphi C_{pw} \rho_w + (1-\varphi) C_{pr}} \right]_{aq} \text{ for energy flow} \quad (D12.4)$$

$p$  and  $q$  are fitting parameters which are calculated from the water and energy balance equations with appropriate boundary conditions.

The water balance equation for a linear flow is

$$\frac{\partial P}{\partial t} = \alpha \frac{\partial^2 P}{\partial x^2} \quad (D12.5)$$

The energy balance equation for a linear flow is

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} - u \frac{\partial T}{\partial x} \quad (D12.6)$$

where  $u$  is a velocity of a moving heat front

$$u = \left[ \frac{u_w \rho_w C_{pw}}{\varphi \rho_w C_{pw} + (1-\varphi) C_{pr}} \right]_{aq} \quad (D12.7)$$

Because the energy balance equation is more complex, this will be used to show the evaluation of p and q parameters for a finite aquifer.

At the reservoir-aquifer interface the temperature profile must satisfy

$$\frac{\partial T_{ar}}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \Big|_{x=0} - u \frac{\partial T}{\partial x} \Big|_{x=0} \quad (D12.8)$$

or

$$\frac{\Delta_t T_{ar}}{\Delta t} = \alpha \left( \frac{T_{ar}}{d^2} - \frac{2p}{d} + 2q \right) - u \left( -\frac{T_{ar}}{d} + p \right) \quad (D12.9)$$

by substituting the derivatives.

The parameter q is then:

$$q = \frac{\frac{d^2}{\alpha} \frac{\Delta_t T_{ar}}{\Delta t} + \frac{ud}{\alpha} (pd - T_{ar}) - T_{ar} + 2pd}{2d^2} \quad (D12.10)$$

The conservation of energy in the aquifer will be satisfied by integrating the energy equation ( $\alpha$  and  $u$  are assumed to be constant)

$$\frac{\partial}{\partial t} \int_0^h T dx = \alpha \frac{\partial T}{\partial x} \Big|_0^h - u T \Big|_0^h \quad (D12.11)$$

From this equation the parameter p is calculated as:

$$p = \frac{I^N - u \Delta t T_{ar} + T_{ar} \text{coef1} - \frac{\Delta_t T_{ar}}{\Delta t} \text{coef2} + H_{loss}}{\text{coef3}} \quad (D12.12)$$

where

$$I^N = \left\{ T_{ar} d + pd^2 + 2d^3 q - e^{-h/d} \left[ T_{ar} d + p(d^2 + dh) + q(2d^3 + 2d^2 h + dh^2) \right] \right\} \quad (D12.13)$$

is the N-level integral of temperature,

$$\text{coef1} = \frac{\alpha \Delta t}{d} + \frac{ud^2}{\alpha} - e^{-h/d} \left[ \frac{ud^2}{\alpha} + \left( 1 + \frac{h}{2d} \right) \left( h + \frac{hdu}{\alpha} \right) \right] \quad (D12.14)$$

$$\text{coef2} = \frac{d^3}{\alpha} - e^{-h/d} \left[ \frac{d^3}{\alpha} + \frac{d^2 h}{\alpha} \left( 1 + \frac{h}{2d} \right) \right] \quad (D12.15)$$

$$\text{coef3} = 3d^2 + \alpha \Delta t + \frac{ud^3}{\alpha} - e^{-h/d} \left[ 3d^2 + \frac{ud^3}{\alpha} + h(3d + h) + \frac{hd^2 u}{\alpha} \left( 1 + \frac{h}{2d} \right) \right] \quad (D12.16)$$

and  $H_{loss}$  is the conductive heat loss to adjacent formation. The parameter p and q for the pressure profile are similar (terms with velocity  $u$  and  $H_{loss}$  are zero).

The rate per unit area at which water flows to/from an aquifer is expressed as:

$$qaq_w = -\lambda \frac{\partial P}{\partial x} \Big|_{x=0} = -\lambda \left( p - \frac{T_{ar}}{d} \right) \quad (D12.17)$$

$$\lambda = \frac{k_{gv} k_{rw}}{\mu} \text{ for water flow from reservoir to aquifer} \quad (D12.18)$$

$$\lambda = \frac{k_{av}}{\mu_{aq}} \text{ for reversed flow} \quad (D12.19)$$

The energy flow rate per unit area of the reservoir-aquifer interface is

$$qaq_E = -\kappa \frac{\partial T}{\partial x} \Big|_{x=0} + u_w H_w \rho_w \quad (D12.20)$$

$$\kappa = (1 - \varphi) \kappa_r + \varphi \kappa_w S_w \text{ flow from reservoir to an aquifer} \quad (D12.21)$$

$$\kappa = (1 - \varphi_{aq}) \kappa_{r_{aq}} + \varphi_{aq} \kappa_{w_{aq}} \text{ for reversed flow} \quad (D12.22)$$

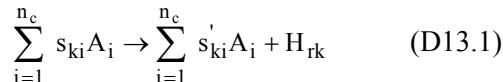
The flow rates are calculated for each grid block adjacent to the aquifer and the values are used in the reservoir flow equations as source/sink terms. The simulation will automatically stop when the temperature at the interface approaches the steam temperature value (gas phase is not allowed to appear in the aquifer).

This model gives adequate results when the flow of oil from a reservoir to the aquifer is negligible and when transverse water and energy flow in the aquifer does not play a major role.

## D.13 Chemical Reactions

Chemical reactions have traditionally been used almost exclusively in combustion processes. However, reactions may be used in any thermal or isothermal simulation if desired. Since reactions are treated as source/sink terms for each component and energy, they may be thought of as another way in which to link together the different components of a problem when rate is important. In particular, interphase mass transfer rates can be modelled, involving either well defined components or "dispersed phase" components such as emulsion droplets.

The general heterogeneous mass transfer reaction no. k is represented symbolically as



which proceeds at the rate of  $r_k$  moles per day per reservoir volume. As expressed above, this relationship has one degree of freedom, which is a proportionality factor. The quantities  $s_{ki}$ ,  $s'_{ki}$  and  $H_{rk}$  can be multiplied by an arbitrary factor  $a$ , but  $r_k$  must be divided by  $a$  so that the source/sink terms remain

$$(s'_{ki} - s_{ki}) \bullet r_k \text{ and } H_{rk} r_k \quad (D13.2)$$

Usually the factor  $a$  is chosen such that  $s_{ki} = 1$  for the main reacting component.

### Kinetic Model

The kinetic model, also known as reaction kinetics, determines the speed of reaction  $r_k$ . The general expression is

$$r_k = r_{rk} \bullet \exp(-E_{ak}/RT) \bullet \prod_{i=1}^{n_c} C_i^{e_k} \quad (D13.3)$$

The activation energy  $E_{ak}$  determines the temperature dependence of  $r_k$ . While the enthalpies of reaction can be characterized between well defined limits (and can even be calculated from first principles), the observed activation energies can vary dramatically. This is because certain components in the rock surface can act as catalysts. The concentration factor for reacting component i is

$$C_i = \varphi_f \rho_j S_j x_{ji} \quad j = w, o, g \quad (D13.4)$$

where  $j$  is the phase in which component i is reacting, and  $x_{ji}$  represents water, oil or gas mole fractions. For the solid component

$$C_i = \varphi_v c_i \quad (D13.5)$$

The partial pressure form  $C_i = y_i p_g$  is available also.

The factor  $r_{rk}$  is the constant part of  $r_k$ . Its unit can be quite complex, and must account for the units of the various  $C_i$ , which are moles per pore volume or pressure, raised to the power of  $e_{ik}$  and then multiplied together.

The kinetic model can represent a reacting component in only one phase at a time. If a component reacts in more than one phase, it must be modelled in two separate reactions.

## Field-Scale Combustion

In the expression for reaction rate, above, the temperature T is the temperature in the burning or reacting region. In the thermal model the grid block temperature is used. In lab scale simulations, a grid block is usually not much larger than the actual combustion burning region (10-20 cm), so use of a grid block temperature for reaction temperature is appropriate.

However, field-scale grid blocks usually are many times larger than this, and the averaged block temperature is not a good indication of the peak temperature of a combustion zone. Unrealistic predictions for fuel lay-down and front extinction will result<sup>20</sup>.

One solution is to use an assumed front temperature  $T_f$ , by following these steps.

1. Replace  $r_{rk}$  with  $r'_{rk} = r_{rk} \cdot \exp(-E_{ak}/RT_f)$
2. Replace  $E_{ak}$  with zero.

This has the effect of making the reaction rate independent of grid block temperature. Other techniques have also been proposed<sup>21</sup>.

## Mass and Volume Conservation

Because the component conservation equations have mole units and the reactions are treated as source/sink terms, moles of each component and energy will be conserved. However, the reaction stoichiometry should be mass conserving as well in order for the reaction to make sense physically. This is important especially when the molecular weight of a pseudo-oil component is not well-defined or is arbitrary.

Mass-conserving stoichiometry satisfies

$$\sum_{i=1}^{n_c} s_{ki} M_i = \sum_{i=1}^{n_c} s'_{ki} M_i \quad (D13.6)$$

Even though a molecular weight is not required by the STARS model for the solid component, a reasonable value should be chosen for the above calculation.

If mass is not conserved in a reaction, the effect probably will not show up in the simulation until the final results are analyzed or compared with a laboratory report.

On the other hand, conservation of volume during reaction is not required in general. However, there is one condition under which large volume changes caused by reactions should be avoided. It is when  $S_g = 0$  and there are reactions between liquids, or between liquids and solids.

Consider a liquid-saturated reservoir ( $S_g = 0$ ) in which a heavy oil cracks into a solid fuel. Even though this reaction is meant to happen at higher temperatures, the model will calculate a nonzero reaction rate at the initial reservoir temperature. Therefore, some oil will be replaced by the solid from the start of the simulation. A significant discrepancy between the volumes consumed and produced, in conjunction with a low overall reservoir compressibility, will result in large uncontrollable pressure changes. This situation can be remedied by ensuring that volumes are more nearly conserved.

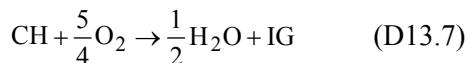
## Combustion Example

Table D.6 shows a typical component set for a combustion model.

Component	Symbol	M	Description
1	H <sub>2</sub> O	18	Water
2	HO	500	Heavy Oil
3	LO	200	Light Oil
4	IG	44	Inert Gas, CO <sub>2</sub> , N <sub>2</sub>
5	O <sub>2</sub>	32	Oxygen
6	CH	13	Solid Fuel

**Table D.6: Sample combustion component set**

The coke-burning reaction is



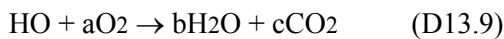
We can see that mass is conserved, since each side represents a molecular mass of 53. In addition, if we replace IG with CO<sub>2</sub>, we see that carbon, oxygen and hydrogen atoms are balanced.

Reaction enthalpies for burning hydrocarbons<sup>19</sup> commonly range from 16,000 to 22,000 Btu/lb. In this case, assume that the reaction enthalpy H<sub>rk</sub> is 6.3 • 10<sup>5</sup> J/gm mole or 20,830 Btu/lb, and activation energy E<sub>ak</sub> is 53,000 J/gm mole. The reaction rate is (T in K degrees).

$$r_1 = r_{r1} \cdot \exp(-53,000 / RT) \cdot [\varphi_v \cdot c_6] \cdot [y_5 \cdot p_g] \quad (\text{D13.8})$$

in units of gm mole per day per m<sup>3</sup>.

The reaction of heavy oil burning is



where a, b and c are stoichiometric coefficients. The task of finding them is simplified if we assume that component HO has the same hydrogen-carbon ratio as the CH component. Since a mole of HO is 500/13 = 38.46 times as massive as a mole of CH, the coefficients can be obtained directly from the coke-burning reaction.

$$a = (38.46) \frac{5}{4} = 48.08 \quad (\text{D13.10})$$

$$b = (38.46) \frac{1}{2} = 19.23 \quad (\text{D13.11})$$

$$c = (38.46)1 = 38.46 \quad (\text{D13.12})$$

Assume that the reaction enthalpy is 18,000 Btu/lb. This converts into  $2.09 \cdot 10^7$  J per gm mole of HO. So



Assuming the activation energy is the same as for coke burning, the reaction rate is

$$r_2 = r_{r2} + \bullet \exp(-53,000/RT) \cdot [\rho_f \rho_o S_o x_2] \cdot [y_5 p_g] \quad (\text{D13.14})$$

The constant factors  $r_{r1}$  and  $r_{r2}$  would be found by matching a combustion tube performance in which the observed average energy released per fuel consumed is observed to be, for example, 20,300 Btu/lb. Let  $a$  be the mass fraction of fuel burned via the coke burning reaction. Since energy is produced by either coke or oil burning, the mass fraction  $a$  satisfies

$$20,830 \cdot a + 18,000 \cdot (1 - a) = 20,300 \quad (\text{D13.15})$$

$$a = 0.813 \quad (\text{D13.16})$$

With this information the trial-and-error matching can proceed toward determining the values of  $r_{r1}$  and  $r_{r2}$ .

## D.14 Basic Concepts for Nonequilibrium Mass Transfer

The basic function of a simulator is to solve a set of equations that describe the flow and accumulation of a set of components or pseudo-components in a porous medium. Here we consider three components: Component 1 is water, component 2 is oil, and component 3 is surfactant. Flow can occur in the water (aqueous) phase and the oil (oleic) phase.

Components can accumulate in the fluid phases as well as on the surface of the matrix rock which is called the solid (immobile) phase. Most generally, there is a separate equation for each component in each phase when mass transfer between phases is **not** assumed to be at instantaneous equilibrium. The equations are

*Water Phase*

$$\frac{\partial}{\partial t}(\varphi_f S_w \rho_w w_i) + \nabla \cdot (\rho_w V_w w_i) = -R_{woi} - R_{wsi} + Q_{wi} \quad i=1,2,3 \quad (\text{D14.1})$$

*Oil Phase*

$$\frac{\partial}{\partial t}(\varphi_f S_o \rho_o x_i) + \nabla \cdot (\rho_o V_o x_i) = -R_{owi} - R_{osi} + Q_{oi} \quad i=1,2,3 \quad (\text{D14.2})$$

*Solid Phase*

$$\frac{\partial}{\partial t}(\varphi_v C_i) = -R_{swi} - R_{soi} \quad i=1,2,3 \quad (\text{D14.3})$$

where  $R_{jki} = -R_{kji}$  is the nonequilibrium net mass transfer of component  $i$  from phase  $j$  to phase  $k$ , for  $j \neq k$ .

Phase densities and viscosities are obtained from component values using the appropriate mixing rules. The external source terms  $Q_{wi}$  and  $Q_{oi}$  account for injection and production of fluids. The mass transfer of a molecularly dissolved component accounts for nonequilibrium phase partitioning. On the other hand, mass transfer of globules is interpreted as globules separating from or merging with the parent phase. The presence of an extra fluid phase (liquid or gas) would entail additional but analogous equations.

## D.15 Stable Emulsion Flow and In Situ Generation Concepts

In the presence of surfactants, liquid dispersions of oil-in-water or water-in-oil emulsions can form which are of sufficient stability to last for periods of hours, days or weeks. The stability and emulsion type are determined by factors such as temperature, ionic concentrations, pH, applied shear rate, and volume ratio of water to oil. Given that such quasi-equilibrium structures exist, the task here is to model how they interact with and flow through a porous medium. These emulsions can be injected externally or formed in situ. The major assumption is that they propagate unaltered through the reservoir, at least during the time of interest.

In the treatment of stable emulsion flow, the representation of fluid-fluid phase behavior is quite simple. An emulsion is called stable when:

1. Component mass transfer between phases is small enough to be neglected. This corresponds to zeroing all the liquid-liquid mass transfer terms  $R_{\text{W}}^i$  in the above equations.

The following assumptions are also employed

2. The surfactant component remains with the globules at all times, in a constant proportion. Therefore, explicit recognition of the surfactant as a separate component is not required, and all its flow equations can be dropped. This assumption is reexamined at the end of this section.
3. Only one emulsion type (oil-in-water or water-in-oil) is normally required at one time, eliminating two more equations. For concreteness we will consider an oil-in-water system.
4. The water and oil component in their continuous phase do not interact with the solid matrix, so that the corresponding mass transfer terms can be ignored.

These assumptions imply that only the transfer of an oil globule between the water and solid phases  $R_{\text{WSI}}$  is of interest. Possible relaxation of assumptions 1-3 can be considered but is not done here.

The modelling of particle capture, that is, mass transfer between the liquid globule and the solid matrix, is analogous to a filtration theory (solid particle-solid matrix) approach<sup>22</sup>. Two main capture mechanisms are thought to be operative: straining, in which droplets clog the pore throats, and interception, with droplets captured by van der Waals colloidal forces. These terms are also used in solid particle filtration, while in polymer flow, the analogous capture modes are termed mechanical entrapment and adsorption, respectively. For emulsions, there is usually a range of particle sizes, and straining dominates large particle capture while interception contributes primarily to small droplet capture. For most simulation purposes, it is probably sufficient to assume a monodisperse (uniform) emulsion size distribution, and the capture parameters employed would then reflect combined straining and interception modes. The possibility of re-entrainment of captured particles should also be considered; this is expected to come primarily from interception-captured droplets, and only after a critical velocity is achieved. In polymer flow, desorption is perhaps more prevalent than analogous emulsion re-entrainment, although the reason here is probably more chemical than mechanical.

As an example of globule-solid matrix mass transfer, consider the mass transfer term given by

$$R_{ws2} = kA(\varphi_f S_w \rho_w w_2) - kB(\varphi_f S_w \rho_w w_2)\varphi_v C_2 - kC\varphi_v C_2 \quad (D15.1)$$

describing oil-in-water emulsion capture and re-entrainment. Rate A corresponds to primary capture of the emulsion by the porous rock; rate B corresponds to microscopic flow diversion because previous rock capture sites are occupied; and rate C corresponds to re-entrainment. The rate constants  $k_A$ ,  $k_B$  and  $k_C$  are all functions of temperature. In solid filtration theory, the combination

$$\Lambda = k_A - k_B \varphi_v C_i \quad (D15.2)$$

represents a filtration coefficient which depends on the amount of captured material.

With the possibility of emulsion trapping comes the associated phenomena of blockage and reduction of permeability. This permeability reduction is normally expressed as a phase resistance factor  $RKJ > 1.0$ . The form of the resistance factor again depends on whether the trapping mechanism is assumed to be a quasi-equilibrium or nonequilibrium process. Section D.7 discusses quasi-equilibrium blockage while the analogous resistance factor expression for the nonequilibrium generation of solid (blocked) component is

$$RKW = 1.0 + RRSFT C_c \quad (D15.3)$$

for the water phase blockage as an example. The actual blockage factor RRSFT is expected to be permeability dependent. The simulator allows the tabular input of RRSFT versus permeability.

If emulsions are created in situ (as is generally observed in heavy oils<sup>23</sup>), the above approach must be generalized. In particular, the (oil in water) emulsion can no longer be viewed as dispersed oil globule, but rather as a new chemical species in the water phase consisting of both oil and surfactant molecules. This species is formed when molecularly dissolved surfactant contacts the oil phase. Thus an extra flow equation for surfactant is required, containing a heterogeneous mass transfer source term describing the emulsion creation process. The stoichiometry for the formation process is written as

$$(Oil)_o + a(S)_w = (E)_w \quad a \sim 10^{-4} \quad (D15.4)$$

with the stoichiometric coefficient "a" being small.

Estimates of reaction stoichiometry can be obtained by assuming that in a spherical emulsion globule E:

- a) The number of oil molecules largely outnumber the surfactant molecules, so that the volume of the globule is basically the volume occupied by the oil molecules and,
- b) The surfactant molecules occupy the surface of the globules.

These assumptions can be represented by the equations

$$\frac{\pi \langle D_d \rangle^3}{6} = v_o N_o \quad (D15.5)$$

$$\pi \langle D_d \rangle^2 = a_o N_s \quad (D15.6)$$

and be solved for  $N_o/N_s$ , the ratio of the number of oil to surfactant molecules in the globule. with typical values for globule diameter  $\langle D_a \rangle = 6 \times 10^{-4}$  cm, volume per oil molecule  $v_o = 35 \times 10^{-23}$  cm<sup>3</sup> and area per surfactant molecule  $a_o = 70 \times 10^{-16}$  cm<sup>2</sup>, the result is

$$\frac{N_s}{N_o} = a = 5 \times 10^{-4} \quad (D15.7)$$

With an assumed stoichiometry, the remaining task is to quantify the order and rate of the emulsion generation process. Very little work appears to have been done quantifying this to date, but more recent attempts at modelling foam generation should have important implications in emulsion generation modelling as well.

## D.16 A Lamella Density Model of Foam

A second example of dispersed component modelling is a lamella density<sup>24</sup> approach to modelling foam mechanisms. Here, a lamella "component" in the gas phase is defined, whose concentration determines the flow properties (viscosity, relative permeability, resistance factor) of the gas phase. A foam is a gas phase containing lamella.

Similarly to the flow of stable emulsions or polymers, pre-formed foam injected into the porous media is characterized by some capture (trapping) of its dispersed component species and this trapping mechanism could be viewed as either quasi-static (like polymer adsorption) or kinetic (like fines or emulsion capture). Obviously, this basic capture process affects both the rate of propagation and the local lamella density associated with stable foam motion. In addition, however, both in situ generation and coalescence of foam lamella also affect local lamella density (and hence foam properties and propagation rates), and it is these processes which can also be modelled by nonequilibrium mass transfer rate expressions. Obviously, unstable emulsion flow problems should be handled similarly.

An average lamella is defined as a combination of water component and aqueous surfactant in a given ratio:

$$(H_2O)_w + b(S)_w = (L)_g ; \quad b \approx 10^{-5} \quad (D16.1)$$

where the stoichiometry coefficient  $b$  is small (describing the number of moles of surfactant to the number of moles of water in the average lamella). The exact value of " $b$ " depends on a number of factors including the assumed shape of the lamella, the ratio of the water molar volume to the specific surface area of the surfactant (if it is assumed that the surfactant molecules coat the surface of the lamella to stabilize it) etc. The important point here is that  $b$  is small but non zero which implies that the physical properties of the lamella are related to those of water, and that lamella require available surfactant for their formation.

Given the above ideas, the remaining (non trivial) task is to form appropriate rate expressions for lamella capture (if an equilibrium adsorption isotherm is not assumed), for lamella generation and lamella coalescence and to fit these to experiments devised to separate contribution factors. For example, the requirement of a critical gas velocity for foam generation has been noted, as well as the inhibiting effect of non-condensable gas on foam coalescence. It is further possible that the role of oil both as a foam preventer (inhibiting foam generation) and a defoamer (accelerating foam decay) can be incorporated into these rate expressions. Such work is ongoing and is primarily dependent on the ability of experiment to separate mechanisms. Some preliminary matching of experiments by Friedmann et al<sup>25</sup> has been conducted using these ideas.

While important in simulating features of laboratory foam experiments, it remains an open question if such a detailed mechanistic approach to foam simulation is necessary or appropriate at the field-scale.

## D.17 Oil Banking Theory

In this section it will be shown via two-phase flow theory why oil banking and plugging occur when hot water or steam is injected.

The fractional flow equation for oil in a one-dimensional problem is

$$\frac{dS_o}{dt} = -v_T \frac{df_o}{dx} \quad (D17.1)$$

where  $f$  denotes fractional flow and  $v_T$  is total fluid velocity. Here we will assume that  $v_T$  is constant and positive, that is, the flow direction is from left to right.

The fractional flow of oil is defined as

$$f_o = \frac{\frac{k_{ro}}{\mu_o}}{\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w}} \quad (D17.2)$$

The fractional flow gradient is

$$\frac{df_o}{dx} = \frac{\left(\frac{k_{rw}}{\mu_w}\right) \frac{d}{dx} \left(\frac{k_{ro}}{\mu_o}\right) - \left(\frac{k_{ro}}{\mu_o}\right) \frac{d}{dx} \left(\frac{k_{rw}}{\mu_w}\right)}{\left(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w}\right)^2} \quad (D17.3)$$

Here  $k_{ro}$  is a function of  $S_o$ ,  $\mu_o$  is a function of temperature  $T$ ,  $k_{rw}$  is a function of  $S_w$ , and  $\mu_w$  is constant

$$\frac{d}{dx} \left(\frac{k_{ro}}{\mu_o}\right) = \frac{1}{\mu_o} \frac{dk_{ro}}{dS_o} \frac{dS_o}{dx} - \frac{k_{ro}}{\mu_o} \frac{d \ln \mu_o}{dT} \frac{dT}{dx} \quad (D17.4)$$

$$\frac{d}{dx} \left(\frac{k_{rw}}{\mu_w}\right) = \frac{1}{\mu_w} \frac{dk_{rw}}{dS_w} \frac{dS_w}{dx} = -\frac{1}{\mu_w} \frac{dk_{rw}}{dS_w} \frac{dS_o}{dx} \text{ since } S_o + S_w = 1 \quad (D17.5)$$

Therefore, the fractional flow equation may be written as

$$\frac{dS_o}{dt} = -v_T \frac{df_o}{dx} = A \left[ B \frac{dS_o}{dx} + C \frac{dT}{dx} \right] \quad (D17.6)$$

where

$$A = -\frac{v_T}{\mu_w \mu_o} \left[ \frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w} \right]^{-2} \quad (D17.7)$$

$$B = k_{rw} \frac{dk_{ro}}{dS_o} + k_{ro} \frac{dk_{rw}}{dS_w}, \text{ and} \quad (D17.8)$$

$$C = -k_{ro} k_{rw} \frac{d}{dT} (\ln \mu_o) \quad (D17.9)$$

Consider the case of isothermal Buckley-Leverett flow in an initially uniform reservoir. If water is injected at  $x = 0$ , then  $dS_o/dx$  will always be nonnegative. Since the factor B is always positive, and the factor A is always negative, we can conclude from D17.6 with  $dT/dx = 0$  that  $dS_o/dt$  is always non-positive. This means that the oil saturation at any one point will never increase; in other words, no oil bank can form.

In a hot water flood the region at the front of the hot water zone has a sharp temperature gradient where  $dT/dx$  is negative. Also, the factor C is always nonnegative, and for heavy oils at low temperatures it can have a very large magnitude. Therefore, in this region the negative temperature term in D17.6 completely dominates the positive saturation term, resulting in a positive value for  $dS_o/dt$ . This means that the oil saturation will increase, forming a bank.

However, as the temperature behind this region levels off,  $dT/dx$  becomes small and the sign of  $dS_o/dt$  given by D17.6 reverses. This means that the oil saturation decreases from its maximum value in the bank to some low value near the injector. The maximum oil saturation reached in the bank depends mostly on the C factor in D17.6, and is usually highest for heavy oils.

Under conditions experienced by some reservoirs, the oil bank attempts to displace all the mobile water, thereby creating a plug. This is observed as a sudden drop in injectivity. Usually the only recourse is some form of fracturing.

## References

1. Prats, M., Thermal Recovery, SPE Monograph, Vol. 7, SPE of AIME, Dallas, Texas, 1982.
2. Reid, R.C., Prausnitz, J.M., and Sherwood, T.K., The Properties of Gases and Liquids, Third Edition, McGraw-Hill, 1977.
3. Perry, R.H., and Chilton, C.H., Chemical Engineer's Handbook 3-226 to 3-250, McGraw-Hill, New York, 1973.
4. Aziz, K., Ramesh, B., and Woo, P.T., "Fourth SPE Comparative Solution Project: A Comparison of Steam Injection Simulators," JPT, December 1987, pp. 1576.
5. Buchanan, L., and Raicar, M., "Sensitivity Study of Field Scale Combustion Simulation for a Lloydminster-type Heavy Oil Reservoir," paper CIM 83-34-47, presented at the 34th Annual CIM Technical Meeting, Banff, Alberta, May 1983.
6. CMGPROP User's Manual, Computer Modelling Group.
7. Sandler, S.I., Chemical and Engineering Thermodynamics, John Wiley and Sons, 1977.
8. van Quy, N., Simandoux, P., and Corteville, J., "A Numerical Study of Diphasic Multicomponent Flow," SPEJ, April 1972, pp. 17.
9. Young, L.C., Stephenson, R.E., "A Generalized Compositional Approach for Reservoir Simulation," SPEJ, October 1983, pp. 727.
10. Hutchinson, C.A., and Braun, P.H., "Phase Relations of Miscible Displacement in Oil Recovery," AIChE J., March 1961, pp. 64.
11. Pope, G.A., and Nelson, R.C., "A Chemical Flooding Compositional Simulator," SPEJ, October 1978, pp. 339.

12. Redlich, O, and Kwong, J.N.S., Chem. Rev., Vol. 44, 1949, pp. 233.
13. Aziz, K., and Settari, A., Petroleum Reservoir Simulation, Applied Science Publishers, 1979.
14. Kular, G., Lowe, K., and Coombe, D.A., "Foam Application in an Oil Sands Steam Flood Process," paper SPE 19690, presented at the 64th Annual Technical Conference of SPE, San Antonio, Texas, October 8-11, 1989.
15. Coombe, D.A., Oballa, V., and Buchanan, W.L., "Foam Modelling Based on Lamella as a Dispersed Component," CMG Technical Report 90.08.T, April 1990.
16. Anand, J., Somerton, W.H., and Gomaa, E., "Predicting Thermal Conductivities of formations from Other Known Properties," SPEJ, Vol. 13, No. 5, October 1973, pp. 267-273.
17. Somerton, W.H., Keesee, J.A., and Chu, S.L., "Thermal Behavior of Unconsolidated Oil Sands," SPEJ, Vol. 14, No. 5, October 1974, pp. 513-521.
18. Vinsome, P.K.W., and Westerveld, J., "A Simple Method for Predicting Cap and Base Rock Heat Losses in Thermal Reservoir Simulators," JCPT, July-September 1980, pp. 87-90.
19. Burger, J.G., and Sahuquet, B.C., "Chemical Aspects of In-Situ Combustion - Heat of Combustion and Kinetics," SPEJ, October 1972, pp. 410.
20. Coats, K.H., "Some Observations on Field Scale Simulations of the In-Situ Combustion Process," paper SPE 12247, presented at the 1983 SPE Reservoir Simulation Symposium, San Francisco, California, November 1983.
21. Ito, Y., and Chow, A.K.-Y., "A Field Scale In-Situ Combustion Simulator With Channeling Considerations," SPERE, May 1988, pp. 419.
22. Schmidt, D.P., Soo, H., and Radke, C.J., "Linear Oil Displacement by the Emulsion Entrapment Process," SPEJ, June 1984, pp. 351.
23. Vittoratos, E., "Flow Regimes During Cyclic Steam Stimulation at Cold Lake," paper CIM/SPE 90-107, presented at the CIM/SPE International Technical Meeting, Calgary, Alberta, June 10, 1990.
24. Coombe, D.A., Oballa, V., and Buchanan, W.L., "Foam Modelling Based on Lamella as Dispersed Component," CMG Technical Report 90.08.T, April 1990.
25. Friedmann, F., Chen, W.M., and Gauglitz, P.A., "Experimental and Simulation Study of High Temperature Foam Displacement in Porous Media," paper SPE/DOE 17357, presented at the Enhanced Oil Recovery Symposium, Tulsa, Oklahoma, April 1988.

---

## D.18 Converting Black-Oil PVT to STARS

In general, the data requirements of a reservoir simulator will depend largely on the recovery mechanisms that are important. Thermal recovery mechanisms ranging from hot water to combustion are diverse and fundamentally different from black-oil processes. There are two key differences in the data required to describe these processes. The first, of course, is due to the accounting of energy and the effect of temperature on the process parameters. The second is that the most important fluid properties, density, viscosity and phase equilibrium, are modelled in completely different ways.

This appendix section deals with converting the IMEX black-oil PVT data  $R_s$ ,  $B_o$  and  $\mu_o$  into their STARS equivalents. Provided below are some detailed descriptions of the conversions along with some relevant theory and an example.

The most crucial step in the conversion procedure is testing your converted data. The best way to test is to use both simulators to solve the same problem, and compare the results. A small simple problem usually is sufficient, as long as it exercises the PVT data.

### CHOOSING COMPONENTS

A typical black-oil problem involves three components: water, dead oil and solution gas. These will be the components used in STARS, numbered in this order: water #1, dead oil #2 and solution gas #3. Water is found in the water and gas phases; dead oil is found only in the oil phase; and the solution gas is found in the oil phase and as free gas. It is important to distinguish between components and phases since the labels used are very similar. Here, the hydrocarbon components are called dead oil and solution gas, and the phases are called oil and gas.

In the following, units are mentioned in both field and SI systems.

### CONVERTING DENSITY DATA

A black-oil model refers to an amount of material in terms of standard volume, whereas STARS deals in moles. Therefore,  $B_o$  must be converted to the mole density form. We will need values for

- $\rho_o^{ST}$  - density (lb/bbl or kg/m<sup>3</sup>) of oil at standard T and p, and  $R_s = 0$  (IMEX's DENOS),
- $\rho_g^{ST}$  - density (lb/bbl or kg/m<sup>3</sup>) of free gas at standard T and p (IMEX's DENGS),
- $C_o$  - oil compressibility (1/psia or 1/kPa) (IMEX's CO),
- $B_o$  - formation volume factor (rb/stb or rm<sup>3</sup>/sm<sup>3</sup>), function of  $p_b$ ,
- $R_s$  - solution gas-oil ratio (scf/stb or sm<sup>3</sup>/sm<sup>3</sup>), function of  $p_b$ ,
- $p_b$  - saturation or bubble-point pressure (psia or kPa),
- $p_r$  - reference pressure (psia or kPa), where  $R_s(p_r) = 0$ ,
- $T_r$  - reference temperature (°F or °C),
- $M_2$  - molecular weight of dead oil component, and
- $M_3$  - molecular weight of solution gas component

Molecular weights are somewhat arbitrary since the dead oil and solution gas are really pseudo components, that is, fictional chemicals or multiple chemicals grouped together.  $M_3$  will be 16 lb/lb mol for 100% methane, and higher if some C<sub>2</sub>, C<sub>3</sub>, etc. are present.  $M_2$  is in the vicinity of 100 lb/lb mol for light oils and up to 500 lb/lb mol for heavier oils.

In a unit volume of live oil at pressure  $p$  and bubble-point pressure  $p_b$  the moles of dead oil and solution gas are

$$\frac{\rho_o^{ST}}{M_2 B_o(p, p_b)} \text{ and } \frac{R_s(p_b) \rho_g^{ST}}{M_3 B_o(p, p_b)} \quad (1)$$

where  $B_o(p, p_b) = B_o(p_b) * [1 - C_o(p - p_b)]$ . This is the result of converting standard volumes to mass, and then to moles. Therefore, the oil mole fractions are

$$x_3 = \frac{R_s(p_b) \rho_g^{ST} / M_3}{\rho_o^{ST} / M_2 + R_s(p_b) \rho_g^{ST} / M_3} \quad (2)$$

and  $x_2 = 1 - x_3$ . Note that  $x_3$  does not depend on  $B_o$ . Finally, the mole density of the live oil is

$$\rho_o(p, p_b) = \frac{\rho_o^{ST} / M_2 + R_s(p_b) \rho_g^{ST} / M_3}{B_o(p_b) * [1 - C_o(p - p_b)]} \quad (3)$$

The STARS expression for oil phase mole density is

$$\frac{1}{\rho_o(p, T, x_i)} = \sum_{i=2}^3 x_i \frac{\exp[C_{Ti}(T - T_r) - C_{oi}(p - p_r)]}{\rho_{oi}^{ST}} \quad (4)$$

where

$\rho_{oi}^{ST}$  - mole density of pure component I in the oil (liquid) phase, at  $P_r$  and  $T_r$ ,

$C_{oi}$  - oil phase compressibility of component i,

$C_{Ti}$  - oil phase thermal expansion coefficient of component i.

Equation (4) shows that it is the partial molar volume (i.e., the inverse of mole density) that is mixed linearly with oil mole fraction. This reflects ideal mixing. Also note how the pressure and temperature effects factor in.

Density conversion from IMEX to STARS is achieved by equating  $\rho_o(p, p_b)$  to  $\rho_o(p, T, x_i)$ .

The value of  $\rho_{o2}^{ST}$  for dead oil corresponds to the reference density  $\rho_o^{ST}$  at reference pressure and temperature and  $R_s = 0$ , that is, where  $B_o = 1$ :

$$\rho_{o2}^{ST} = \rho_o^{ST} / M_2. \quad (5)$$

The first entry in a black-oil PVT table usually corresponds to reference pressure, but at reservoir temperature  $T^*$ . Therefore, a value of  $B_o > 1$  at  $p_b = p_r$  must correspond to thermal expansion, since  $R_s(p_r) = 0$ . (Nonlinear mixing is beyond the scope of this discussion).

Substituting  $p = p_b = p_r$ ,  $T = T^*$  and  $R_s = 0$  into (2), (3) and (4) gives

$$B_o(p_r) = \exp[C_{T2}(T^* - T_r)] = B_o^*. \quad (6)$$

This can be used to calculate  $c_{T2}$ , if appropriate. Assume that  $c_{T3} = c_{T2}$ , so that  $B_o^*$  may appear as a constant factor. Assume also that  $c_{o2} = c_{o3} = C_o$ .

The only unknown left in (4) is  $\rho_{o3}^{ST}$ . Equate  $\rho_o(p, p_b)$  and  $\rho_o(p, T^*, x_i)$ , and substitute the expressions from (2) for  $x_2$  and  $x_3$ :

$$\rho_{o3}^{ST} = \frac{R_s(p_b) \rho_g^{ST} / M_3}{\left[ \frac{B_o(p_b)/B_o^*}{\exp[c_{o2}(p_b - p_r)]} - 1 \right]} \quad (7)$$

The physical meaning of  $\rho_{o3}^{ST}$  is clear; it is the mole density at  $p_r$  and  $T_r$  of solution gas dissolved in oil. This is not to be confused with the density of pure liquid solution gas, which usually does not exist at standard conditions. Neither should it be confused with the gas phase density, or the bulk density of solution gas in the live oil. Even though  $\rho_{o3}^{ST}$  cannot be measured directly, its value can be calculated.

A black-oil PVT table is capable of giving a value of  $\rho_{o3}^{ST}$  for each  $p_b$  entry; hence,  $\rho_{o3}^{ST}$  is not unique. In fact, values of  $\rho_{o3}^{ST}$  from lower  $p_b$  represent lighter parts of the solution gas pseudo component; values from higher  $p_b$  correspond to higher carbon numbers. Since only one value of  $\rho_{o3}^{ST}$  may be used in STARS, it is possible that STARS will match only one PVT table entry exactly, and the density at all other  $p_b$  will be approximate. You can pick an average value, or the value corresponding to the initial  $p_b$  in the reservoir to get the correct initial mass in place.

It is interesting to note that  $\rho_{o3}^{ST}$  from equation (7) does not depend upon dead oil density  $\rho_o^{ST}$  or molecular weight  $M_2$ . In fact, the variability of  $\rho_{o3}^{ST}$  at different values of  $p_b$  will not depend upon  $M_3$ , since  $\rho_{o3}^{ST}$  is proportioned to the constant  $1/M_3$ .

## CONVERTING VISCOSITY DATA

Part of the PVT table in a black-oil model is oil phase viscosity versus bubble point pressure  $\mu_o(p_b)$ . The STARS mixing rule for liquid viscosity is

$$\ln[\mu_o(x_i)] = \sum_{i=2}^3 x_i \ln[\mu_{oi}] \quad (8)$$

where

- $\mu_{o2}$  - viscosity of pure liquid dead oil component, and
- $\mu_{o3}$  - viscosity of pure liquid solution gas component.

The object is to determine  $\mu_{o2}$  and  $\mu_{o3}$ . At the reference pressure  $p_b$  entry, that is,  $p = p_b = p_r$ , we have  $R_s(p_r) = 0$ , so  $x_3 = 0$  and  $x_2 = 1$ . This gives

$$\mu_{o2} = \mu_o(p_r). \quad (9)$$

At a higher  $p_b$  where  $R_s(p_b) > 0$ , we get

$$\ln[\mu_{o3}] = (1/x_3) \bullet (\ln[\mu_o(p_b)] - x_2 \bullet \ln[\mu_{o2}]) \quad (10)$$

As with the calculation of liquid density  $\rho_{o3}^{ST}$ , you will have to choose one value of  $\mu_{o3}$  from the values produced by each  $p_b$  entry.

## PHASE EQUILIBRIUM DATA

To model phase equilibrium STARS uses K values, which are the ratios between gas mole fraction and liquid mole fraction:

$$K_i = y_i / x_i \text{ or } y_i = K_i x_i . \quad (11)$$

At saturated conditions  $p = p_b$ , the gas phase exists and the gas mole fraction constraint

$$y_1 + y_2 + y_3 = 1 \quad (12)$$

must be satisfied. The water and dead oil components do not vaporize significantly, so  $y_1 = y_2 = 0$  since  $K_1 = K_2 = 0$ , leaving  $y_3 = K_3 x_3 = 1$ . So  $K_3(p_b) = 1/x_3$  and the solution gas K value as a function of pressure p is

$$K_3(p) = 1 + \frac{\rho_o^{ST} / M_2}{R_s(p) \rho_g^{ST} / M_3} . \quad (13)$$

STARS has both correlation and table options for entering K value data. Equally spaced table entries can be used directly. The correlation form

$$K(p, T) = \left( \frac{a_1}{p} + a_2 + a_3 p \right) \exp[-a_4/(T - a_5)] \quad (14)$$

may be useful, especially if  $R_s(p)$  is nearly directly proportioned to p. Temperature dependence can be estimated by adapting values of  $a_4$  and  $a_5$  from other similar components (see Table 2 in the STARS User's Guide).

STARS will print out the gas mole fraction for each grid block which contains a gas phase. For blocks without gas phase (undersaturated or  $p > p_b$ ) it reports the potential value of  $y_3$  which is the product

$$K_3(p) x_3 = \frac{\left[ 1 + \frac{\rho_o^{ST} / M_2}{R_s(p) \rho_g^{ST} / M_3} \right]}{\left[ 1 + \frac{\rho_o^{ST} / M_2}{R_s(p_b) \rho_g^{ST} / M_3} \right]} < 1 \quad (15)$$

since  $R_s(p) > R_s(p_b)$ . This indicates how close an undersaturated block is to its bubble point.

### EXAMPLE

Consider the black-oil PVT table D.7. Entries 2 to 10 were taken from an IMEX data set. The first entry is obtained by linearly extrapolating entries 2 and 3 to a pressure  $p_r$  for which  $R_s(p_r) = 0$ . Also obtained from the IMEX data were  $\rho_o^{ST} = 56.7 \text{ lb/ft}^3$ ,  $\rho_g^{ST} = 0.05616 \text{ lb/ft}^3$  and  $C_o = 1.147 \bullet 10^{-5} \text{ 1/psia}$ .

Entry	$p_b$ (psia)	$B_o$ (rb/stb)	$R_s$ (scf/stb)	$\mu_o$ (cp)
1	1.8	1.046	0.0	4.086
2	15.0	1.050	5.0	3.934
3	115.0	1.080	43.0	2.776
4	615.0	1.126	160.0	2.053
5	1215.0	1.170	265.0	1.600
6	2015.0	1.223	403.0	1.196
7	3015.0	1.293	583.0	0.880
8	4015.0	1.372	782.0	0.725
9	5015.0	1.449	983.0	0.638
10	6015.0	1.526	1183.0	0.572

Table D.7: Black-oil type PVT table

A dead oil molecular weight of  $M_2 = 444$  lb/lb mol was chosen for an oil about this density. Assuming an ideal gas mole density of 0.00262 lb mol/ft<sup>3</sup>, corresponding to 1 atm and 63.3°F, the solution gas molecular weight is estimated as

$$\rho_g^{ST} / (0.00262 \text{ lb mol/ft}^3) = 21.44 \text{ lb/lb mol.}$$

The dead oil standard mole density is  $\rho_o^{ST} / (444 \text{ lb/lb mol}) = 0.1277 \text{ lb mol/ft}^3$ . The base value of  $B_o$  is  $B_o^* = 1.046 \text{ rb/stb}$ , and the oil compressibility  $c_{o2} = C_o = 1.147 \cdot 10^{-5} \text{ 1/psia}$ . The dead oil viscosity is  $\mu_{o2} = 4.086 \text{ cp}$ .

Entry	$x_3$	$\rho_{o3}^{ST}$ (lb mol/ft <sup>3</sup> )	$K_3$ (p)	$\mu_{o3}$ (cp)
1	0.0000			
2	0.0179	0.6176	55.74	0.491
3	0.1358	0.5963	7.366	0.237
4	0.3689	0.8898	2.711	0.632
5	0.4919	0.9219	2.033	0.607
6	0.5955	0.9563	1.679	0.519
7	0.6805	0.9709	1.470	0.428
8	0.7407	0.9737	1.350	0.396
9	0.7822	0.9768	1.278	0.380
10	0.8121	0.9740	1.231	0.363

Table D.8: Analysis of PVT data for STARS conversion

Table D.8 shows the result of applying equations (2), (7), (10) and (13) to the live-oil entries of Table D.7. All the K values  $K_3(p)$  will be needed, since this table shows the pressure dependence of  $K_3$  at reservoir temperature  $T^*$ . However, only one value of each of  $\rho_{o3}^{ST}$  and  $\mu_{o3}$  must be chosen. You could choose the values corresponding to entry 7, since this seems representative of the upper, stabilized portion of the pressure range. On the other hand, you could choose rounded-off representative values, for example, 0.95 lb mol/ft<sup>3</sup> and 0.4 cp. If it is a priority for STARS to match closely the initial amounts of dead oil and solution gas in place, then you would chose the entry closest to the initial  $p_b$ .

## D.19 Other Aquifer Models

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, more recent method 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.

### 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

$\mu$  = 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 D.9 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).

$$P(t_d) = \frac{2}{\pi} t_d^{1/2}, t_d < 0.01 \quad (4)$$

$$P(t_d) = \frac{1}{2} [\log_{10} t_d + 0.80907], t_d > 1000 \quad (5)$$

Table D.9 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,

$$P(t_d) = \frac{2}{(R_d^2 - 1)} \left[ \frac{1}{4} + t_d \right] - \frac{(3R_d^4 - 4R_d^4 \log_{10} R_d - 2R_d^2 - 1)}{4(R_d^2 - 1)^2} + 2 \sum_{\beta_1, \beta_2}^{\infty} \frac{e^{-\beta^2 t_d} J_1^2(\beta_n R_d)}{\beta_n^2 [J_1^2(\beta_n R_d) - J_1^2(\beta_n)]} \quad (6)$$

Where the  $B_1, B_2$  are the roots of the following equation

$$[J_1(\beta_n R_d) Y_0(\beta_n R_d) - Y_1(\beta_n R_d) J_0(\beta_n R_d)] = 0 \quad (7)$$

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  $t_d$ .

$$P(t_d) = \frac{2}{(R_d^2 - 1)} \left[ \frac{1}{4} + t_d \right] - \frac{(3R_d^4 - 4R_d^4 \log_{10} R_d - 2R_d^2 - 1)}{4(R_d^2 - 1)^2} \quad (8)$$

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.

$$W_e(t_{dj}) = W_e(t_{dj-1}) + \left[ \frac{UU * \Delta P(t_{dj}) - W_e(t_{dj-1}) * P'(t_{dj})}{P(t_{dj}) - t_{dj-1} P'(t_{dj})} \right] (t_{dj} - t_{dj-1}) \quad (9)$$

$$UU = \text{Aquifer constant} = 1.119 * \phi * C_e * R_o^2 * h * \left[ \frac{\theta}{360} \right] \quad (10)$$

$$\Delta P(t_{dj}) = \text{Change in boundary pressure} = P_{res}(t_j) - P_{res}(t_{j-1})$$

$P(t_{dj})$  = 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

$\theta$  = 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 needs neither tables nor 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 constitutes 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

$$[\bar{P}_{an-1}] = P_i \left[ 1.0 - \frac{\sum_{j=1}^{j=n-1} \Delta W_{ej}}{W_{ei}} \right] = \text{Average aquifer pressure} \quad (12)$$

$$[\bar{P}_n] = \frac{P_{n-1} + P_n}{2} = \text{Interface pressure between aquifer/reservoir} \quad (13)$$

$$J_w = \frac{7.08 * 10^{-3} kh}{\mu \left[ \ln \left( \frac{R_e}{R_o} \right) - \frac{3}{4} \right]} = \text{Aquifer productivity index} \quad (14)$$

$$W_{ei} = C_e * f * \pi * (R_e^2 - R_o^2) * h\phi * P_i = \text{Initial amount of encroachable water} \quad (15)$$

$\Delta W_{en}$  is the incremental water influx in the interval  $\Delta t_n$ .

### Selecting Type of Aquifer

A user is able to select the type of aquifers to be connected to a specific reservoir. Those types are BOUNDARY, 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.

### 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

Both the Carter-Tracy and Fetkovich solutions use the same aquifer parameters:

- 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 defaults:

Parameter	Boundary	Region	Bottom
POR_AQ	0.3	0.3	0.3
PERM_AQ	100	100	100
AREA	1056000 ft <sup>2</sup>	AREA-R (depends on defined region)	1210000 ft <sup>2</sup>
H_AQ	Nz*Dz=240'	Nz*Dz=240'	$\sqrt{(1210000)} = 1100'$
R <sub>o</sub>	$1056000/(240*2\pi) = 700.28'$	$\sqrt{(AREA/\pi)}$	$\sqrt{(1100 * 240/\pi)} = 289.88'$
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 convert this data to the required parameters above. This could be furnished as follows:

Assume

W<sub>i</sub> = Volume of water available for Injection. (STB)

J<sub>w</sub> = 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:

$$R_o = \frac{\text{AREA} - \text{Contact}}{H_AQ * 2\pi}$$

2. Use the W<sub>i</sub> (STB) in the equation below to determine the external reservoir radius (R<sub>e</sub> - ft) – use contact angle θ as 1.0, and use any reasonable value for the aquifer porosity.

$$W_i = \theta * \frac{\pi[R_e^2 - R_o^2]}{5.615} * H_AQ * \text{POR\_AQ}$$

W<sub>i</sub> (ft<sup>3</sup>) is reported in the simulation output, which can be used either for checking or for computing R<sub>e</sub>.

3. Use the Aquifer Injectivity Index (STB/D/psi) to determine the effective aquifer permeability (md), as follows:

$$J_w = \frac{7.08 * 10^{-3} * \theta * \text{PERM\_AQ} * H\_AQ}{\mu_w * \ln[R_e / R_o]}$$

The  $\mu_w$  is the water or (aquifer fluid viscosity) in cp.

For further description of the above subject, the reader could consult these references:

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 D.9 - Radial Flow, Constant Terminal Pressure  
and Constant Terminal Rate Cases for Infinite Reservoirs**

The default \*AQFUNC pressure influence function is tabulated here.

<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>
0.01	0.112	30.00	2.147
0.05	0.229	40.00	2.282
0.10	0.315	50.00	2.388
0.15	0.376	60.00	2.476
0.20	0.424	70.00	2.550
0.30	0.503	80.00	2.615
0.50	0.616	90.00	2.672
0.70	0.702	100.00	2.723
1.00	0.802	200.00	3.064
1.50	0.927	300.00	3.263
2.00	1.020	400.00	3.406
3.00	1.169	500.00	3.516
5.00	1.362	600.00	3.608
7.00	1.500	700.00	3.684
10.00	1.651	800.00	3.750
15.00	1.829	900.00	3.809
20.00	1.960	1000.00	3.860

**Table D.10 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs**

<b>R1 = 1.5</b>		<b>R = 2.0</b>		<b>R = 2.5</b>		<b>R = 3.0</b>	
<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>
0.06	0.251	0.22	0.443	0.40	0.565	0.52	0.627
0.08	0.288	0.24	0.459	0.42	0.576	0.54	0.636
0.1	0.322	0.26	0.476	0.44	0.587	0.56	0.645
0.12	0.355	0.28	0.492	0.46	0.598	0.60	0.662
0.14	0.387	0.30	0.507	0.48	0.608	0.65	0.683
0.16	0.420	0.32	0.522	0.50	0.618	0.70	0.703
0.18	0.452	0.34	0.536	0.52	0.628	0.75	0.721
0.20	0.484	0.36	0.551	0.54	0.638	0.80	0.740
0.22	0.516	0.38	0.565	0.56	0.647	0.85	0.758
0.24	0.548	0.40	0.579	0.58	0.657	0.90	0.776
0.26	0.580	0.42	0.593	0.60	0.666	0.95	0.791
0.28	0.612	0.44	0.607	0.65	0.688	1.0	0.806
0.30	0.644	0.46	0.621	0.70	0.710	1.2	0.865
0.35	0.724	0.48	0.634	0.75	0.731	1.4	0.920
0.40	0.804	0.50	0.648	0.80	0.752	1.6	0.973
0.45	0.884	0.60	0.715	0.85	0.772	2.0	1.076
0.50	0.964	0.70	0.782	0.90	0.792	3.0	1.228
0.55	1.044	0.80	0.849	0.95	0.812	4.0	1.578
0.60	1.124	0.90	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 D.10 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs  
(Continued)**

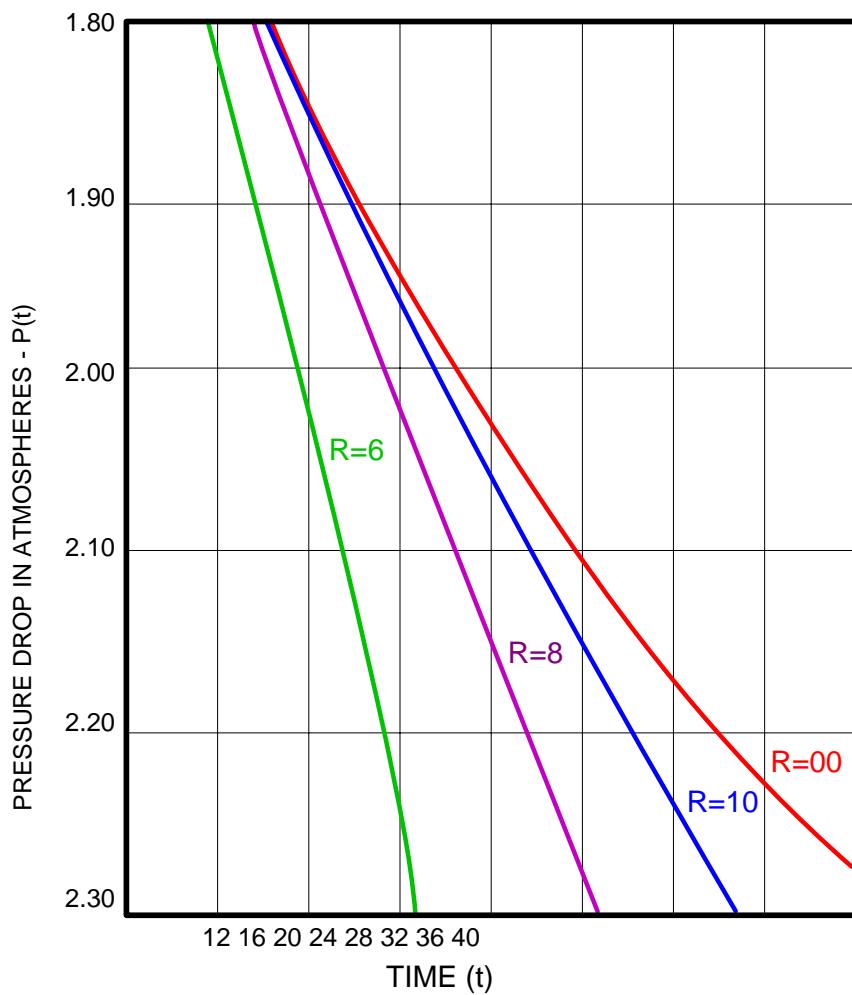
<b>R = 3.5</b>		<b>R = 4.0</b>		<b>R = 4.5</b>	
<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>
1.0	0.802	1.5	0.927	2.0	1.023
1.1	0.830	1.6	0.948	2.1	1.040
1.2	0.857	1.7	0.968	2.2	1.056
1.3	0.882	1.8	0.988	2.3	1.072
1.4	0.906	1.9	1.007	2.4	1.087
1.5	0.929	2.0	1.025	2.5	1.102
1.6	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 D.10 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs  
(Continued)**

<b>R = 5.0</b>		<b>R = 6.0</b>		<b>R = 7.0</b>	
<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>
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 D.10 – Constant Terminal Rate Case Radial Flow – Limited Reservoirs  
(Continued)**

<b>R = 8.0</b>		<b>R = 9.0</b>		<b>R = 10</b>	
<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>	<b>td</b>	<b>P(td)</b>
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 D.20**  
**Radial Flow, Constant Terminal Rate Case,**  
**Pressure Drop Versus Time,  $P(t_d)$  Versus  $t_d$**

---

## D.20 Velocity-Dependent Viscosity

For Newtonian fluid flow, viscosity depends upon only individual component viscosities (\*AVISC, etc., for liquid phase, \*AVG, etc., for gas phase) as well as a mixing rule (default linear or \*VSMIXFUNC, etc., for non-linear). For a more detailed discussion on how STARS models viscosity for Newtonian fluid flow, see the manual page for these keywords as well as Appendix D.5. For Newtonian fluid flow, viscosity is independent of velocity.

Laboratory and field tests suggest that correct modelling of the velocity-dependent properties of the viscosity of fluid phases containing polymer helps improve polymer-flooding efficiency (see references at the end of this section). STARS has a method for modelling velocity-dependent viscosity and reporting several associated parameters such as phase velocity magnitudes, shear rates and shear stresses.

For modeling velocity dependent viscosities, STARS provides the following options:

1. A velocity-dependent shear thinning model based on a power law relation between fluid viscosity and Darcy velocity. This model is based on a user-defined power index of value less than 1 and a lower velocity at which the fluid phase containing polymer approaches its Newtonian value. Using the power law relation, STARS calculates an upper velocity to ensure that fluid viscosity approaches its lowest possible value defined by the same fluid in the absence of polymer. Newtonian values for the fluid viscosity are determined using the standard viscosity data and calculations.
2. A velocity-dependent shear thickening model based on a power law relation between fluid viscosity and Darcy velocity. This model is based on a user-defined power index of value greater than 1, a maximum viscosity and a maximum velocity at which the fluid phase viscosity reaches its maximum value. Using the power law relation, STARS calculates a lower velocity corresponding to the same fluid undergoing Newtonian flow. Newtonian values for the fluid viscosity are determined using the standard viscosity data and calculations.
3. A velocity-dependent combined shear thinning and thickening model based on adding the effects of the two above shear thinning and thickening power law relations. This relation is bounded by two plateaus; one plateau ensures a Newtonian fluid viscosity for lower velocities, and one plateau ensures a maximum viscosity limit for higher velocities.
4. A velocity dependent viscosity based on tabular input. This method provides greater flexibility in defining specific data or representing complicated functions of fluid viscosity versus Darcy velocity.

### Velocity Dependent Skin Factor

In many cases, much of the shear thinning or thickening occurs within the injection well grid block. Although non-Newtonian flow between grid blocks is accounted for, modelling of non-Newtonian flow within the well grid block requires a user-input skin factor. The following is a discussion of the standard radial inflow well model based on Newtonian flow and a derivation of the recommended skin factor that can be used to correct for actual non-Newtonian flow within the well grid block.

For model input protocol and a detailed discussion of well skin factors, see “Well Element Geometry (Conditional)” of section “Well and Recurrent Data” and “Radial Inflow Well Model” of section “Appendix A: Well Model Details”.

The Radial Inflow Well Model is based on a steady-state Newtonian flow which incorporates a dimensionless pressure-drop skin factor. This equation uses an equivalent well block radius  $r_e$  defined as the radius at which the steady-state flowing pressure of the actual well is equal to the numerically calculated pressure for the well block.

The pressure drop between the well bottom-hole pressure and the well block pressure  $P_k$  can be represented by rewriting equations A1.1a and A1.1b:

$$P_{wfk} - P_k = \left( \frac{\mu}{k} \right) \left( \frac{q_{jk}}{2\pi h} \right) \left[ \ln \left( \frac{r_e}{r_w} \right) + S \right], \quad (\text{D20.1})$$

where the well fraction and layer thickness factor have been ignored.

A similar equation can be written for a velocity dependent viscosity and non-Newtonian flow (Odeh<sup>3</sup>):

$$P_{wfk} - P_k = \left( \frac{\mu_{ref}}{ku_{ref}^{n-1}} \right) \left( \frac{q_{jk}}{2\pi h} \right)^n \left( \frac{1}{1-n} \right) [r_e^{1-n} - r_w^{1-n}], \quad (\text{D20.2})$$

where  $n$ ,  $\mu_{ref}$  and  $u_{ref}$  are the power index, reference viscosity and reference velocity. These parameters are obtained from the power law equations given manual pages for \*SHEARTHIN and \*SHEARTHICK.

The user must determine whether shear thinning or shear thickening is dominating in the well grid block and use the appropriate power index and reference parameters from the power law relations. For example, if thinning is the dominant (or only) velocity dependent viscosity effect, then  $n = n_{thin}$ ,  $u_{ref} = u_{l,lower}$ , and  $\mu_{ref} = \mu_{l,p}$ .

A skin factor to account for the pressure drop difference between Newtonian and non-Newtonian flow can be determined by equating equations D20.1 and D20.2:

$$S = \left( \frac{1}{1-n} \right) \left[ 1 - \left( \frac{r_w}{r_e} \right)^{1-n} \right] + \ln \left( \frac{r_w}{r_e} \right), \quad (\text{D20.3})$$

where

$$\mu = \mu_{ref} \left[ \frac{u_l}{u_{ref}} \right]^{n-1},$$

and

$$u_l = \frac{q_{jk}}{2\pi h r_e}.$$

This skin factor should provide the necessary correction to the Newtonian Radial Inflow Well Model to account for non-Newtonian flow within the well grid block.

Equation D20.3 can be solved by assuming that the equivalent well block radius for Newtonian and non-Newtonian flow is similar and given by equation A1.2. However, technically these values will be different and a more accurate determination of the well block radius for non-Newtonian flow can be done using the velocity-dependent power law for viscosity and a derivation similar to the one outlined in Peaceman<sup>5</sup>.

#### References:

1. Cannella, W., Huh, C., and Seright, R.S., "Prediction of Xanthan Rheology in Porous Media", paper SPE 18089, presented at the 63<sup>rd</sup> Annual Tech. Conference SPE, Houston, Texas, October 1988.
2. Delshad, M., Kim, D.H., Magbagbeola, O.A., Huh, C., Pope, G.A., and Tarahhom, F., "Mechanistic Interpretation and Utilization of Viscoelastic Behavior of Polymer Solutions for Improved Polymer-Flood Efficiency", paper SPE 113620, presented at the SPE/DOE Improved Oil Recovery Symposium, Tulsa, Oklahoma, April 2008.
3. Odeh, A.S., and Yang, H.T., "Flow of Non-Newtonian Power-Law Fluids Through Porous Media", SPEJ, June 1979, pp. 155-163.
4. Shahin, G.T., and Thigpen, D.R., "Injecting Polyacrylamide Into Gulf Coast Sands: The White Castle Polymer Injectivity Test", paper SPE/DOE 24119, presented at the SPE/DOE 8<sup>th</sup> Symposium on Enhanced Oil Recovery, Tulsa, Oklahoma, April 1992.
5. Teeuw, D., and Hesselink, T.H., "Power-Law Flow and Hydrodynamic Behaviour of Biopolymer Solutions in Porous Media", paper SPE 8982, presented at the SPE 5<sup>th</sup> International Symposium on Oilfield and Geothermal Chemistry, Stanford, California, May 1980.
6. 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.



# Appendix E: Grid Design

## Overview

This appendix contains detailed descriptions of advanced grid models used by STARS, divided into the following sections.

- E.1 Nonuniform Formation Properties
- E.2 Resolution of Process Phenomena
- E.3 Variable Depth and Thickness
- E.4 Grid Orientation
- E.5 Symmetry Elements
- E.6 Local Grid Refinement
- E.7 Hybrid Grid
- E.8 Naturally Fractured Reservoirs

## E.1 Nonuniform Formation Properties

Much to the frustration of the engineer attempting to conduct a simulation study, oil-bearing reservoirs are notoriously nonuniform. Examples of types of heterogeneities are: distinctive formation layers and zones, pay zones separated by impermeable rock, permeability grading up or down with height, high-permeability channels, micro-fractures, hydraulic (macro) fractures and shale lenses.

Heterogeneities are divided into two groups:

1. Phenomenon must be modelled as a distinct item, separate from adjacent item.  
Examples are high-permeability channels, separate pay zones, and grading permeability (if important to the process).
2. Phenomenon can have an effect by modifying other data, or by averaging.  
Examples are micro fractures and shale lenses, where the gross permeability may be changed to account for the effect.

The way to treat any one nonuniformity will depend upon its interaction with the process. For example, Figure E.1 shows the position of the combustion front in a reservoir cross-section in which the largest permeability is in the middle of the pay zone, with a maximum contrast of 4.4. Despite the large amounts of air and steam involved, the sweep pattern and breakthrough time are dominated by the permeability distribution. Thus the heterogeneous permeability must be modelled in this case. The adequate representation of reservoir nonuniformities usually is one of the highest priorities in grid design.

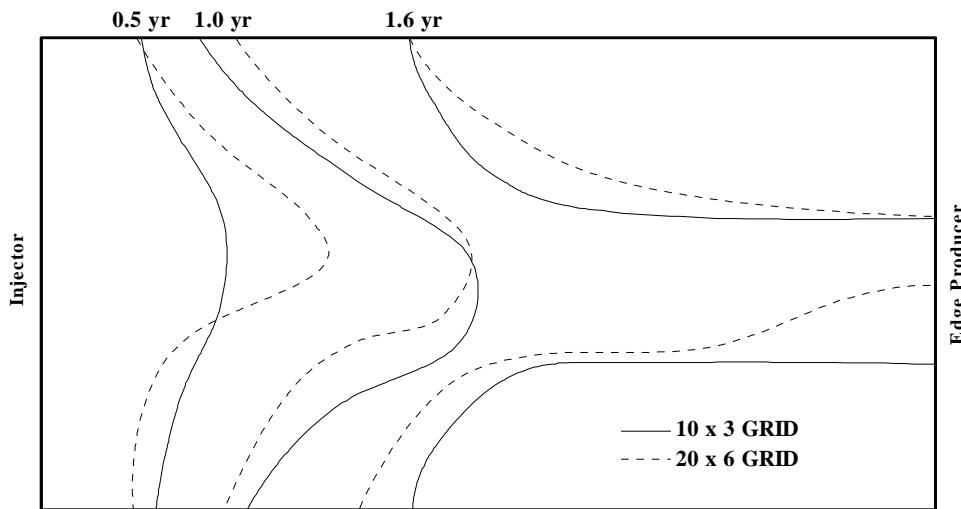


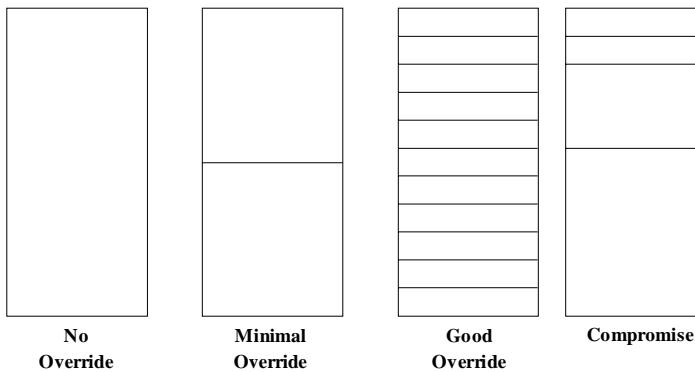
Figure E.1: Sensitivity of vertical sweep to cross-sectional grid size

## E.2 Resolution of Process Phenomena

Examples of process phenomena are fluid banks, fronts, and property gradients. Even in a uniform reservoir, numerical dispersion will introduce errors into the simulation results.

### Gravity Override

The tendency for gas to rise in the reservoir is very often a key process phenomenon in thermal EOR, and so it must be modelled adequately. To observe gas override, at least two vertical blocks are required. Many vertical blocks will model it best, but sometimes only a few are needed near the top for adequate representation of gas breakthrough (Figure E.2).



*Figure E.2: Capturing gas override*

A set of sensitivity runs is the best way to determine if the grid is adequate. The result of two such runs is shown in Figure E.1. The positions of the fire front shows some differences between the two grids, but for the purpose of studying well spacing effects the coarser (and cheaper) 10x3 grid was deemed accurate enough.

### Distinct Pay Zones

If the process is sufficiently influenced by differences in pay zone properties, then each pay zone should be modelled as at least one block layer. If detail in a zone is important, then several block layers are needed for that zone. On the other hand, it may be possible to lump several pay zones into one block layer if the process reacts similarly to each zone.

### Bottom Water

A bottom water zone may play several different roles, such as aquifer source, thief zone for mobilized oil, and a high-mobility channel for steam or air. A water zone may require several block layers if air or steam override will occur there, or if oil will plug the zone.

### Steam Fronts

Steam fronts are rather insensitive to horizontal block size, even though the front may be sharp. This is helped by the fact that the steam condensation front seems to be inherently stable, that is, fingers tend to shrink. Also, the piston-like nature of the process is quite independent of physical parameters, such as steam temperature and gas saturation. Only when other phenomena are important (additive contact area, near-well gradients, cycling) will the areal block need to be smaller.

## **Combustion Fronts**

The nature of the burning front in fire flooding makes it very different from steam flooding. A burning zone is typically only about 5 to 10 cm wide in a combustion tube, and perhaps larger in the field. The peak temperature in this small zone may be the key parameter in the entire process, but this temperature cannot be represented by the averaged block temperature. In field-scale combustion simulation the following constraints usually are accepted:

1. All reaction stoichiometry and enthalpies are assumed to be known.
2. Fuel lay down is assumed to be a known value, to be deposited when the steam zone arrives.
3. Burning rates are simplified to complete combustion when oxygen and fuel are contacted; alternatively, a burning front temperature is assumed.

## **Near-Well Phenomena**

In the vicinity of a wellbore the pressure gradients and flow rates are high. Large block sizes will introduce the assumption of uniform pressure gradient, saturations and temperature.

Better horizontal resolution is required if these gradients are important (see Section E.7 on Hybrid Grids). Vertical resolution is needed if coning is anticipated, or if gravity override will affect production capacity.

Cyclic steam simulations, with a single well, usually have blocks near the wellbore of about 1m, and the block sizes increase away from the wellbore.

## **Fractures and Channels**

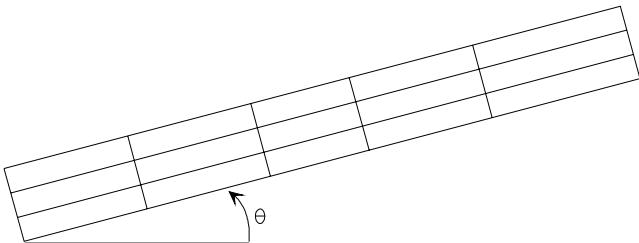
Fractures and channels can be modelled several different ways.

1. Use an entire row or layer of blocks which are assigned very large permeabilities and porosities. Transmissibilities can be modified at certain times during the run. This is the brute-force method, and is easy to control.
2. The effects of micro-fractures can be averaged into the block's gross permeability and porosity, and the transmissibilities of entire regions can be changed with time.
3. Fractures opening and closing can be modelled with a simple pressure-dependent transmissibility model. Transmissibility varies between a minimum value and a maximum value as a function of pressure.

---

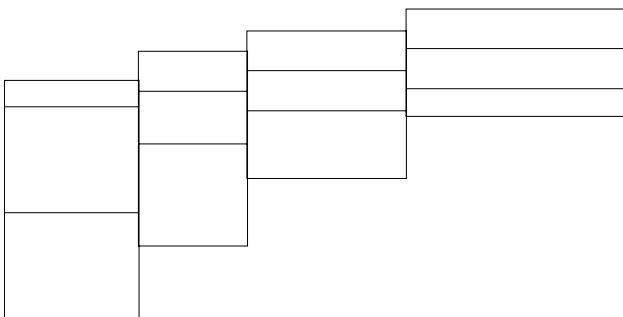
### E.3 Variable Depth and Thickness

A reservoir that is tilted at a constant angle is a candidate for specifying only the tile angle  $\theta$  (Figure E.3).



*Figure E.3: Tilted reservoir*

A reservoir that is bent and tilted, or has a pay zone whose thickness varies significantly, can be modelled with another device. Each column of blocks can be situated in space independent of other columns (Figure E.4). Care must be taken to ensure that the reservoir described is connected and realistic.



*Figure E.4: Tilted and thinning reservoir*

## E.4 Grid Orientation

Grid orientation occurs when the numerical result depends too much on which direction the grid lines run. Consider the example in figures E.5, E.6 and E.7. An areal combustion simulation was run, and the fire front positions at two times are shown. Breakthrough occurs first at the edge producer, which is then shut in. Injection continues until the fire front breaks through at the corner producer. The cylindrical and diagonal grids behave similarly, but the parallel grid result is completely incorrect (breakthrough occurs along the line between producers). All three cases were run using five-point areal discretization.

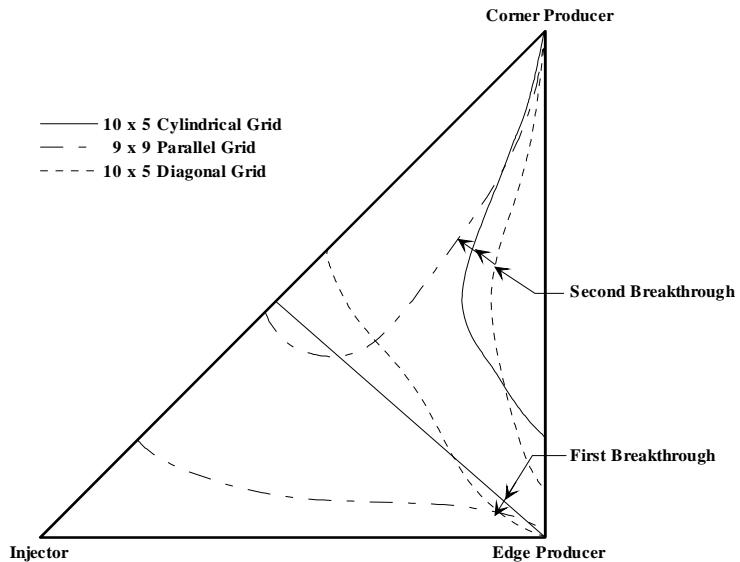


Figure E.5: Sensitivity of areal sweep to areal grid type

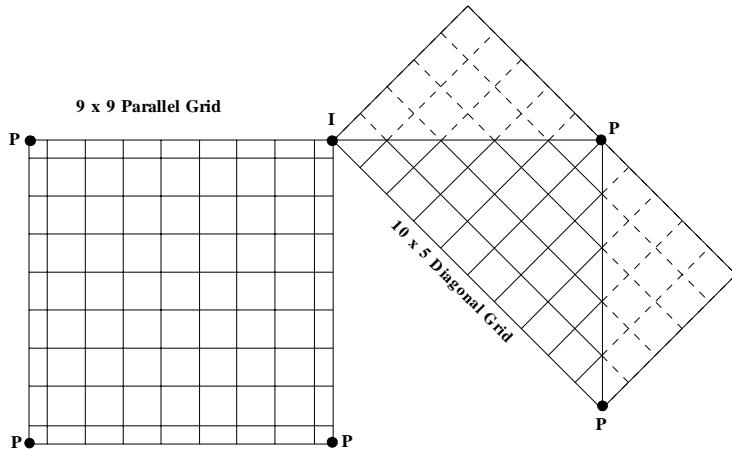
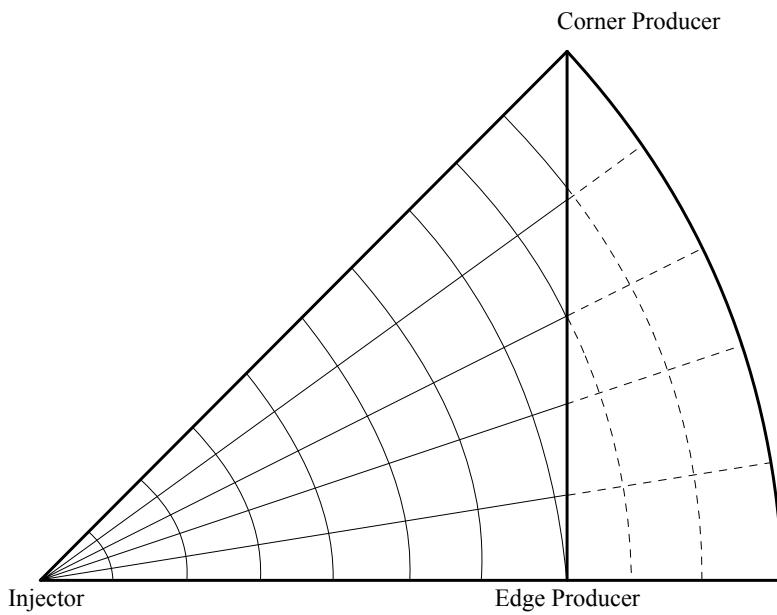


Figure E.6: Parallel and diagonal rectangular grids



*Figure E.7: Areal 10x5 cylindrical grid*

The availability of nine-point options greatly reduces the concern over grid orientation. However, since nine-point runs are usually about twice as expensive as five-point runs, test runs are called for to see if nine-point is necessary.

## E.5 Symmetry Elements

Symmetry elements are used frequently in thermal simulation for a number of reasons:

1. Compared with black-oil models, thermal models require much more CPU and storage per grid block. Therefore, less blocks can be used for a given computer storage limit.
2. Thermal EOR processes require more grid blocks per well or per pattern, since fronts are sharp and distinct.
3. Accuracy can be maximized for use in test and sensitivity runs.
4. Some results from one element may be generalized to other elements and patterns.
5. Pattern interference can be investigated by sensitivity runs with different injection share or production share.
6. Full-pattern or multipattern runs can be done once an acceptable coarse grid is obtained.

Figure E.8 shows how a symmetry element may be picked from a pattern. Each of the grids in Figure E.6 and E.7 are attempts to model the pattern element contained within the dotted line.

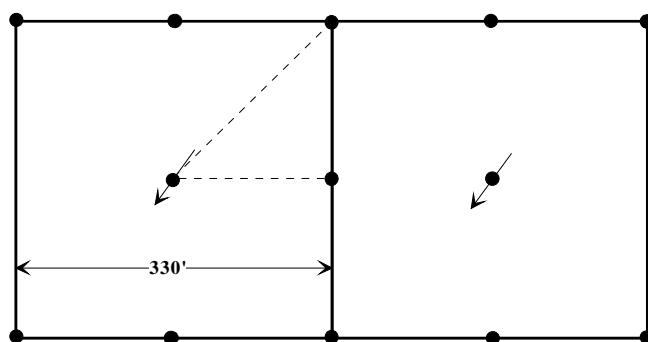


Figure E.8a: Two 2.5 acre inverted nine-spot patterns

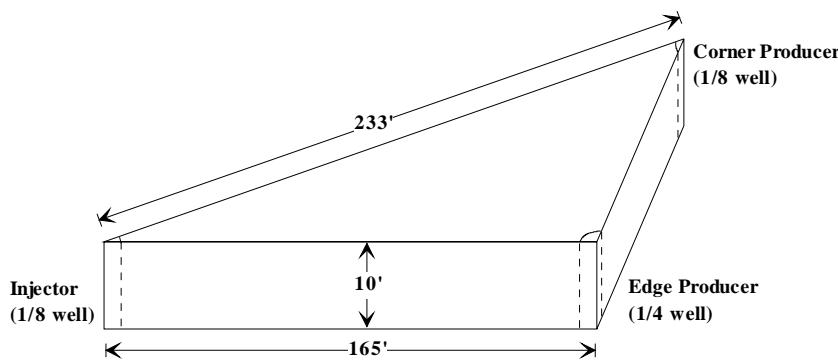


Figure E.8b: Basic reservoir one eighth of 2.5 acre nine-spot pattern, 10 ft. pay

---

## E.6 Local Grid Refinement

Local Grid Refinement (LGR) technique can provide higher resolution in areas of interest e.g. fractures, wellbore region, etc. with a considerable reduction of total number of grid blocks. This can be achieved by setting up a coarser grid and refining only the grid blocks of interest. The refinement is effected by specifying the number of subblocks in all directions in a certain coarse grid block. There is no interpolation of pressure, temperature, saturation, etc. to account for any variation in i, j or k direction and therefore it is recommended to use maximum of 4 subblocks in each direction.

Certain types of multi-level refinement are allowed (see section **Multi-level Regular Refinement** in the description of keyword \*REFINE in the Reservoir Description chapter). A subblock may also be a zero porosity block or a null block. By default the refined blocks have the same properties as the coarse block. However, any quantity entered as a grid array (see **Input of Grid Property Arrays** in Chapter 3) may be altered for the specific subblocks using subkeyword \*RG.

## E.7 Hybrid Grid

The treatment of near-wellbore flow phenomena has a strong influence on reservoir simulation results. In the current practice for field-scale simulation with communicating wells, a coarse Cartesian grid is used to discretize the reservoir, and wells are represented as sources or sinks within the coarse grid block. This representation of wells is usually not adequate, especially in reservoirs containing heavy viscous oil. The ability to inject steam depends on the viscosity reduction by heat conduction and/or convection in the vicinity of the wellbore. Because the well is located within a coarse grid block, either initial steam injectivity is very low or very high bottom hole pressure is encountered depending on the injection strategy. Similarly, for producers the coning behavior as well as breakthrough time matching cannot be properly handled.

The best grid for an accurate simulation of near-wellbore phenomena is the cylindrical grid system, because it follows the potential and stream lines near the well. However, it can be used only for a single well study. To incorporate accurate modelling of near-wellbore phenomena in field-scale simulation, Pedrosa and Aziz suggested the use of cylindrical refinement around wells within a Cartesian grid, referred to as a 'hybrid grid' (Figure E.9).

In the hybrid grid approach the entire reservoir is divided into two regions - well and reservoir region. In both regions the actual flow geometry is taken into account. When the well region can be considered homogeneous the flow will be radial and the fluid flow equations are formulated in cylindrical coordinates. When the well region is anisotropic the flow will be elliptical and the fluid flow in elliptical coordinate systems is modelled. Further away from wells, in the reservoir region, linear streamlines are considered which means the Cartesian coordinate system is used to describe the fluid flow.

### Well Region (Radial Flow)

The fluid flow equations used in this option are similar to the equations used in the simulator for the cylindrical coordinate system. However, steady state flow of an incompressible fluid through a radial block is assumed in evaluating the location of a nodal point. The nodal point is determined by the fact that the steady state pressure distribution and the volumetrically averaged block pressure must be equal to the nodal point. The result is that the location of the nodal point  $i$  depends only on the inner ( $r_{i-1/2}$ ) and outer block radius ( $r_{i+1/2}$ ):

$$r_i = r_{i-1/2} \exp \left[ \frac{\alpha^2}{\alpha_{-1}^2} \ln \alpha - \frac{1}{2} \right] \text{ where } \alpha = \frac{r_{i+1/2}}{r_{i-1/2}} \quad (\text{E7.1})$$

It is assumed that the coarse Cartesian block area is equal to the superimposed cylindrical grid area.

### Well Region (Elliptical Flow)

When the formation is anisotropic in the vicinity of the wellbore (in the plane perpendicular to well direction) the flow is elliptical. The y-z coordinate system is transformed to elliptical coordinates

$$z = \left( \frac{k_y}{k_z} \right)^{1/4} L \cosh \xi \cos \zeta$$

$k_y > k_z$  well is in x direction      (E7.2)

$$y = \left( \frac{k_z}{k_y} \right)^{1/4} L \sinh \xi \sin \zeta$$

where  $\xi$  represents a family of confocal ellipses and  $\zeta$  a family of confocal hyperbolas. Both have a semi-focal length

$$L = r_{we} \left[ (k_y - k_z) / (k_y + k_z) \right]^{1/2} \quad (E7.3)$$

Again, the equality of the original grid and the superimposed grid area is valid. The elliptical block distance is calculated as:

$$\Delta\xi = \frac{\xi_{nr+1/2} - \xi_{l+1/2}}{nr - 1} \quad (E7.4)$$

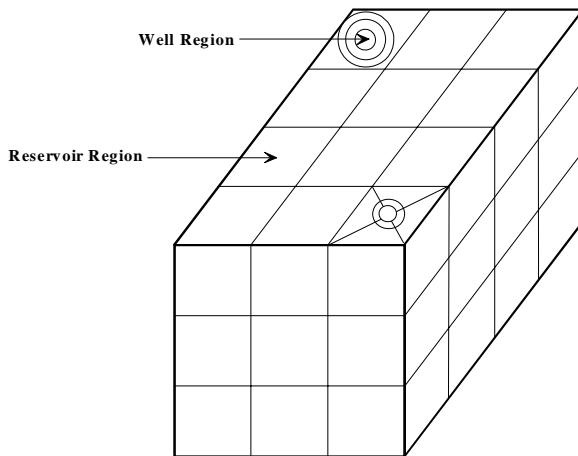
nr - number of radial subdivisions

With the same assumptions as in the radial case, the location of nodal points in the  $\xi$  direction is calculated from:

$$\xi_i = \frac{(\xi_{i+1/2} \sinh 2\xi_{i+1/2} - \xi_{i-1/2} \sinh 2\xi_{i-1/2}) - 0.5(\cosh 2\xi_{i+1/2} - \cosh 2\xi_{i-1/2})}{\sinh 2\xi_{i+1/2} - \sinh 2\xi_{i-1/2}} \quad (E7.5)$$

### Reservoir Region

Equations for fluid flow in a Cartesian coordinate system are used. Irregularly shaped grid blocks are formed at the well and reservoir region boundary. Depending on the flow direction, they are treated either as fictitious radial (elliptical) or fictitious Cartesian blocks with the same volume, transmissibilities and rock properties.



*Figure E.9: Schematic of a hybrid grid reservoir*

## E.8 Naturally Fractured Reservoirs

### Simulation Models for Naturally Fractured Reservoirs

Several models have been introduced to handle the processes which govern the fluid and heat flow in naturally fractured reservoirs (NFR). To model these processes effectively, the viscous, capillary, gravitational and diffusive effects must be quantified. The importance and interaction between these forces depend on the type (geometry) of the fractured reservoir and the recovery strategy. In all models, the reservoir is divided into matrix and fracture interacting continua with a superimposed computational grid (Figure E.10). In general, each grid block may contain several fracture and matrix continua (elements) which are lumped together. When a fractured reservoir exhibits substantial matrix heterogeneities, this lumping may lead to erroneous results. Therefore, one must be careful when choosing grid block sizes.

Generally, the models may be classified into two groups: dual porosity and dual permeability.

#### 1. Dual Porosity Models

Dual porosity models assume that the fracture network is the primary continuum for fluid flow. The low permeability, high storativity matrix is considered to be a sink or a source to the fracture, which is appropriate for well-fractured reservoirs which have complete matrix discontinuity. The models may be sub-divided according to their abilities to treat the fluid and heat flow.

##### a) Standard Dual-Porosity Model (DP)

As shown in Figure E.11, this is the simplest model to describe behavior in fractured reservoirs<sup>1,2,3</sup>. The matrix and the fracture communicate through a single exchange term. There is no direct communication between interblock matrices, i.e., neighboring blocks are connected through fracture flow only. The fluid or heat inside matrix can be transferred only to fracture. It is assumed that fracture and matrix within a grid block are at the same depth and, therefore, it is not possible to simulate gravity drainage effects with this model. A quasi steady state is assumed inside each matrix element which may lead to incorrect results in reservoirs with large matrix elements, particularly at the initial stages of reservoir depletion due to delayed matrix response.

##### b) Multiple Interacting Continua Model (MINC)

This model was proposed by Pruess, et al<sup>4</sup> for geothermal reservoirs and later was applied to hydrocarbon reservoirs by Gilman<sup>5</sup> and Wu, et al<sup>6</sup>. The matrix element is divided into several nested volume domains (Figure E.12) that communicate with each other. Therefore, pressure, saturation, temperature, etc., gradients are established inside matrix, allowing transient interaction between fracture and matrix. Due to the matrix discretization, the transmissibility for matrix-fracture flow is higher than in DP or dual permeability models for the same matrix size. This results in earlier and increased matrix-fracture response semi-analytical approaches<sup>7</sup> to modelling this transient behavior are not considered in this report.

c) Vertical Refinement Model (VR)

Several authors<sup>3,5,8</sup> used the VR model to consider gravitational effects and the gravity drainage mechanism. In this model matrix is refined in the vertical direction (Figure E.13), accounting for transient flow behavior in the matrix. Complete phase segregation in the fracture is assumed. The matrices communicate with only the fracture in the off-vertical directions, and with each other in the vertical direction. The matrix sub-blocks as well as the fracture have different depth and, hence, this model is suitable to simulate the gravity drainage process as well as processes with phase segregation inside the matrix. Similar to the MINC model, the fracture and matrix will start communicating earlier due to smaller matrix sub-blocks.

Again, other semi-analytical approaches to modelling gravity drainage are not considered in this report.

2. Dual Permeability Models (DK)

In a dual permeability model both the fracture network and the matrix participate in the fluid and heat flow. These models are suitable for moderately to poorly fractured reservoirs or fractured brecciated reservoirs<sup>9</sup> where the assumption of complete matrix discontinuity is not valid. They are also used for problems which require capillary continuity. The standard dual permeability model (DK)<sup>3,8-11</sup>, is similar to DP model, but has an additional communication between matrices of the adjacent grid blocks (Figure E.15). Gravity drainage can be simulated only to a certain degree, because of the quasi steady-state assumption inside matrix blocks. This degree will vary with the complexity of a process and would be quite low for thermal heavy oil/bitumen recovery, where the oil mobility is strongly temperature dependent. In this case, models with additional vertical matrix refinement should be used.

### Governing Equations

This section describes the conservation equations for the models mentioned in the previous chapter - dual-porosity, multiple interacting continua, vertical refinement and dual permeability. These models are implemented in the thermal simulator STARS. The set of equations is an extension of the set of equations for the single-porosity system. The equation set consists of mass and energy conservation equations for both fracture and matrix continua, e.g., the basic dual-porosity model will have twice as many equations for the same number of grid blocks as single-porosity model. First, the general form of equations will be shown, and later the special terms will be described for each model.

Mass balance in fracture, component ic

$$\sum_{ph=1}^{nph} \underbrace{\left[ \Delta \left[ T \rho_{ph} \lambda_{ph} x_{ph,ic} (\Delta p + \Delta P c_{ph} - \gamma_{ph} \Delta z) \right]_f \right]}_{\text{fracture-fracture flow}} + \underbrace{\sigma_p \rho_{ph} \lambda_{ph} x_{ph,ic} \Delta \Phi_{fm}}_{\text{fracture-matrix flow}} =$$

$$\left[ \underbrace{\frac{V}{\Delta t} \Delta_t \sum_{ph=1}^{nph} \phi \rho_{ph} S_{ph} x_{ph,ic}}_{\text{accumulation}} - \underbrace{\sum_{rx=1}^{nrx} s_{rx}}_{\text{reaction}} - \underbrace{q}_{\text{injection / production}} \right]_f \quad (E8.1)$$

Mass balance in matrix, component ic

$$[\text{matrix-matrix flow}] - \sum_{ph=1}^{nmp} \underbrace{\sigma_p \rho_{ph} \lambda_{ph} x_{ph,ic} \Delta \Phi_{fm}}_{\text{fracture-matrix flow}} =$$

$$\left[ \underbrace{\frac{V}{\Delta t} \Delta_t \sum_{ph=1}^{nph} \phi \rho_{ph} S_{ph} x_{ph,ic}}_{\text{accumulation}} - \underbrace{\sum_{rx=1}^{nrx} s_{rx}}_{\text{reaction}} - \underbrace{q^*}_{\text{injection / production}} \right]_f \quad (E8.2)$$

Total energy balance in fracture

$$\sum_{ph=1}^{nph} \underbrace{\Delta [T \rho_{ph} \lambda_{ph} H_{ph} (\Delta p + \Delta P c_{ph} - \gamma_{ph} \Delta z)]}_f + \underbrace{\Delta [T_c \Delta T]}_f + \underbrace{\Delta [T_c \Delta T]}_{\text{fracture heat conduction}}$$

$$\sum_{ph=1}^{nph} \underbrace{\sigma_p \rho_{ph} \lambda_{ph} H_{ph} \Delta \Phi_{fm} + \sigma_T \Delta T_{fm}}_{\text{fracture-matrix flow}}$$

$$= \left[ \underbrace{\frac{V}{\Delta t} \Delta_t \sum_{ph=1}^{nph} \phi \rho_{ph} S_{ph} U_{ph}}_{\text{fluid accumulation}} + \underbrace{(1-\phi)(\rho c_p)_r (T - T_r)}_{\text{rock accumulation}} \right]_f$$

$$- \sum_{rx=1}^{nrx} \underbrace{s_{ru_{rx}}}_{\text{reaction}} - \underbrace{qu}_{\text{injection / production}} - \underbrace{hloss}_{\text{heatloss O/U}} - \underbrace{hext}_{\text{external heaters}} \quad (E8.3)$$

Total energy balance in matrix

$$[\text{matrix - matrix flow}] - \sum_{ph=1}^{nph} \underbrace{\sigma_p \rho_{ph} \lambda_{ph} H_{ph} \Delta \Phi_{fm} + \sigma_T \Delta T_{fm}}_{\text{fracture-matrix flow}} =$$

$$= \left[ \underbrace{\frac{V}{\Delta t} \Delta_t \sum_{ph=1}^{nph} \phi \rho_{ph} S_{ph} U_{ph}}_{\text{fluid accumulation}} + \underbrace{(1-\phi)(\rho c_p)_r (T - T_r)}_{\text{rock accumulation}} \right]_m$$

$$-\sum_{rx=1}^{nrx} sru_{rx} - \underbrace{qu}_{\text{reaction}} - \underbrace{hloss}_{\text{injection / production}} - \underbrace{hext}_{\text{heatloss O/U}} - \underbrace{\text{external heaters}}_{\text{heaters}} \quad (\text{E8.4})$$

In addition to the conservation equations there is a constraint equation which may be solved simultaneously with the above reservoir flow equations:

Saturation constraint

$$\sum_{ph=1}^{nph} S_{ph} = 1 \quad (\text{E8.5})$$

or gas mole fraction constraint

$$\sum_{ic=1}^{nc} y_{ic} = 1 \quad (\text{E8.6})$$

The mobility in all the transfer terms is taken from the upstream direction, i.e., at some point of time during the simulation the fracture properties can be used to calculate the fracture-matrix flow. Heat conductivity for all transfer terms (fracture-matrix, fracture-fracture, matrix-matrix) is calculated as a sum of resistivities in all directions. It means that both fracture and matrix values are used.

In STARS the parameter  $\sigma_p$  called the transmissibility factor, represents the transmissibility in the fracture-matrix fluid flow term. There are two choices to calculate the  $\sigma_p$  depending on the \*SHAPE keyword.

The Gilman and Kazemi formulation is:

$$\sigma_p = 4V_b \sum_i \frac{k_{mi}^*}{L_i^2} \quad (\text{E8.7})$$

$L_i$  is the fracture spacing in x, y and z direction

$k_{mi}^*$  is the effective matrix permeability in all directions

$V_b$  is a block volume

The K-Harmonic formulation is:

$$\sigma_p = 4V_b \sum_i \left[ \frac{1}{L_i} \left( \frac{k_{fi}^* k_{mi}^*}{L_{fi} k_{mi}^* A_{fi} / A_{mi} + L_{mi} k_{fi}^*} \right) \right] \quad (\text{E8.7a})$$

$L_{fi}$  is the fracture width in all directions

$L_{mi}$  is the matrix size in all directions

$k_{fi}^*$  is the effective fracture permeability in all directions

$A_{fi}$  and  $A_{mi}$  is the fracture/matrix area perpendicular to the flow

$L_f$  and  $L_m$  are calculated from input fracture spacing  $L_x$ ,  $L_y$ ,  $L_z$  (also termed element sizes) in the x, y, and z directions which can be expressed as the sum of fracture widths  $L_{fx}$ ,  $L_{fy}$ ,  $L_{fz}$  and matrix sizes  $L_{mx}$ ,  $L_{my}$ ,  $L_{mz}$  for each direction. The number of matrix-fracture elements in a grid block is expressed by the ratio  $V_b/V$ . The K-Harmonic (KH) option of evaluating the  $\sigma_p$  is more general. The Gilman and Kazemi (GK) option assumes that fracture permeability is much higher than matrix permeability and therefore is a sub-set of K-Harmonic calculation. In many cases the GK assumption is true. When matrix permeability is fairly high or NFR is used together with options that increase matrix permeability e.g. dilation, rock dissolution then the KH option is more appropriate.

For thermal processes, conductive transport of energy between fracture and matrix must also be considered. The parameter  $\sigma_T$  denotes a thermal transmissibility in the fracture-matrix conductive flow. It is calculated as:

$$1/\sigma_T = \left[ \sum_i \frac{L_{fi}}{A_{mi}\kappa_f} + \frac{L_{mi}}{A_{mi}\kappa_m} \right] \frac{V}{4V_b} \quad (E8.8)$$

$\kappa_f, \kappa_m$  are fracture, matrix intrinsic heat conductivities.

While  $\kappa$  is isotropic, it can vary from grid block to grid block.

## Fracture and Matrix Properties

History matching a fractured reservoir correctly is much more difficult than matching a conventional reservoir, due to a higher number of uncertainties in input data (fracture and matrix properties, fracture distribution, etc). It is possible to match production rate histories with wrong data, but it is unlikely that such data will accurately predict future reservoir behavior. Therefore, it is important that appropriate values of fluid and rock properties be assigned to both fracture and matrix blocks.

### Fracture and Matrix Fractions:

Consider an element with gross volume  $V$  that is subdivided into a “matrix” block and a “fracture” block with gross volumes  $V_m$  and  $V_f$ , respectively; therefore  $V = V_m + V_f$ . Let  $F_f$  (\*FRFRAC) be the fraction of  $V$  that is deemed to be fracture block gross volume, so that

$$F_f = V_f / V$$

Normally  $V_f$  is the volume of the open fracture space and  $V_m$  is the gross volume of the porous rock. However, we consider also the case where the fracture block does in fact contain some (porous) rock. This situation is important for thermal simulation because a thin rock layer surrounding a fracture can be in thermal equilibrium with the fluids in the fracture at typical field time scales. Let the fracture gross volume  $V_f$  be subdivided further into open fracture volume  $V_{ff}$  and matrix gross volume  $V_{fm}$ , so that  $V_f = V_{ff} + V_{fm}$ . Therefore, the matrix-in-fracture fraction (\*FORMINFRAC) is defined as

$$F_{fm} = V_{fm} / V_f$$

When fracture block does not contain rock corresponds to  $F_{fm} = 0$ . Note that there is a distinction between “fracture” and “fracture block” and between “matrix” and “matrix block”. In general the “fracture block” contains the “fracture” and perhaps some “matrix (formation)”, whereas the “matrix block” contains the remaining “matrix”.

Property data for both matrix and fracture blocks are based on the volume or areas of the corresponding fundamental block. This principle will be applied to each of the properties discussed below. Therefore, for each property there is a distinction between “intrinsic” data, corresponding to the particular volume of interest, and “effective” data (denoted with \*) that is entered into the simulator. In general, user input data for a property depends upon fractions  $F_f$  and  $F_{fm}$  as well as the intrinsic data.

#### *Fracture-Width Case:*

Fracture width and matrix size is calculated from input values of \*DIFRAC, \*DJFRAC, \*DKFRAC and \*FRFRAC as follows:

$$\delta = 1 - (1 - F_f)^{1/b}$$

Parameter b depends on the number of fractured directions. When all directions are fractured  $b=3$ . The fracture width  $L_{fi} = L_i * \delta$  and the matrix size is  $L_{mi} = L_i - L_{fi}$ . i denotes a direction x, y or z.

For a simplified cubic fracture element where the true fracture width  $L_f$  is very small compared to the fracture spacing L. In this case volume fraction  $F_f$  reduces to  $3L_f/L$  and area fractions ( $A_f/A$ ) reduce to  $2L_f/L$ , as found in many papers.

#### *Porosity:*

The pore volume in the fracture block is the open fracture volume  $V_{ff}$  plus the pore volume of the rock (formation)  $\varphi_r V_{fm}$  that is included in the fracture block. A hypothetical intrinsic fracture porosity can be defined as

$$\varphi_f = (V_{ff} + \varphi_r V_{fm}) / V_f = 1 - F_{fm}(1 - \varphi_r)$$

$F_{fm}$  (\*FORMINFRAC) and  $\varphi_r$  (\*POROSITY \*FRACTURE) are input parameters. When fracture does not contain rock the intrinsic fracture porosity becomes 1.

The effective fracture porosity is:

$$\varphi_f^* = \varphi_f F_f$$

#### *Permeability:*

The following applies to each direction separately. Let  $A_m$ ,  $A_f$  and  $A$  be the cross-sectional areas for matrix, fracture and element, respectively, in the block and direction of interest. Area ratio ( $A_f/A$ ) normal to the fracture may be much different compared to a direction parallel to the fracture.

Intrinsic matrix permeability  $k_m$ , is the user input matrix block permeability (keywords \*PERMI \*MATRIX, etc.). The effective matrix permeability is used in the fracture-matrix transfer term calculation. It is expressed as:

$$k_m^* = k_m(A_m/A) \quad (E8.11)$$

Based on intrinsic fracture permeability  $k_f$ , the user input fracture permeability (effective value keywords \*PERMI \*FRACTURE, etc.) is

$$k_f^* = k_f(A_f/A) \quad (E8.12)$$

When fracture contains rock then  $k_f$  should be a weighted average of a permeability in the true fracture and the formation contained in the “fracture block”.

One has to keep in mind these relationships, particularly in well-fractured reservoirs where fracture spacings are fairly small ( $L_f$  may not be negligible compared to  $L$ ) and during simulation when fracture is assumed to be an extended region of high permeability.

User input permeability and porosity values for fracture and matrix can be obtained from core and log analysis data<sup>13</sup>. This task in many cases is very involved and gives only approximate values. The fracture spacing and matrix size can be estimated from the above mentioned analysis as well.

#### *Fluid Properties:*

Fluid properties (e.g., density, heat capacity) are applied to the fluid in the pore volume of both matrix and fracture blocks. Since user input porosities (matrix and fracture) give the desired pore volumes, the same (unmodified) fluid property data is used by both matrix and fracture blocks.

#### *Rock Heat Capacity:*

Volumetric rock heat capacity (incremental heat per volume per temperature difference) is applied to the rock volume, that is, the “ $1-\phi$ ” portion of the block. The input rock heat capacities are intrinsic values for both fracture and matrix. STARS calculates the intrinsic porosities (see above) and uses them to calculate the appropriate block heat capacity. The use of intrinsic values vs. effective is necessary due to possible changes in porosity during simulation. Porosity changes may result from dilation/recompaction, rock or solid components reactions, etc. Rock volume in the matrix block is  $V_m(1-\phi_m)$ , so the desired rock heat capacity is  $V_m(1-\phi_m)C_{pr}$ . Rock volume in the fracture block is  $V_{fm}(1-\phi_f)$ , so the desired rock heat capacity is  $V_{fm}(1-\phi_f)C_{pr}$ . Rock heat capacity may be the same value for fracture and matrix. When it is necessary different blocks as well as fracture and matrix may be assigned different rock heat capacity via the \*ROCKTYPE keyword.

#### *Heat Conductivity:*

Block thermal conductivity is the volume-weighted average of rock and phase values. Let  $\kappa_{fl}$  and  $\kappa_r$  be the intrinsic thermal conductivity for saturation-weighted fluid and rock, respectively. The intrinsic thermal conductivity for a block is, in general,

$$\kappa = \kappa_{fl} \phi + \kappa_r (1-\phi) \quad (\text{E8.15})$$

Multiplying this quantity by cross-sectional area and temperature gradient gives conductive heat flow.  $\kappa_{fl}$  and  $\kappa_r$  are the input values.  $\kappa_r$  may be the same for both fracture and matrix or may have different values assigned via \*ROCKTYPE keyword. The following applies to each direction separately since area factors may vary with direction (see *Permeability*, above) and thermal conductivities may be anisotropic.

Conductive heat flow in the fluid phases in the matrix block is  $A_m \kappa_{fl} \phi_m gradT$ . Conductive heat flow in the rock in the matrix block is  $A_m \kappa_r (1-\phi_m) gradT$ .

Conductive heat flow in the fluid phases in the fracture block is  $A_f \kappa_{fl} \phi_f gradT$ . Conductive heat flow in the rock (formation) in the fracture block is  $A_f \kappa_r (1-\phi_f) gradT$ .

Intrinsic fracture porosity is equal to one when fracture does not contain rock and only fluid will participate in all heat related calculations.

### *Other Properties:*

In addition, the fluid and rock interaction must be defined in terms of relative permeabilities and capillary pressure. Although it is difficult to measure these values in fractured cores, the behavior in the matrix is described in a way similar to a conventional reservoir. The characterization of these values in fracture is much more controversial. Usually, relative permeabilities in the fracture are defined as straight lines and capillary pressure is neglected, which should be a correct assumption for a thin fracture. However, the question is: how much rock is included in the so-called fracture? Do the values of user input fracture porosity and permeability obtained from log data represent only the response of the openings (real fractures) or even the response of a small matrix portion in the immediate fracture vicinity? If "fracture" contains rock then one has to account for the rock-fluid interaction on the fracture, i.e. consideration of capillary pressure, non-linear dependence of relative permeabilities on saturation and phase segregation in the fracture. These effects may be negligible if the rock amounts to a small percentage, although they may be important when the fracture is partially filled with high permeability rock (deposits). Firoozabadi<sup>14</sup> claims to measure capillary pressure in fracture as high as 0.275 MPa. In many fractured reservoirs where capillary effects are a main source of fracture-matrix interaction, these values can make a real difference in prediction of reservoir behavior. Other authors also realized that capillary discontinuity in the fracture in many circumstances causes erroneous results.

### **Dual-Porosity Formulation**

In the dual-porosity approach, the matrix blocks are assumed to be isolated and act as sinks or sources to the fracture network. The matrix-matrix flow in Equations E8.2 and E8.4 is zero. Because the fracture network is the primary flow continuum, as a default, the wells are connected to the fracture only ( $q$  and  $qu$  in Equations E8.2 and E8.4 are equal to 0). In DP the flow from/to the matrix to/from the fracture is considered in all fractured directions. Furthermore, the assumption of small fracture width ( $L_f \ll L_m$ ) is invoked in these formulae (standard DP implementation – \*SHAPE \*GK).

### **Multiple Interacting Continua Formulation**

In the MINC model the neighboring inner-block matrices communicate together and this action is depicted in the matrix-matrix flow term (Equations E8.2 and E8.4):

$$\text{matrix - matrix fluid flow} = \sum_{ph=1}^{nph} \sigma_{mm} \rho_{ph} \lambda_{ph} x_{ph,ic} \Delta \Phi_{mm} \quad (\text{E8.19})$$

The inner matrix blocks are nested, i.e. they and the fracture have the same depth; hence, the gravity term is not included in the potential calculation. Similarly,

$$\text{matrix - matrix heat flow} = \sum_{ph=1}^{nph} \sigma_{mm} \rho_{ph} \lambda_{ph} H_{ph} \Delta \Phi_{mm} + \sigma_{T,mm} \Delta T \quad (\text{E8.20})$$

The transmissibility factors  $\sigma_{mm}$  and  $\sigma_{T,mm}$  are defined as

$$\sigma_{mm} = 4 \left[ \frac{A_{m,i,x} A_x}{A_{m,x}} \frac{k_{mi,x} k_{m,i+1,x}}{(L_i k_{i+1} + L_{i+1} k_i)_{m,x}} + \frac{A_{m,i,y} A_y}{A_{m,y}} \frac{k_{mi,y} k_{m,i+1,y}}{(L_i k_{i+1} + L_{i+1} k)_{m,y}} + \right]$$

$$\left[ \frac{A_{m,i,z} A_z}{A_{m,z}} \frac{k_{mi,z} k_{m,i+1,z}}{(L_i k_{i+1} + L_{i+1} k)_{m,z}} \right] \frac{V_b}{V} \quad (E8.21)$$

$$\begin{aligned} \sigma_{T,mm} = 4\kappa & \left[ \frac{A_{m,i,x} A_x}{A_{m,x} (L_i k_{i+1} + L_{i+1} k_i)_{m,x}} + \frac{A_{m,i,y} A_y}{A_{m,y} (L_i k_{i+1} + L_{i+1} k_i)_{m,y}} + \right. \\ & \left. \frac{A_{m,i,z} A_z}{A_{m,z} (L_i k_{i+1} + L_{i+1} k_i)_{m,z}} \right] \frac{V_b}{V} \end{aligned} \quad (E8.22)$$

where  $L_i$  and  $L_{i+1}$  are subdomain matrix sizes. These flow terms are calculated for each subdomain and care must be taken in evaluating the area available to flow ( $A_{mi}$ ) between the inner matrix blocks. As a default, wells are perforated only in the fractures. However, a user may specify also perforations in the matrix (see STARS user manual \*PERF keyword).

### Vertical Refinement Formulation

This approach is a combination of dual-porosity with matrix refinement based on Gilman and Kazemi<sup>15</sup>. It is assumed that a fluid volume in horizontal fractures is negligible and therefore matrix communicates with vertical fractures only. In a 3-dimensional fracture the sub-matrices are connected to the fracture in the off-vertical directions and to each other in the vertical direction. The matrix-matrix flow term in Equations E8.2 and E8.4 represents the interaction between matrix sub-blocks and is evaluated as is shown in Equations E8.19 and E8.20. The gravity term is included in the potential gradient calculation between matrices and fractures as well.

The transmissibility factors  $\sigma_{mm}$  and  $\sigma_{T,mm}$  are expressed as:

$$\sigma_{mm} = 2 \left[ \frac{A_{m,i,z} A_z}{A_{m,z}} \frac{k_{mi,z} k_{m,i+1,z}}{(L_i k_{i+1} + L_{i+1} k)_{m,z}} \right] \frac{V_b}{V} \quad (E8.23)$$

$$\sigma_{T,mm} = 2\kappa \left[ \frac{A_{m,i,z} A_z}{A_{m,z} (L_i k_{i+1} + L_{i+1} k_i)_{m,z}} \right] \frac{V_b}{V} \quad (E8.24)$$

Complete phase segregation is assumed in the fracture. The water, oil and gas levels are determined almost the same way as in Gilman et al<sup>8</sup>. The formulas below are valid when water is the heavier liquid. The phase levels in the fracture are defined by (Figure E.14)

$$Z_w = \left[ \frac{S_w - S_{wc}}{1 - S_{or} - S_{wc}} \right]_f d_z \quad (E8.25)$$

$$Z_g = \left[ \frac{S_g - S_{gc}}{1 - S_{lr} - S_{gc}} \right]_f d_z \quad (E8.26)$$

$$Z_o = d_z - Z_w - Z_g \quad (E8.27)$$

Fracture pressure  $p_i$  at each depth  $Z_i$  corresponding to the center of an inner block matrix "i" is calculated as:

$$p_i = p_f + \gamma Z_o + \bar{\gamma} (Z_w - Z_i) + B \quad (E8.28)$$

where  $p_f$  is the fracture pressure. The appropriate gravity contributions depend on the relative positions of location depth  $Z_i$  and the segregated fluid heights, namely

$$\begin{aligned} Z_o + Z_w &< Z_i < d_z & \gamma = \gamma_g & \bar{\gamma} = \gamma_g \\ Z_w &< Z_i < Z_o + Z_w & \gamma = \gamma_o & \bar{\gamma} = \gamma_o \\ Z_w &> Z_i & \gamma = \gamma_w & \bar{\gamma} = \gamma_w \end{aligned} \quad (E8.29)$$

$$\begin{aligned} Z_g &> d_z / 2 & B = \gamma_g (Z_g - d_z / 2) \\ Z_g &< d_z / 2 \text{ and } Z_w < d_z / 2 & B = \gamma_o (Z_g - d_z / 2) \\ Z_g &< d_z / 2 \text{ and } Z_w > d_z / 2 & B = \gamma_w (d_z / 2 - Z_w) - \gamma_o Z_o \end{aligned} \quad (E8.30)$$

When fracture is in an upstream direction, its properties are used in calculation of fracture-matrix flow. Relative permeabilities in the fracture must be corrected to represent the phase segregation. The weighting factors, actually normalized saturations, for water, gas and oil phase are calculated as follows:

$$1 > wfw = \frac{Z_w - \left( Z_i - \frac{L_{mi}}{2} \right)}{L_{mi}} > 0 \quad (E8.31)$$

$$1 > wfg = \left( Z_i + \frac{L_{mi}}{2} \right) - (Z_w + Z_o) > 0 \quad (E8.32)$$

$$wfo = 1 - wfw - wfg \quad (E8.33)$$

When straight lines relative permeability curves with 0 to 1 range are used, then the weighting factors represent directly saturations as well as relative permeability. When relative permeability dependence on saturation is nonlinear and irreducible saturations are not zero, then relative permeabilities are calculated as usual using wfw, wfo and wfg values. Phase segregation in the fracture must also be considered in calculation of heat flow (both convective and conductive) from fracture to matrix. In tar sand reservoirs (Athabasca) it may happen that the oil phase is heavier than water. Analogous formulae can be written but are not given here.

Because of the assumption of negligible horizontal fracture volume, the use of this option is valid strictly for diminutive horizontal fracturing. In a 3-dimensional fracture with sizable fluid volume in a horizontal direction, fracture volume should be used instead of vertical distances in the formulae above and fracture should communicate with matrix in all directions. This idea needs to be investigated further.

## Dual-Permeability Formulation

In this model, the fluids and heat flow in the fracture network as well as in the matrices. Therefore, the matrix-matrix flow term (Equations E8.2 and E8.4) describes the flow between matrices in adjacent grid blocks:

$$\text{matrix - matrix fluid flow} = \sum_{\text{ph}=1}^{\text{nph}} \Delta \left[ T \rho_{\text{ph}} \lambda_{\text{ph}} x_{\text{ph},ic} (\Delta p + \Delta P c_{\text{ph}} - \gamma_{\text{ph}} \Delta z) \right] \quad (\text{E8.34})$$

$$\text{matrix - matrix heat flow} = \sum_{\text{ph}=1}^{\text{nph}} \Delta \left[ T \rho_{\text{ph}} \lambda_{\text{ph}} H_{\text{ph}} (\Delta p + \Delta P c - \gamma_{\text{ph}} \Delta z) \right] + \Delta [T_c \Delta T] \quad (\text{E8.35})$$

In some fractured reservoirs the inter-block matrix flow plays an important role and may be even higher than the flow between fracture and matrix. To avoid high mass and energy accumulation (lower production rates) in the matrix around the producer, or high flow rates in the fracture around the injector, the well should be perforated through the matrix blocks as well as the fracture ( $q$  and  $qu$  terms in Equation E8.2 and E8.4 may be different from zero). The additional matrix perforation must be specified by the user. Also, the assumption of small fracture width is again implied for \*SHAPE \*GK.

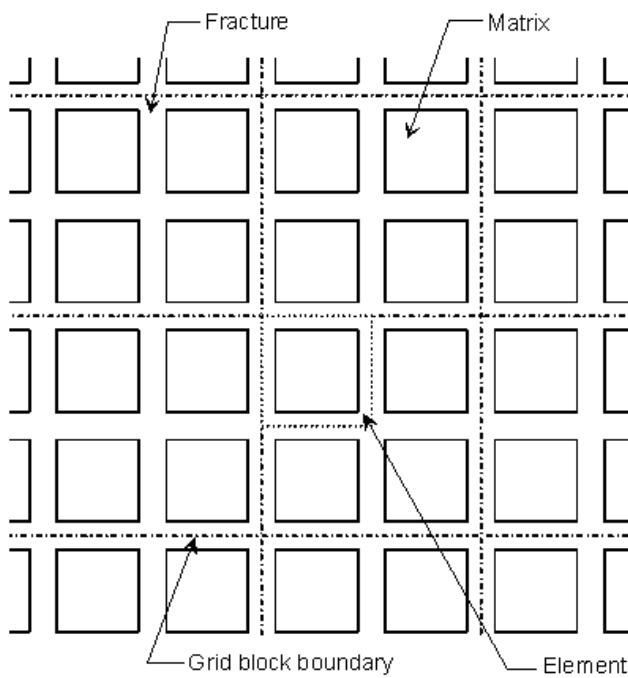
## Implementation and Numerical Considerations

The above mentioned models were implemented in the multicomponent, multiphase thermal simulator STARS. This simulator can also be run in an isothermal mode. The simulator allows a reservoir description whereby fracture spacing and grid block sizes can vary throughout the reservoir. In particular, the description allows fractured regions in addition to nonfractured (single porosity) regions.

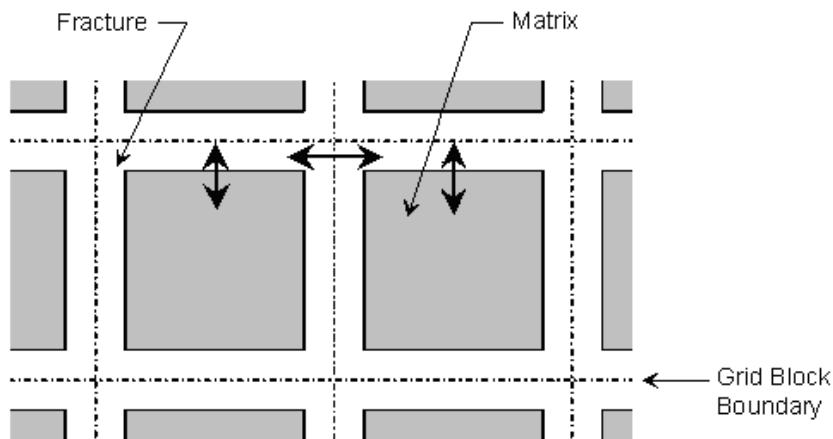
In fractured regions, each user input grid block represents one fracture block and one or more matrix blocks which are numbered consecutively. Except for the dual permeability option, the resulting Jacobian matrix is first preprocessed - fracture-matrix and inner block matrix-matrix equations are eliminated exactly - which results in a smaller incidence matrix. This is solved either by direct or iterative incomplete LU methods, accelerated with GMRES<sup>16</sup>.

Either fully implicit or adaptive implicit (AIM) techniques<sup>17</sup> can be used to solve the governing equations. The AIM allows further Jacobian matrix preprocessing and substantial run time savings when appropriate. Because of the high transmissibility contrasts between matrix and fracture blocks, the AIM can be employed effectively in naturally fractured reservoir models: fractures are implicit fractures and most matrix blocks are explicit.

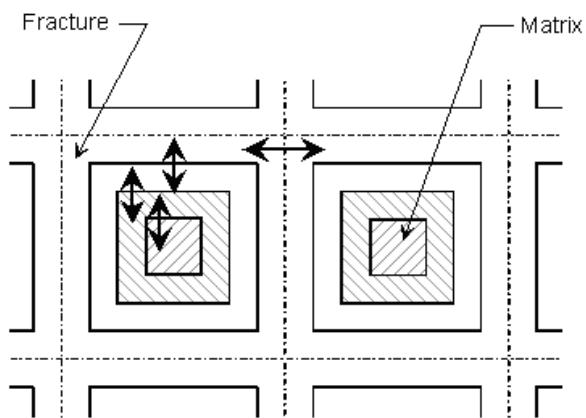
Because the physical behavior in matrix and fracture varies considerably, different convergence criteria as well as allowed changes in primary variables within a timestep should be used.



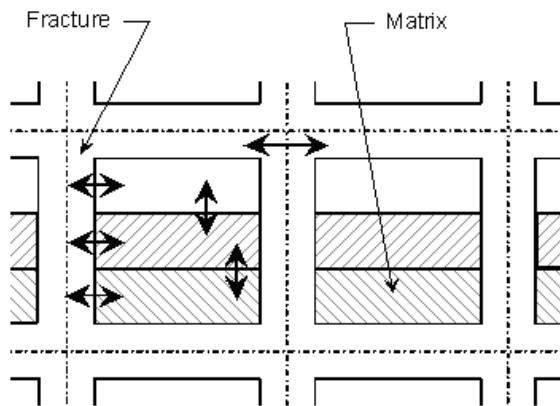
**Figure E.10:** Fractured reservoir



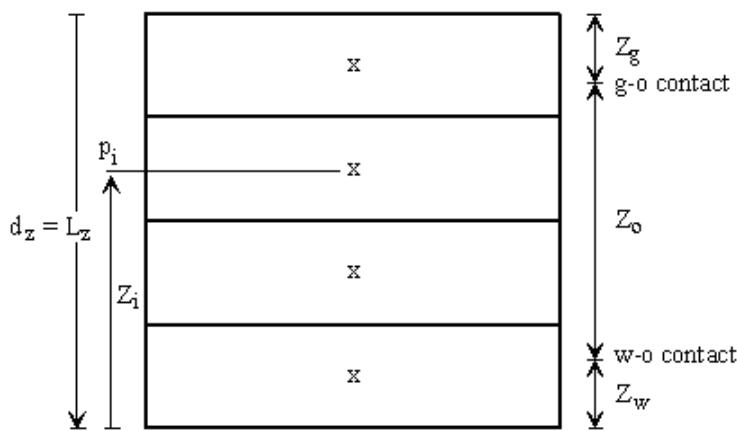
**Figure E.11:** Dual porosity model



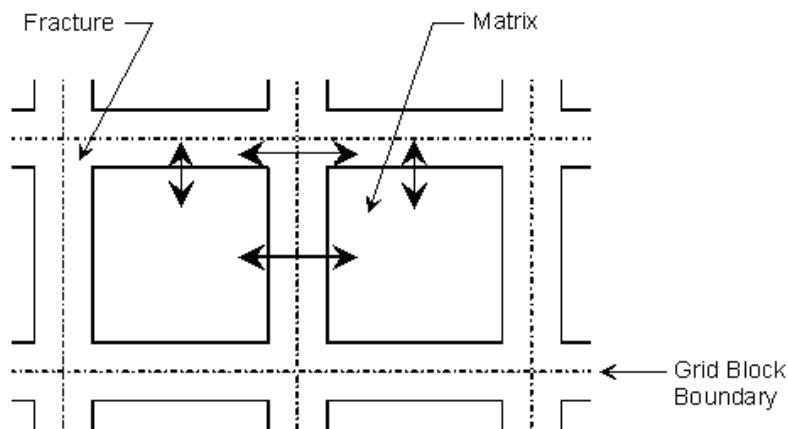
**Figure E.12:** MINC model



**Figure E.13:** Vertical refinement model



**Figure E.14:** Meaning of symbols used in the evaluation of fracture pressure  $P_i$  at each subdomain, based on fracture pressure at the midpoint of the fracture  $P_f$



**Figure E.15: Dual permeability model**

## References

1. Kazemi, K., Merrill, L.S. Jr., Porterfield, K.P., and Zeman, P.R., "Simulation of Water-Oil Flow in Naturally Fractured Reservoirs," SPEJ, December 1976, pp. 317-326.
2. Thomas, L.K., Dixon, T.N., and Pierson, R.G., "Fractured Reservoir Simulation," SPEJ, February 1983, pp. 42-54.
3. Lee, B.Y.Q., and Tan, T.B.S., "Application of a Multiple Porosity/Permeability Simulator in Fractured Reservoir Simulation," SPE 16009, presented at the 9th SPE Symposium on Reservoir Simulation, San Antonio, Texas, February 1-4, 1987.
4. Pruess, K., and Narasimhan, T.N., "A Practical Method for Modelling Fluid and Heat Flow in Fractured Porous Media," SPEJ, February 1985, pp. 14-26.
5. Gilman, J.R., "An Efficient Finite-Difference Method for Simulating Phase Segregation in the Matrix Blocks in Double-Porosity Reservoirs," SPEJ, July 1986, pp. 403-413.
6. Wu, Y.S., and Pruess, K., "A Multiple-Porosity Method for Simulation of Naturally Fractured Petroleum Reservoirs," SPEJ, February 1988, pp. 327-336.
7. Pruess, K., and Wu, Y.-S., "A New Semianalytical Method for Numerical Simulation of Fluid and Heat Flow in Fractured Reservoirs," SPE 18426, presented at the SPE Symposium on Reservoir Simulation, Houston, Texas, February 6-8, 1989.
8. Gilman, J.R., and Kazemi, H., "Improve Calculations for Viscous and Gravity Displacement in Matrix Blocks in Dual-Porosity Simulators," SPE 16010, presented at the 9th SPE Symposium on Reservoir Simulation, San Antonio, Texas, February 1-4, 1987.
9. Hill, A.C., and Thomas, G.W., "A New Approach for Simulating Complex Fractured Reservoirs," SPE 13537, presented at the SPE Middle East Oil Technical Conference and Exhibition, Bahrain, March 11-14, 1985.

10. Chen, W.H., Wasserman, M.L., and Fitzmorris, R.E., "A Thermal Simulator for Naturally Fractured Reservoirs," SPE 16008, presented at the 9th SPE Symposium on Reservoir Simulation, San Antonio, Texas, February 1-4, 1987.
11. Fung, L.S.-K., and Collins, D.A., "An Evaluation of the Improved Dual Porosity Model for the Simulation of Gravity Effects in Naturally Fractured Reservoirs," CIM 88-39-05, presented at the 39th Annual Technical Meeting of Petroleum Society of CIM, Calgary, Alberta, June 12-16, 1988.
12. Coats, K.H., "Implicit Compositional Simulation of Single Porosity and Dual Porosity Reservoirs," 1st International Forum on Reservoir Simulation, Alpbach, Austria, September 12-16, 1988.
13. Saidi, A.M., Reservoir Engineering of Fractured Reservoirs, Total Edition Press, 1987.
14. Firoozabadi, A., Hauge, J., "Capillary Pressure in Fractured Porous Media," SPE 18747, presented at the SPE California Regional Meeting, Bakersfield, CA, April 5-7, 1989.
15. Gilman, J.R., and Kazemi, H., "Improvements in Simulation of Naturally Fractured Reservoirs," SPE 10511, presented at the 6th SPE Symposium on Reservoir Simulation, New Orleans, Louisiana, January 31 - February 3, 1982.
16. Nghiem, L., and Rozon, B., "A Unified and Flexible Approach for Handling and Solving Large Systems of Equations in Reservoir Simulation," 1st International Forum on Reservoir Simulation, Alpbach, Austria, September 12-16, 1988.
17. Oballa, V., Coombe, D.A., and Buchanan, W.L., "Adaptive Implicit Method in Thermal Simulation," SPE 18767, presented at the SPE California Regional Meeting, Bakersfield, California, April 5-7, 1989.

# Appendix F: Equations

## Overview

This appendix contains details of the equations solved by STARS, divided into the following sections.

- F.1 Overview
- F.2 Conservation Equations
- F.3 Phase Equilibrium Relationships
- F.4 Well Equations
- F.5 Summary of Conservation Equations
- F.6 Solution of Nonlinear Equations – Newton’s Method
- F.7 Solution of Linear Equations – General Sparse Solver
- F.8 Treatment of Solid Components
- F.9 Adaptive-Implicit Method
- F.10 Use of Constraint Equations in the Sxy Formulation

---

## F.1 Overview

These equations are the result of expressing all the relevant physical phenomena in mathematical form. The equations are reviewed here so that the user of a thermal model can understand why so much input data is required, where it is used, and how the various property models are related to the final result.

There is one conservation equation for each chemical component for which a separate accounting is desired, along with some equations describing phase equilibrium between phases. There exists a set of these equations for each region of interest, which is usually a discretized grid block. Lastly, there is an equation describing the operating condition of each injection and production well.

## F.2 Conservation Equations

A conservation equation is constructed for each component of a set of identifiable chemical components that completely describe all the fluids of interest.

All conservation equations are based on a region of interest (with volume V) in which

- rate of change of accumulation
- = net rate of inflow from adjacent regions
- + net rate of addition from sources and sinks

Each of these three terms will be considered separately, below.

### Accumulation Terms

The total gross volume of a grid block may be composed of the following:

- $V_r$  – solid (inert) rock matrix (r)
- $V_s$  – solid and adsorbed component (s)
- $V_w$  – water or aqueous phase (w)
- $V_o$  – oil or oleic phase (o)
- $V_g$  – gaseous phase (g)

The total (gross, bulk) volume is

$$V = V_r + V_s + V_w + V_o + V_g \quad (\text{F2.1})$$

The fluid volume is defined as:

$$V_f = V_w + V_o + V_g \quad (\text{F2.2})$$

and the void volume is defined as:

$$V_v = V - V_r = V_f + V_s \quad (\text{F2.3})$$

Void porosity is defined as:

$$\phi_v = V_v / V \quad (\text{F2.4})$$

Fluid porosity is defined as

$$\phi_f = V_f / V = (V_v - V_s) / V = (V_v / V) \bullet (1 - V_s / V_v) \quad (\text{F2.5})$$

If we substitute the definition of  $\phi_v$ , and recognize that  $V_s/V_v$ , the fraction of void volume occupied by solid and adsorbed components together is equal to  $\sum c_{si}/\rho_{si}$ , then

$$\phi_f = \phi_v \bullet (1 - \sum c_{si} / \rho_{si}) \quad (\text{F2.6})$$

Note that if there is no solid or adsorbed component, then  $c_{si} = 0$  and  $V_s = 0$ , making  $V_v = V_f$  and  $\phi_v = \phi_f$ .

Keyword \*VOLCONST provides two ways of specifying the volume constraint.

Subkeyword \*BULK makes bulk volume V constant, so rock volume  $V_r = V \cdot (1 - \phi_v)$  varies with void porosity. Alternately, subkeyword \*ROCK makes rock volume  $V_r$  constant, so bulk volume  $V = V_r / (1 - \phi_v)$  varies with porosity. In each case all other volumes are derived from V and  $V_r$ , that is,  $V_v = V \cdot \phi_v$ ,  $V_f = V \cdot \phi_f$ , etc.

The saturations are defined as

$$\begin{aligned} S_w &= V_w / V_f = V_w / \varphi_f V, \\ S_o &= V_o / V_f = V_o / \varphi_f V, \text{ and} \\ S_g &= V_g / V_f = V_g / \varphi_f V, \text{ so that} \\ S_w + S_o + S_g &= 1 \end{aligned} \quad (\text{F2.7})$$

The accumulation term for a flowing and adsorbed component  $i$  is

$$V \frac{\partial}{\partial t} [\varphi_f (\rho_w S_w w_i + \rho_o S_o x_i + \rho_g S_g y_i) + \varphi_v A d_i] \quad (\text{F2.8})$$

In F2.8: remove “V” at front; replace  $\varphi_f$  with  $V_f$ ; replace  $\varphi_v$  with  $V_v$

The accumulation term for solid component is

$$V \frac{\partial}{\partial t} [\varphi_v c_i] \quad (\text{F2.9})$$

In F2.9: remove “V” at front; replace  $\varphi_v$  with  $V_v$

The accumulation term for energy is

$$V \frac{\partial}{\partial t} [\varphi_f (\rho_w S_w U_w + \rho_o S_o U_o + \rho_g S_g U_g) + \varphi_v c_s U_s + (1 - \varphi_v) U_r] \quad (\text{F2.10})$$

In F2.10: remove “V” at front; replace  $\varphi_f$  with  $V_f$ ; replace  $\varphi_v$  with  $V_v$ ; replace  $(1 - \varphi_v)$  with  $V_r$

Here  $U_j$ ,  $j=w,o,g,s$  are the internal energies as a function of temperature and phase composition, and

$$\rho_j, j = w, o, g$$

are fluid phase densities.  $U_r$  is energy per rock volume, and  $c_s$  is total solid concentration.

### Flow Terms

The flow term of flowing component  $I$  between two regions is

$$\rho_w v_w w_i + \rho_o v_o x_i + \rho_g v_g y_i + \phi \rho_w D_{wi} \Delta w_i + \phi \rho_g D_{gi} \Delta y_i + \phi \rho_o D_{oi} \Delta x_i \quad (\text{F2.11})$$

Solid components have no flow terms.

The flow term of energy between two regions is

$$\rho_w v_w H_w + \rho_o v_o H_o + \rho_g v_g H_g + K \Delta T \quad (\text{F2.12})$$

The volumetric flow rates are

$$v_j = T \left( \frac{k_{rj}}{\mu_j r_j} \right) \Delta \Phi_j \quad j = w, o, g \quad (\text{F2.13})$$

1. T is the transmissibility between the two regions, accounting for the cross sectional area, the node spacing and other geometrical considerations (e.g. partial grid blocks), as well as the permeability at the interface. The units of transmissibility are therefore

$$\left(\frac{A}{l}\right)^{\text{eff}} k^{\text{eff}}.$$

A block centered grid system is used<sup>1</sup>. Information regarding the nine-point option<sup>2</sup> is found in SPE 12248. The effective permeability  $k$  is an area weighted harmonic average of the absolute permeability in the two regions, correct for rectangular, radial and variable - thickness grids.

2.  $D_{ji}$  ( $j=w,o,g$ ) are the component dispersibilities in the three phases and are again the product of geometric factors and component dispersion coefficients, with units

$$\left(\frac{A}{l}\right)^{\text{eff}} D_{ji}^{\text{eff}}.$$

The effective dispersion coefficients at the interface are the geometric mean (square root of the product) of the dispersion coefficients input for the two regions.

3. K is the thermal transmissibility at the interface between the two regions, with units

$$\left(\frac{A}{l}\right)^{\text{eff}} \lambda^{\text{eff}}.$$

The effective thermal conductivity at the interface is the harmonic average of the two regions, that is, resistance to heat conduction is in series.

4. Values of  $kr_j$ ,  $\mu_j$ ,  $p_j$ ,  $r_j$ ,  $H_j$ ,  $w_i$ ,  $x_i$  and  $y_i$  are taken from the phase upstream region. The phase resistance factors  $r_j$  are normally 1.0 but large values are associated with blocking phenomena.
5. The potential at a grid node is  $\Phi_j = p_j - \lambda_{jgh}$ . The potential difference  $\Delta\Phi_j$  is the value at the node of the adjacent region minus the value at the node of the current region of interest. A positive value for  $\Delta\Phi_j$  represents inflow, a negative value gives outflow. The concentration differences  $\Delta_{wi}$ ,  $\Delta_{xi}$ ,  $\Delta_{yi}$  are the differences in phase concentrations between the nodes, following the same sign convention as  $\Delta\Phi$ . If a phase is not present in one of the adjacent blocks, the concentration difference is set to zero (no dispersive transport).  $\Delta T$  is the temperature drop between the nodes, again following the same sign convention as  $\Delta\Phi$ .

### Well Source/Sink Terms

Well source/sink terms are the means by which all the thermal EOR processes are driven.

The well source/sink term for flowing component  $i$  is

$$\rho_w q_{wk} w_i + \rho_o q_{ok} x_i + \rho_g q_{gk} y_i \quad (\text{F2.14})$$

Solid components have no well terms.

The well source/sink term for energy is

$$\rho_w q_{wk} H_w + \rho_o q_{ok} H_o + \rho_g q_{gk} H_g \quad (F2.15)$$

Note the similarity between these terms and the interblock flow terms discussed above. The volumetric flow rate  $q$  is analogous to  $V$ , but is calculated very differently.

The well phase rates are

$$q_{jk} = I_{jk} \bullet (p_{wfk} - p_k) \quad j=w,o,g \quad (F2.16)$$

1. Subscript k refers to the fact that the region of interest contains layer no. K of a well which may be completed also in other blocks or regions.
2.  $I_{jk}$  is the phase  $j$  index for well layer  $k$ , and may be specified in various ways.
3.  $p_k$  is the node pressure of the region of interest which contains well layer  $k$ .
4.  $p_{wfk}$  is the flowing wellbore pressure in well layer  $k$ .

The condition for injection is  $p_{wfk} > p_k$  in which case  $q_{jk}$  is positive and fluid properties are taken from wellbore conditions.

The condition for production is  $p_{wfk} < p_k$  in which case  $q_{jk} < 0$  and fluid properties are taken from the producing region.

A wellbore heatloss model may be used to estimate  $H_w$  and  $H_g$  for injected water as a function of time.

### Chemical Reaction and Interphase Mass Transfer Source/Sink Terms

The reaction source/sink term for component i is

$$V \sum_{k=1}^{n_r} (s'_{ki} - s_{ki}) \bullet r_k \quad (F2.17)$$

and the reaction source/sink term for energy is

$$V \sum_{k=1}^{n_r} H_{rk} r_k \quad (F2.18)$$

1.  $s'_{ki}$  is the product stoichiometric coefficient of component I in reaction k.
2.  $s_{ki}$  is the reactant stoichiometric coefficient of component I in reaction k.
3.  $H_{rk}$  is the enthalpy of reaction k.
4.  $r_k$  is the volumetric rate of reaction k, calculated from a model for reaction kinetics.

## **Heat Loss Source/Sink Terms**

The heat loss source/sink term for energy is

$$\sum_{k=1}^{n_f} HL_k + HL_v + HL_c \quad (F2.19)$$

1.  $HL_k$  is the rate of heat transfer to the region of interest through block face number  $k$ , from the adjacent formation. The heat transfer rate and heat accumulated in the overburden are calculated using an analytical solution for an infinite overburden. Heat flow back into the reservoir block may occur.
2.  $HL_v$  is the rate of heat transfer calculated from a convective model.
3.  $HL_c$  represents a constant heat transfer model.

## **Thermal Aquifer Source/Sink Terms**

The aquifer source/sink term for water component is

$$\sum_{k=1}^{n_f} \rho_w qaq_{wk} \quad (F2.20)$$

and for the energy

$$\sum_{k=1}^{n_f} (HA_{CV} + HA_{CD})_k \quad (F2.21)$$

1.  $qaq_w$  is a volumetric water flow rate through a block face  $k$  to/from the adjacent aquifer.
2.  $HACV$  is a rate of heat transferred by convection to/from the adjacent aquifer.
3.  $HACD$  is a rate of heat transferred by conduction to/from the adjacent aquifer.

All flow rates are calculated using a semi-analytical model with various boundary conditions.

## F.3 Phase Equilibrium Relationships

The phase mole fractions are related by the equilibrium ratios, or K values:

$$\begin{aligned}y_i &= K_i^{go} x_i \quad ; \quad x_i = K_i^{og} y_i \\x_i &= K_i^{ow} w_i \quad ; \quad w_i = K_i^{wo} x_i \\w_i &= K_i^{wg} y_i \quad ; \quad y_i = K_i^{gw} w_i\end{aligned}\quad (\text{F3.1})$$

From these definitions, it can be seen that only two of the above K values are independent (for three phases). The choice employed is made for convenience and can differ for different components (see Section D.3).

The phase mole fractions are also constrained by

$$\sum_{i=1}^{n_c} y_i = 1 \text{ when } S_g > 0, \quad (\text{F3.2})$$

$$\sum_{i=1}^{n_c} x_i = 1 \text{ when } S_o > 0, \text{ and} \quad (\text{F3.3})$$

$$\sum_{i=1}^{n_c} w_i = 1 \text{ when } S_w > 0. \quad (\text{F3.4})$$

The phase pressures and saturations are constrained by

$$\begin{aligned}S_w + S_o + S_g &= 1 \\p_w &= p_o - p_{cow}(S_w), \text{ and} \\p_g &= p_o + p_{cog}(S_g)\end{aligned}\quad (\text{F3.5})$$

---

## F.4 Well Equations

Simple single-block wells may be characterized with a constant rate or pressure, but a fully-coupled treatment of a well completed in several blocks requires a more comprehensive approach<sup>3</sup>. Each equation listed below represents a well operating condition, and exactly one equation per active well is in force at any one time.

Of the  $n_{\text{lay}}$  layers of a well, one is designated as the bottom-hole layer; its flowing wellbore pressure is  $p_{\text{wf}}$ ,

The volumetric well phase rate  $q_{jk} = I_{jk} (p_{\text{wf}} - p_k)$  was defined in the well source/sink section, earlier in this chapter. The index  $I_{jk}$  may contain the mobility factor ( $k_{rj}/\mu_j$ ), through which the well equations can be tightly coupled to the reservoir conditions. This explains why conservation equations and well equations should be solved simultaneously.

In the following, the subscript “spec” indicates a quantity specified by the user as an operating condition. These equations apply to both injection and production wells.

### Constant Pressure

$$p_{\text{wf}} = p_{\text{spec}} \quad (\text{F4.1})$$

This is the most simple well equation. Rates are calculated, and can be checked against auxiliary operating constraints.

### Constant Water Rate

$$\sum_{k=1}^{n_{\text{lay}}} q_{wk} = q_{\text{spec}} \quad (\text{F4.2})$$

This is solved simultaneously with the conservation equations, with  $p_{\text{wf}}$  as an additional variable. Even though  $q_{\text{spec}}$  is constant, the distribution of water to different layers depends on  $I_{jk}$  which can change with time.

### Constant Oil Rate

$$\sum_{k=1}^{n_{\text{lay}}} q_{ok} = q_{\text{spec}} \quad (\text{F4.3})$$

### Constant Gas Rate

$$\sum_{k=1}^{n_{\text{lay}}} q_{gk} = q_{\text{spec}} \quad (\text{F4.4})$$

### Constant Liquid rate

$$\sum_{k=1}^{n_{\text{lay}}} (q_{wk} + q_{ok}) = q_{\text{spec}} \quad (\text{F4.5})$$

## Constant Steam Production Rate

$$\frac{1}{\rho_w^{ST}} \left\{ \sum_{k=1}^{n_{lay}} q_{gk} y_1 \rho_g \right\} = q_{spec} \quad (F4.6)$$

where  $y_1$  and  $\rho_g$  are values from the grid block containing well layer k.

The water, oil, gas and liquid rates are generally specified at surface conditions (although production wells can alternately employ bottomhole (reservoir) rates). In case of surface conditions,

- a) the mole rate  $\rho_w q_{wk} w_i + \rho_o q_{ok} x_i + \rho_g q_{gk} y_i$  is evaluated for each component;
- b) a flash is performed, to obtain phase splits and compositions using surface condition K-values;
- c) surface densities are calculated;
- d) surface mole phase rates are multiplied by surface densities to get surface volume rates.

The wellbore pressure  $p_{wfk}$  at each layer is obtained by adding  $p_{wf}$  (at  $k=1$ ) the accumulated fluid head.

$$p_{wfk} = p_{wf} + \int_{h_1}^{h_k} \gamma_{av} g dh \quad (F4.7)$$

where  $h_k$  denotes the elevation of layer k, and  $\gamma_{av}$  denotes an average mass density of the fluids in the wellbore.

## F.5 Summary of Conservation Equations

The (spatially discretized) conservation equation of flowing component I is

$$\begin{aligned}
 & V \frac{\partial}{\partial t} [\varphi_f (\rho_w S_w w_i + \rho_o S_o x_i + \rho_g S_g y_i) + \varphi_v A d_i] \\
 &= \sum_{k=1}^{n_f} [T_w \rho_w w_i \Delta \Phi_w + T_o \rho_o x_i \Delta \Phi_o + T_g \rho_g y_i \Delta \Phi_g] + V \sum_{k=1}^{n_r} (s'_{ki} - s_{ki}) r_k \\
 &+ \sum_{k=i}^{n_f} [\phi D_{wi} \rho_w \Delta w_i + \phi D_{oi} \rho_o \Delta x_i + \phi D_{gi} \rho_g \Delta y_i] + \delta_{iw} \sum_{k=1}^{n_f} \rho_w q_{adwk} \\
 &+ \rho_w q_{wk} w_i + \rho_o q_{ok} x_i + \rho_g q_{gk} y_i \quad [\text{well layer } k]
 \end{aligned} \tag{F5.1}$$

In F5.1: remove “V” at front; replace  $\varphi_f$  with  $V_f$ ; replace  $\varphi_v$  with  $V_v$

where  $n_f$  is the number of neighboring regions or grid block faces.

The conservation equation of solid component i is

$$V \frac{\partial}{\partial t} [\varphi_v c_i] = V \sum_{k=1}^{n_r} (s'_{ki} - s_{ki}) r_k \tag{F5.2}$$

In F5.2: remove “V” at front; replace  $\varphi_v$  with  $V_v$

The (spatially discretized) conservation equation of energy is

$$\begin{aligned}
 & V \frac{\partial}{\partial t} [\varphi_f (\rho_w S_w U_w + \rho_o S_o U_o + \rho_g S_g U_g) + \varphi_v c_s U_s + (1 - \varphi_v) U_r] \\
 &= \sum_{k=1}^{n_f} [T_w \rho_w H_w \Delta \Phi_w + T_o \rho_o H_o \Delta \Phi_o + T_g \rho_g H_g \Delta \Phi_g] + \sum_{k=1}^{n_f} K \Delta T \\
 &+ \rho_w q_{wk} H_w + \rho_o q_{ok} H_o + \rho_g q_{gk} H_g \quad [\text{well layer } k] \\
 &+ V \sum_{k=1}^{n_r} H_{rk} r_k + H L_o + H L_v + H L_c + \sum_{k=1}^{n_r} (H A_{CV} + H A_{CD})_k
 \end{aligned} \tag{F5.3}$$

In F2.10: remove “V” at front; replace  $\varphi_f$  with  $V_f$ ; replace  $\varphi_v$  with  $V_v$ ; replace  $(1-\varphi_v)$  with  $V_r$

The phase transmissibilities  $T_j$  are

$$T_j = T \left( \frac{k_{rj}}{\mu_j r_j} \right) \quad j = w, o, g \tag{F5.4}$$

## F.6 Solution of Nonlinear Equations – Newton's Method

See Aziz and Settari<sup>4</sup> for a general review of flow equation solution methods. Of the equations discussed above, the following neq equations are solved simultaneously for each grid block, along with the well equations. It is assumed that the appropriate time discretization has been done<sup>1</sup>. Most generally, there are

- nc component conservation equations
- energy conservation equation
- phase constraint equation  $\sum_{i=1}^{n_c} y_i = 1$  or  $S_w + S_o + S_g = 1$  (optional)

The use/nonuse of a phase constraint equation depends on the choice of thermal flash algorithm employed. When required, either the gas mole fraction or the saturation constraint equation is solved, depending upon phase equilibrium conditions. The other constraint is satisfied during calculation of the properties.

For isothermal problems, obviously the energy conservation equation, along with the phase constraint equation, is not required.

Equations are solved simultaneously, using Newton's method, in a generalized form which can handle many coupled equations. The equations summarized above are written in residual form as

$$R_i = [\text{net inflow rate}] + [\text{net source/sink rate}] - [\text{rate of change of accumulation}] \quad (\text{F6.1})$$

and the equation is solved when  $R_i = 0$ . Evaluation of the residuals  $R_i$  amounts to calculating all the terms in the equations. The following calculation sequence is used.

1. Choose primary variables
2. K values
3. Remaining saturations and mole fractions
4. Densities, enthalpies, internal energies
5. Reaction rates, solid concentration, reaction source/sink terms
6. Porosity and accumulation terms
7. Relative permeabilities, viscosities, velocities, flow terms
8. Well rates and source/sink terms
9.  $R_i$  for  $n_{c+1}$  conservation equations and one phase constraint (when required)

*Table F.1: Residual calculation sequence*

If there are nb active grid blocks and nw open wells, then the total number of equations will be

$$N_{eq} = nb \bullet (neq) + nw \quad (F6.2)$$

There are also  $N_{eq}$  primary variables. Let  $X_i$  represent all primary variables, with  $i=1$  to  $N_{eq}$ . In general, each residual  $R_i$  could depend on each  $X_i$ , which is written as

$$\mathbf{R} = \mathbf{R}(\mathbf{X}) \quad (F6.3)$$

where  $\mathbf{R}$  and  $\mathbf{X}$  are  $N_{eq}$ -length vectors. Advancing the solution over one timestep consists of solving  $\mathbf{R}(\mathbf{X}) = \mathbf{0}$ . This is accomplished using Newton's method, which is written as

$$\mathbf{X}^{k+1} = \mathbf{X}^k - [\mathbf{J}^k]^{-1} \bullet \mathbf{R}^k \quad \text{or} \quad \mathbf{J}^k (\mathbf{X}^{k+1} - \mathbf{X}^k) = -\mathbf{R}^k \quad (F6.4)$$

where  $\mathbf{J} = d\mathbf{R}/d\mathbf{X}$  is the Jacobian matrix of derivatives and k is the Newton iteration number.

The initial  $\mathbf{X}^0$  is usually  $\mathbf{X}^N$ , the solution to the previous timestep. The iterative process is considered converged when both  $(\mathbf{X}^{k+1} - \mathbf{X}^k)$  and R are sufficiently small, at which time the solution at the current time is  $\mathbf{X}^{N+1} = \mathbf{X}^{k+1}$ .

The entries in the Jacobian are

$$J_{ij} = \frac{\partial R_i}{\partial X_j} \quad i = 1 \text{ to } N_{eq}, j = 1 \text{ to } N_{eq} \quad (F6.5)$$

In general, J has  $N_{eq}^2$  entries.

However, entries corresponding to i and j from unconnected grid blocks or wells will be zero. In fact, most of the  $J_{ij}$  entries will be zero, making J a banded sparse matrix. An example of such a sparse matrix is shown in Figure F.1 in which each small square corresponds to a  $neq \times neq$  submatrix; an empty square denotes a zero submatrix due to no grid connection. The sparse matrix solver used to solve the equations is described in reference 7.

The non-zero Jacobian entries are estimated using numerical differentiation.

$$J_{ij} \cong \frac{R_i(\mathbf{X} + \delta X_j) - R_i(\mathbf{X})}{\delta X_j} \quad (F6.6)$$

where the sum  $\mathbf{X} + \delta X_j$  represents the addition of  $\delta X_j$  to  $X_j$  while keeping the other  $X_m$ ,  $m \neq j$ , unchanged. When  $\delta X_j$  is small, this cordslope is a good approximation to the tangent slope  $\partial R_i / \partial X_j$ . It is important to note that the task of calculating the Jacobian derivatives has been reduced now to a series of residual calculations, which was described in Table F.1.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	•	x		x											
2	x	•	x		x										
3	x	•			x										
4	x		•	x		x									
5	x		x	•	x		x								
6		x		x	•			x							
7			x			•	x		x						
8			x		x	•	x		x						
9			x		x	•				x					
10				x			•	x			x				
11					x		x	•	x			x			
12						x		x	•				x		
13							x			•	x				
14								x		x	•	x			
15									x		x	•			

Figure F.1: Incidence matrices: naturally ordered 3x5 grid system

The quantity  $R_i(\mathbf{X})$  is called the unshifted residual, and  $\mathbf{X}$  are the unshifted primary variables. The intermediate unshifted properties are saved. When one variable  $X_j$  is shifted, only the properties which depend on that variable are recalculated, thus saving work. Also, storage and computer time are spent only on the grid blocks, wells and interblock connections actually being used.

Finally, increasingly sophisticated features of the model can affect the sparsity of the Jacobian matrix factors which tend to destroy the simple banded structure shown in Figure F.1. On a submatrix level, constraint equations (usually describing phase equilibrium situations) can be solved simultaneously with the flow, resulting in full diagonal submatrices of the Jacobian but zero rows of off diagonal submatrices. The use of adaptive implicit techniques result in variable numbers of equations and primary variables in Jacobian submatrices, depending on whether the grid block is implicit or explicit. Later sections of this chapter indicate when these factors arise. In addition, as described in the next chapter, the well equations can be fully coupled to the reservoir flow equations, yielding additional rows and columns at the bottom of the Jacobian matrix. Finally, more flexible grid design features (see chapter 10) - null blocks, ninepoint discretizations (and possibly dual porosity or dual permeability approaches) also affect the Jacobian sparsity. The necessity of a flexible scheme for storing nonzero Jacobian elements and a general sparse solver to invert the resulting Jacobian matrix is obvious.

---

## F.7 Solution of Linear Equations – General Sparse Solver

The application of Newton's method to the discretized equations of reservoir simulation results in a linear system of equations

$$Ax = b \quad (\text{F7.1})$$

where the matrix A is sparse. Techniques have been developed to efficiently solve these equations (invert the matrix A) in a unified and flexible manner.

The first is submatrix preprocessing (constraint equations and adaptive implicit reductions) to reduce the algebraic work in the factorization and forward/backward substitution steps. Also a subset of the equations of the Jacobian matrix can be preprocessed (well equations and the dual porosity "matrix" equations) to create a smaller Jacobian matrix. Finally the use of a red-black ordering scheme (a generalized D4 numbering system) can further eliminate approximately half of the unknowns in optimal situations.

Eventually a large, still sparse matrix remains to be inverted. For smaller systems, direct Gaussian elimination is preferred. For larger systems, an iterative scheme should be employed. Here an incomplete LU factorization (ILU) is used as a preconditioner, followed by a iterative solution acceleration procedure. The ILU preconditioning step is actually a few steps of Gauss elimination. The result is a flexible approach to solving the linear system of equations, allowing the iterative scheme to approach complete Gauss elimination as the accuracy of the preconditioning step is increased. Symbolic factorization is used to define the nonzero structure of the L and U factors of A. More complete descriptions of the concepts outlined here are available<sup>5,6</sup>. The user can control the solution method employed by appropriate data choices.

## F.8 Treatment of Solid Components

The conservation equation per gross volume of solid component i is

$$\frac{\partial}{\partial t} [\varphi_v c_i] = \sum_{k=1}^{n_r} (s_{ki} - s_{ki}) r_k \quad (\text{F8.1})$$

where

$\varphi_v$	is the void porosity (ratio of void volume to gross volume)
$c_i$	is the concentration of component i in void volume
$s_{ki}$	is the product stoichiometric coefficient of reaction k
$s_{ki}$	is the reactant stoichiometric coefficient of reaction k
$r_k$	is the rate of reaction k

This equation depends entirely on quantities local to the grid block, and so can be solved fully implicitly and simultaneously.

This treatment of solid concentration allows the model to advance timesteps large enough that  $c_i$  and  $\varphi_f$  change significantly.

It is not unusual for solid coke fuel to occupy 5 to 20 percent of the void pore volume.

In these cases, a very implicit and stable method is a requirement for the successful calculation of  $c_i$  and  $\varphi_f$ .

This treatment is complicated by the fact that the fluid  $\varphi_f$  (used to calculate  $r_k$ ) is a function of solid concentration  $c_i$ .

The following are the general steps taken in calculating the reaction rate  $r_k$  and new solid concentration  $c_i$ .

1. Evaluate

$$\varphi_v = \varphi^0 [1 + a(p - p^0) - b(T - T_r)] \quad (\text{F8.2})$$

where

a	is the formation compressibility,
b	is the formation thermal expansion coefficient
$\varphi^0$	is porosity at porosity reference pressure $p^0$ , and
p and T	are the most recent values of pressure and temperature (can be differentiating variables)

2. Evaluate

$$\varphi_f^N = \varphi_v \cdot \left( 1 - \sum_{i=1}^N \frac{c_i^N}{\rho_{si}} \right) \quad (\text{F8.3})$$

where  $\rho_{si}$  is the mole density of component i in the solid phase. At this point  $\varphi_f^N$  is a combination of most recent p and T and N-level  $c_i^N$ .

3. Replace the time derivative with a mass conserving discretization

$$\frac{\partial}{\partial t} [\varphi_v c_i] \rightarrow \frac{\varphi_v c_i - \varphi_v^N c_i^N}{\Delta t} \quad (F8.4)$$

and solve the nonlinear equation

$$R_i = \frac{\varphi_v c_i - \varphi_v^N c_i^N}{\Delta t} + \sum_{k=1}^{n_r} (s'_{ki} - s_{ki}) r_k = 0 \quad (F8.5)$$

Rate  $r_k$  can be split into a product of two parts. The first part  $r_k^*$  does not depend on any concentrations, and contains not only the fluid concentration factors but also the void porosity factors  $\varphi_v^{ekj}$  for any solid concentration factors in the reaction rate. The second part consists of all the solid concentration factors  $c_j^{ekj}$  without the associated porosities.

4. Solid component equations are solved in a way that minimizes the number of equations that must be solved simultaneously. Each set of equations that must be solved simultaneously is called a solid set. The solid sets are solved in an order such that the resulting reordered matrix

$$\partial R_i / \partial c_j$$

is as tri-diagonal as possible.

Example A: Solid #1 equation depends on  $c_1$  and  $c_2$  but solid #2 equation depends only on  $c_2$ . We have two solid sets: in the first, solid #2 equation is solved for  $c_2$ ; and in the second, solid #1 equation is solved for  $c_1$  ( $c_2$  is known).

Example B: Both solid #1 and #2 equations depend on  $c_1$  and  $c_2$ . We have one solid set, consisting of the simultaneous solution of the two equations.

Perform (5) for each solid set.

5. The solution of equations  $R_i = 0$  for each solid set is accomplished via Newton's method. If all the equations in the solid set are linear with respect to all the variables  $c_j$  in the solid set, then only one Newton iteration is done.

In the special case where the equation for solid  $i$  depends only on  $c_i$ , and all the  $ek_i = 1$ , the equation is linear and the solution is

$$c_i = \frac{\varphi_v^N c_i^N + \Delta t \sum_{k=1}^{n_r} s'_{ki} \bullet r_k^*}{\varphi_v + \Delta t \sum_{k=1}^{n_r} s_{ki} \bullet r_k^*} \quad (F8.6)$$

Note that the effect of  $s'_{ki}$  is to increase  $c_i$ , whereas the effect of  $s_{ki}$  is to decrease  $c_i$ .

Complete the calculation of  $r_k$  for each reaction  $k$

$$r_k = r_k^* \bullet \prod_j c_j^{ekj} \quad (F8.7)$$

6. Make adjustments that prevent unphysical (negative) fluid porosity while preserving the stoichiometric relationships between components. Quantity

$$F = 0.8 * \left[ 1 - F_{\text{fluid}} - \left( \phi_v^N / \phi_v \right) \sum_i \left( c_i^N / \rho_{si} \right) \right] \quad F8.8$$

is the maximum available incremental void volume fraction, from the beginning of the timestep.  $F_{\text{fluid}}$  is the fraction of void volume occupied by fluid components (e.g., by adsorption) and is considered a known constant in this calculation. The void porosity ratio accounts for changes in pressure and temperature effects (e.g., the void pore space may decrease). Fraction 0.8 allows fluid porosity to approach 0 asymptotically by discounting the maximum value (at which fluid porosity is zero) by 20%.

All the  $c_i$  were solved from F8.5 in step #5. In fact, a highly nonlinear equation (e.g., with large  $e_{kj}$ ) may not have been solved to within the convergence criterion. In either case, these  $c_i$  do not account for quantity  $F$ . Quantity

$$G = \sum \left( c_i / \rho_{si} \right) - \left( \phi_v^N / \phi_v \right) \sum \left( c_i^N / \rho_{si} \right) \quad F8.9$$

is the incremental void volume fraction corresponding to the unadjusted  $c_i$ . If  $G > F$  then the adjustment fraction is  $X = F/G$ . All solid concentrations are adjusted as

$$c_i^* = X \cdot c_i + (1 - X) \cdot \left( \phi_v^N / \phi_v \right) c_i^N \quad F8.10$$

The resulting incremental void volume fraction is  $F$ , and reaction stoichiometry is preserved. Finally, the rates of all reactions involving solid components are multiplied by  $X$ . The only thing not preserved is the reaction rate form given by F8.7.

If fraction  $F$  is less than  $10^{-3}$  then it is set to zero, effectively maintaining a minimum fluid porosity ratio of about  $10^{-3}$ . In this case all reactions involving a solid may be shut off completely. This algorithm allows those reactions to start up again if conditions cause the required volume to become available. For example, coke lay-down may proceed in the absence of oxygen until  $F < 10^{-3}$  causing that reaction to stop; then, the introduction of oxygen will cause coke to decrease which will let the lay-down reaction proceed again.

Note that this adjustment handles cases where void and solid volumes change even though all solid concentrations remain unchanged. This can happen when the rock and solid phase have significantly different compressibility or thermal expansion coefficients, or when an advanced porosity option is used (e.g., dilation or geomechanics).

7. Using the new solid concentrations  $c_i$ , update the fluid porosity

$$\phi_f = \phi_v \left[ 1 - F_{\text{fluid}} - \sum_i \left( c_i / \rho_{si} \right) \right] \quad (F8.11)$$

for further use in evaluating accumulation terms, etc.

---

## F.9 Adaptive-Implicit Method

The Adaptive-Implicit Method (AIM)<sup>7</sup> provides various degrees of implicitness for different grid blocks, which translates into CPU saving in comparison with a fully implicit method<sup>1</sup>. Blocks that are experiencing large throughputs or rapid changes in primary variables (pressure, saturation, temperature, etc.) are treated fully implicitly. Blocks in which the changes or throughput are small are handled in an "IMPES" fashion.

In an "IMPES" treated block the following quantities are not fully implicit:

1. Transmissibilities
2. Capillary pressure
3. Gravity head
4. Thermal conductivity

These are explicit (from previous timestep), when there is not a phase change during a particular timestep. As soon as one of the phases appears or disappears, the above quantities are updated during the Newton iteration and kept on this iteration level for the rest of the timestep. Such updates are necessary due to the correct upstream values.

The choice of primary variables as well as the solution of the mass and energy balance equations is the same as in the fully implicit method. However, the Jacobian matrix is different.

The structure of a Jacobian matrix for a hypothetical reservoir with 3 grid blocks containing water and dead oil is shown in Figure F.2.

IMPES	IMPES	IMPES
$T^m$ X X X	$T^m$	
$T^m$ X X X	$T^m$	
$T^m$ X X X	$T^m$	
X X X X		
$T^m$	$T^m$ X X X	X X X X
$T^m$	$T^m$ X X X	X X X X
$T^m$	$T^m$ X X X	X X X X
	X X X X	
	$T^m$	X X X X
	$T^m$	X X X X
	$T^m$	X X X X
	$T^m$	X X X X

$T^m$  is an analytical flow derivative with respect to pressure

- Newton's iteration without phase change
  - $m = N$  until phase change
  - $m = N + 1, k^*$  after phase change ( $k^*$  is iteration at which phase change occurred)
- Newton's iteration with phase change
  - $m = N + 1, k + 1$

*Figure F.2: Example of Jacobian matrix*

Two alternatives are available for the switching between fully implicit and "IMPES" treatment

1. Threshold switching criterion is based on a specific threshold change of each primary variable over a timestep. The magnitude of the threshold values is problem dependent and must be supplied by the user. High values may cause insufficient switching of "IMPES" blocks to implicit ones which creates instability problems. Low values cause premature switching and therefore less CPU savings.
2. Stability switching criterion is based on the numerical stability of a local amplification matrix

$$A = \left[ \frac{\partial f_j^{N+1,k+1}}{\partial v_i^{N+1,k+1}} \right]^{-1} \left[ \frac{\partial f_j^{N+1,k+1}}{\partial v_i^m} \right] \quad (F9.1)$$

$f_j^{N+1,k+1}$  - residual of jth conservation equation in the k+1 Newton's iteration of N+1 timestep

$v_i^{N+1,k+1}$  - primary variable i in the k+1 Newton's iteration of N+1 timestep

$v_i^m$  - primary variable i of the mth Newton's iteration level (see explanation in Figure F.2)

All eigenvalues of this matrix must be  $\leq 0$ , so that the velocity of the fastest moving front satisfies the stability condition

$$\left( u \frac{dt}{dx} \right)_{v_i} \leq 1 \quad (F9.2)$$

where  $v_i$  is the primary variable associated with the fastest moving front.

This criterion is problem independent and provides a possibility of backward switching (implicit to "IMPES" mode) in blocks encountering little changes in pressure, temperature, etc.

	FIM Iterations	AIM Iterations	CPU AIM/FIM
SPE 1 (Gauss)	506	553	0.56
SPE 2 (Gauss)	408	439	0.53
SPE 2 (1st ILU)	370	461	0.91
SPE 3 (1st ILU)	408	494	0.90
Steam + 2 additives (Gauss)	397	389	0.39
Field cycling (2nd ILU)	3843	3793	0.59
Field combustion (2nd ILU)	2449	2579	0.67

*Figure F.3: Examples of adaptive implicit savings*

## F.10 Use of Constraint Equations in the S<sub>x</sub>y Formulation

The constraint equations are used to calculate all the remaining saturations and mole fractions from the primary variables. The procedure depends very much on which primary variables are being used. For the purpose of illustration, consider a fluid model consisting of water, two oils, air and coke. Table F.2 contains the resulting equation set.

Equation No.	Equation
1	water conservation
2	heavy oil conservation
3	light oil conservation ( $x_3 < x_2$ )
4	Noncondensable nitrogen conservation
5	noncondensable oxygen conservation
6	energy conservation
7	saturation constraint
8	phase equilibrium constraint

Table F.2: Sample equation set #2

In most cases before air is injected, p, T, S<sub>w</sub>, S<sub>o</sub> and x<sub>3</sub> will be primary variables, and the constraints are used as follows.

$$\begin{aligned}x_2 &= 1 - x_3 \\y_2 &= x_2 \cdot K_2(p, T) \\y_3 &= x_3 \cdot K_3(p, T) \\y_1 &= K_{w1}(p, T)\end{aligned}\quad (\text{F10.1})$$

When gas is absent (wide boiling system) then

$$\begin{aligned}S_g &= 0, \\R_7 &= 1 - S_w - S_o - S_g, \text{ and}\end{aligned}\quad (\text{F10.2})$$

$1 - y_1 - y_2 - y_3 > 0$  must be true.

When gas is present (narrow-boiling system) then

$$\begin{aligned}S_g &= 1 - S_w - S_o, \\R_g &= 1 - y_1 - y_2 - y_3, \text{ and}\end{aligned}\quad (\text{F10.3})$$

$S_g > 0$  must be true.

In each case, one of the phase constraints is satisfied as an equality, and the other constraint is satisfied as an inequality.

### Rules for Choosing Primary Variables

1. Pressure p is always aligned with water, for compatibility with adaptive implicit options.
2. Of the other condensable components which make up the oil phase, oil mole fraction x<sub>i</sub> is used.

3. The component with the largest value of  $x_i$  has  $S_o$  aligned with its equation. This avoids aligning  $S_o$  with an equation whose component has  $x_i = 0$ .
4. When the oil phase has superheated, each  $x_i$  and  $S_o$  is replaced with  $y_i$ . If the component has  $K_i = 0$ , then use  $S_{oxi}$  instead. Primary variable  $S_{oxi}$  is the symbolic product of  $S_o$  (which is 0) and  $x_i$ . Use of  $S_{oxi}$  avoids a bad Jacobian matrix.
5. Conservation equations of noncondensable gases are aligned with the primary variable  $S_{gyi}$ , which is the symbolic product of  $S_g$  and  $y_i$ . Its use avoids a bad Jacobian matrix when  $S_g = 0$ . It makes treatment of all such components the same, and allows the same treatment to be used for steam and combustion.
6. Energy conservation is aligned with temperature, except when  $T$  must go somewhere else.
7. The saturation constraint is used for wide-boiling systems, and is aligned with  $S_w$ .
8. The gas mole fraction constraint is used for narrow-boiling systems, and is aligned with  $T$ .
9. When the water phase superheats,  $y_1$  is aligned with the gas mole fraction constraint.

### Use of Constraint Equations

Tables F.3, F.4 and F.5 show the procedure used in each case of phase absence or presence. The component set from Table F.2 is used. Note the  $K_{w3}$  controls the amount of component 3 in the aqueous phase. Also, it is assumed that  $K_2$  and  $K_3$  are never zero. The constraints that are built in Table F.5 are solved simultaneously with the conservation equations.

	$S_o > 0$	$S_g = 0$
Use:	$x_2 = 1 - x_3$ $y_2 = x_2 \bullet K_2 (p, T)$ $y_3 = x_3 \bullet K_3 (p, T)$	$S_o = 0$ $x_2 = y_2/K_2 (p, T)$ $x_3 = y_3/K_3 (p, T)$
Check for:	$S_o < 0$	$x_2 + x_3 > 1$

*Table F.3: Treatment of oil components 2 and 3*

	$S_w > 0$	$S_w = 0$
Use:	$w_3 = y_3/K_{w3} (p, T)$ $w_1 = 1 - w_3$ $y_1 = w_1 \bullet K_{w1} (p, T)$	$w_3 = y_3/K_{w3} (p, T)$ $w_1 = y_1/K_{w1} (p, T)$ $S_w = 0$
Check for:	$S_w < 0$	$w_1 + w_3 > 1$

*Table F.4: Treatment of water component I*

Let $A = y_1 + y_2 + y_3$		
	Narrow Boiling $A > \frac{1}{2}$ and $S_g > 0$	Wide Boiling $A < \frac{1}{2}$ or $S_g = 0$
Use:	$S_g = 1 - S_w - S_o$ $y_4 = S_g y_4 / S_g$ $y_5 = S_g y_5 / S_g$	$S_g = (S_g y_4 + S_g y_5) / (1 - A)$ $y_4 = S_g y_4 / S_g$ $y_5 = S_g y_5 / S_g$
Check for:	$S_g < 0$	$A > 1$ when $S_g = 0$
Satisfies:	$S_w + S_o + S_g = 1$	$\sum_{i=1}^6 y_i = 1$
Residual:	$R = 1 - \sum_{i=1}^6 y_i$	$R = 1 - S_w - S_o - S_g$

Table F.5: Treatment of noncondensable components 4 and 5

## References

1. Rubin, B., and Buchanan, W.L., "A General Purpose Thermal Model," SPEJ, Vol. 25, No. 2, pp. 202-214.
2. Coats, K.H., and Modine, A.D., "A Consistent Method for Calculating Transmissibilities in Nine-Point Difference Equations," paper SPE 12248, presented at the SPE Reservoir Simulation Symposium, San Francisco, California, November 1983.
3. Trimble, R.H., and McDonald, A.E., "A Strongly Coupled, Fully Implicit, Three Dimensional, Three Phase Well Coning Model," SPEJ, August 1981, pp. 454.
4. Aziz, K., and Settari, A., Petroleum Reservoir Simulation, Applied Science Publishers, 1979.
5. Nghiem, L., and Rozon, B., "A Unified and Flexible Approach for Handling and Solving Large Systems of Equations in Reservoir Simulation," Proceedings of the International Forum on Reservoir Simulation, Alpach, Austria, September 1988.
6. AIMSOL Technical Manual - Sparse Matrix Solver for Advanced Reservoir Simulation Technology, Version 3.3, November 1990.
7. Oballa, V., Coombe, D.A., and Buchanan, W.L., "Adaptive Implicit Method in Thermal Simulation," paper SPE 18767, presented at the 1989 SPE California Regional Meeting, Bakersfield, California, April 5-7, 1989.



# Appendix G: Electrical Heating

---

## Overview

This appendix is organized in the following manner:

- G.1 Brief Description of Theory
- G.2 Mathematical Model Used by STARS
- G.3 Reports and Plots
- G.4 Templates
- G.5 Input Data
- G.6 References

## G.1 Brief Description of Theory

This description is a brief summary based entirely on chapter 2 of Hiebert's thesis<sup>1</sup>. Please consult the references in Hiebert for a more complete review of the subject.

### Simplifying Assumptions

In theory, the full form of Maxwell's equations, along with appropriate boundary conditions, may be solved to find the electric field for any physical configuration at any frequency. The following simplifying assumptions are used here to make the computations more tractable:

1. For **anisotropic electrical conductivity**, the principal axes of the conductivity tensor are parallel to the coordinate axes. Anisotropic conductivities arise when thin silts or shales are intermixed with oil-bearing matrix. With horizontal bedding, for example, current flows through the different materials in parallel in each horizontal direction but in series in the vertical direction. The conductivity tensor assumption above is usually preferable to the expense of modelling each individual material bed with a separate grid block layer.
2. The electrical properties (conductivity, permittivity and magnetic permeability) do not depend on the strengths of the electric or magnetic fields.
3. The **quasi-static approximation** is used, whereby the smallest wavelength resulting from application of a single-frequency potential is much larger than the largest physical length in the reservoir. This allows us to neglect electric fields that are produced by changing magnetic fields. The wavelength of a 60 Hz field will depend on the electrical conductivity of the formations surrounding the electrodes. The table below gives the calculated conductivity and a recommended maximum feature size to use, based on equation G.4 for a typical value of water conductivity (0.8 siemens/m). Even if you need to somewhat exceed the safe feature size, the quasi-static approximation can be used to provide a first estimate for design purposes.

Porosity	Sw	Conductivity	Resistivity	Wavelength	Recommended max. feature size (m)
0.3	1	0.175	5.72	998	100
0.2	1	0.100	9.98	1317	132
0.1	1	0.039	25.79	2118	212
0.05	1	0.015	66.65	3404	340
0.3	0.5	0.044	22.90	1995	200
0.2	0.5	0.025	39.91	2634	263
0.1	0.5	0.010	103.15	4235	424
0.05	0.5	0.004	266.60	6809	681
0.3	0.3	0.016	63.61	3326	333
0.2	0.3	0.009	110.85	4390	439
0.1	0.3	0.003	286.52	7058	706
0.05	0.3	0.001	740.56	11348	1135
0.3	0.2	0.007	143.11	4989	499
0.2	0.2	0.004	249.41	6586	659
0.1	0.2	0.002	644.66	10588	1059
0.05	0.2	0.001	1666.26	17022	1702

If you plan to operate at an electrical frequency higher than 60 Hz, you should calculate the wavelength in the formation at the operating frequency and apply the same criteria of a maximum feature size of about one tenth of a wavelength.

4. We may **neglect displacement current** if the potential frequency used is low enough. This allows us to express the electric field in terms of a scalar electrical potential. For oil sand this assumption is well justified below a potential frequency of 1 MHz, and so is valid at 60 Hz. In general the electrical potential may be a phasor, with real and imaginary parts.

### Current Continuity Equation

With the above assumptions and Ohm's Law, the electric potential may be solved from the current conservation equation

$$\nabla \bullet (\sigma \nabla \psi) = q \quad (G.1a)$$

The electric potential phasor  $\psi$  is  $\psi_R + j\psi_I$  where  $\psi_R$  and  $\psi_I$  vary in space and  $j^2 = -1$ . The electrical source term phasor  $q$  is  $q_R + jq_I$ . The electrical conductivity  $\sigma$  is a diagonal tensor, with no imaginary components since the formation has no capacitance or inductance. The differential operator is real, so equation G.1a can be decoupled into two separate equations.

$$\nabla \bullet (\sigma \nabla \psi_R) = q_R \quad (G.1b)$$

$$\nabla \bullet (\sigma \nabla \psi_I) = q_I \quad (G.1c)$$

In the case of a three-dimensional Cartesian grid, the equation for real components is

$$\frac{\partial}{\partial x} \left[ \sigma_x \frac{\partial \psi_R}{\partial x} \right] + \frac{\partial}{\partial y} \left[ \sigma_y \frac{\partial \psi_R}{\partial y} \right] + \frac{\partial}{\partial z} \left[ \sigma_z \frac{\partial \psi_R}{\partial z} \right] = q_R \quad (G.2a)$$

There is a similar equation for the imaginary components. Note that  $\sigma$  depends on temperature and phase saturations, so that  $\sigma$  and  $\psi$  may vary slowly with time. This dependence couples the current equations and hence the electrical potential to the fluid and reservoir conditions.

If all source terms have  $q_I = 0$  the result is  $\psi_I = 0$  everywhere, in which case equation G.1c does not need to be solved. This applies to non-alternating as well as single-phase alternating cases. For alternating current cases  $\psi$  is the rms (root-mean-square) potential, allowing the same formulas to be used for both direct and alternating cases.

### Heat Generation from Ohmic Losses

The heating rate due to electrical conduction is

$$Q = \sigma_x \left[ \frac{\partial \psi_R}{\partial x} \right]^2 + \sigma_y \left[ \frac{\partial \psi_R}{\partial y} \right]^2 + \sigma_z \left[ \frac{\partial \psi_R}{\partial z} \right]^2 + \sigma_x \left[ \frac{\partial \psi_I}{\partial x} \right]^2 + \sigma_y \left[ \frac{\partial \psi_I}{\partial y} \right]^2 + \sigma_z \left[ \frac{\partial \psi_I}{\partial z} \right]^2 \quad (G.3)$$

This heating rate couples the fluid and reservoir conditions to the electrical potential.

## G.2 Mathematical Model Used by STARS

### Electrical Conductivity

The user enters temperature-dependent electrical conductivity for the water phase  $\sigma_{w,p}(T)$ , oil phase  $\sigma_{o,p}(T)$ , solid phase  $\sigma_{s,p}(T)$  and rock/matrix  $\sigma_{r,p}(T)$  for each grid block in three directions ( $p = i,j,k$ ). The water phase value has possible composition dependence

$$\sigma_{w,p}(T) = \sum_i w_i \cdot \sigma_{w,i,p}(T)$$

where  $w_i$  is water mole fraction and  $\sigma_{w,i,p}(T)$  is electrical conductivity, both of aqueous component  $i$ . The water phase value has an additional dependence on fluid porosity  $\varphi_f$  and water saturation  $S_w$  from the Archie equation, for example,

$$\sigma_{wp}(T, \varphi_f, S_w) = \sigma_{w,p}(T) \cdot [\varphi_f^{1.37} S_w^2 / 0.88] \quad p = i, j, k$$

The solid phase value has possible composition dependence

$$\sigma_{s,p}(T) = \sum_i (c_i/c_s) \cdot \sigma_{s,i,p}(T)$$

where  $c_i$  is concentration and  $\sigma_{s,i,p}(T)$  is electrical conductivity, both of solid component  $i$ , and  $c_s$  is the sum of all the  $c_i$ .

The bulk electrical conductivity is obtained from volume-weighted averaging

$$\sigma_p = \sigma_{w,p}(T, \varphi_f, S_w) + \sigma_{r,p}(T) \cdot (1 - \varphi_v) + \sigma_{s,p}(T) \cdot (\varphi_v - \varphi_f) + \sigma_{o,p}(T) \cdot \varphi_f \cdot S_o \quad (G.4)$$

where the porosities are defined in Appendix F.2. Note that water conductivity  $\sigma_{w,p}(T, \varphi_f, S_w)$  already contains the factor  $\varphi_f \cdot S_w$ . Different values of the conductivities and Archie parameters may be specified for each rock zone.

### Current Conservation Equation

Consider current flow between block  $i$  and block  $i+1$ , separated electrically by two resistances in series. From block center  $i$  to common block face  $i+1/2$  the geometric factor (separation divided by cross-sectional area) is  $T_i$  and the bulk conductivity is  $\sigma_i$ , so the resistance is  $R_i = T_i / \sigma_i$ .

Similarly for block  $i+1$ ,  $R_{i+1} = T_{i+1} / \sigma_{i+1}$ . The current flow from block center  $i$  to block center  $i+1$  is potential drop over resistance in series

$$I_{i,i+1} = (V_i - V_{i+1}) / (R_i + R_{i+1}) \quad (G.5)$$

Therefore, application of equation G.1 to a grid block amounts to constraining to zero the sum of the current flow terms like G.5 between that block and all its neighbours. The current equations for all the blocks are solved simultaneously along with the fluid flow conservation equations. Therefore, when the timestep is converged the resulting potential field reflects the newest reservoir conditions. The error in current material balance usually is very small.

The above applies only to internal block faces, that is, faces between two grid blocks. For external block faces there is no current flow other than specified electric boundaries.

## Boundary Conditions

An electrical boundary is a collection of block *faces* that are assumed to be at the same potential, and through which current flows into and out of the reservoir. Since a grid block's potential is referenced to the geometrical *center* of the block, there is a potential drop between a segment of electrical boundary and its host grid block. A boundary face may be external or internal to the grid.

Each host block has an additional current term similar to G.5, corresponding to the electrical boundary, with a similar definition for geometric term and resistance. The (real or imaginary) current flow from block center  $i$  to boundary  $b$  is

$$I_{i,b} = (V_i - V_b) / R_i \quad (G.6)$$

The current of each boundary segment is saved for purposes of reporting as well as detection and control of a maximum-current type constraint.

## Phase Modes

There are two possible simulation modes with respect to AC phasing. When at least one boundary has an imaginary ( $j$  component) source term, both the real and imaginary components of potential and current are calculated. In this case the model is in ***multi-phase mode*** since the imaginary components can model boundaries of different phases. On the other hand, when there are no imaginary source terms then the  $j$  component equation G.1c is not solved and the run is in ***single-phase mode***. A case containing only real potentials of both signs together can be handled by single-phase mode.

## Heat Generation

The heat generation rate in a grid block is the sum of the rates for all the currents flowing in that block. The real current obtained in G.5 for inter-block flow contributes to the heating rate  $Q$  in two grid blocks

$$Q_i = (I_{i,i+1})^2 * R_i \quad Q_{i+1} = (I_{i,i+1})^2 * R_{i+1} \quad (G.7)$$

and the boundary-block flow in G.6 also contributes to the heating rate

$$Q_i = (I_{i,b})^2 * R_i \quad (G.8)$$

There are similar contributions for imaginary current components.

To maximize convergence stability, the electrical heating rate is kept constant after a specified Newton iteration. Typically this means that a timestep's heating rate is based on the potential field resulting from the previous timestep. However, when a boundary constraint is changed the new value is felt immediately in the heating rate after one fluid/heat flow iteration that establishes the new voltage field.

## Electrical Operating Constraints

There are four types of operating constraint: maximum potential, maximum current, maximum total heating rate and maximum no-flash heating rate. Each boundary must have an initial potential assigned to it, but current, total heat rate and no-flash heat rate constraints are optional.

All boundaries start operating with their (mandatory) potential constraints, providing after one iteration a base potential field against which all other constraints types are tested. This constraint type treats  $V_b$  in equation G.6 as known and calculates the resulting  $I_{i,b}$ .

Before it becomes the operating constraint, a current-type constraint is tested by comparing its value against the magnitude of the boundary's summed  $I_{i,b}$ ; when the value exceeds the specified maximum, that boundary is switched to operate on its current type constraint. A current constraint distributes the specified current amongst the boundary's faces according to weighting factors  $(V_i - V_b)/R_i$  and then treats  $I_{i,b}$  in equation G.6 as known. These weighting factors depend on the last updated (and possibly lagged) resistances and potential field, but the specified total current magnitude for that boundary is honoured. When a current constraint is in effect, the calculated  $V_b$  is tested against that boundary's potential constraint. Since a current type constraint operates by adjusting  $V_b$  as a ratio of its initial value and phase, this constraint type cannot be applied to a boundary with  $V_b = 0$ . Also, a boundary's current type constraint operates best when most or all of the layers contribute current of the same sign, that is, the boundary's potential is either higher or lower than surrounding values.

The total heat rate (power) and no-flash constraints are global constraint types which can override any other constraint. Each global constraint operates by adjusting the entire potential field by a scalar factor. This technique is possible because equation G.1 lacks an accumulation term and resistances do not depend upon potentials, so multiplying all potentials by factor  $x$  results in local (and hence total) heating rates multiplied by  $x^2$ . This adjustment is done at the end of a timestep as well as whenever the heating rate is updated. Before a global constraint becomes the operating constraint, heating rates are calculated from the existing potential field and scale factors are obtained. If the scale factor is less than one, that is, the global constraint is more restrictive, operation is switched to that constraint and the scale factor is applied to the entire potential field. A boundary with zero potential can be used as a reference (ground) since its potential does not change when the factor is applied.

Combining the algorithms described above results in automatic switching between constraints depending on changing conditions, so that the most restrictive constraint is used. For example, it is common to specify maximum potentials and total heating rate. At the start the process may be running on maximum potentials, but the heating rate increases as the reservoir heats and conductivity increases. When the maximum total heating rate is reached, non-zero boundary potentials are decreased so that the total heating rate is equal to the maximum specified.

---

## G.3 Reports and Plots

### Text Output File

1. A **conductivity definition summary** is echoed along with the fluid properties. Included are Archie coefficients and temperature multiplier for each electrical rock type, as well as the (possibly composition dependent) phase and rock conductivities for all grid blocks in all directions.
2. An **electrical boundary definition summary** is echoed along with the fluid wells. Included are boundary name, number and direction, as well as the list of host grid blocks and the corresponding geometrical factor. This geometrical factor correctly accounts for flow from the host block center to the block face in the indicated direction. Also echoed are the various operating constraints, and the location of each boundary number on a grid map.
3. **Special history definitions** are echoed.
4. The **material balance report** includes heat generated by current flow, in the fluid energy units (Joules or Btu).
5. The following **per-block electrical results** are available at the end of the timestep, enabled via subkeywords of \*OUTPRN \*GRID
  - Bulk electrical conductivity in siemens per meter, in three directions if anisotropic.
  - Electric potential in V (real, imaginary, magnitude and phase).
  - Electric power (heat generation rate) in kW.
  - Electric power per bulk volume in kW per user volume unit.
  - Cumulative electric heat in kW-hr.
  - Magnitude of the electrical current vector
6. The following **electric boundary results** are available at the end of the timestep:
  - *Operating conditions*, with electrical potential (real, imaginary, magnitude in V, phase in degrees), current magnitude in A, and operating constraint in effect.
  - *Current conservation*, with real, imaginary and magnitude of both current in A and accumulated flow in A times user's time unit. For perfect convergence real and imaginary currents will sum to zero, so the current and cum flow "Total" values indicate conservation errors. The current magnitude value of each boundary is well defined, and the total of all boundaries represents twice the current flow in the system since current is counted twice, for "in" and "out". The Cum Flow magnitude is the integration of current magnitude over time, a quantity which is not very meaningful physically but provides a scale to compute relative balance error. Finally, "Cum Flow Balance Error (%)" is 100 times the square root of the sum of the squares of the real and imaginary entries of the Cum Flow Totals, divided by one half of the Cum Flow "Mag" Total.

- Rate in kW and accumulation in kW-hr of electrical heating summed over the grid.
  - *Layer summaries*, with complex Current in A, accumulated flow in A times the user's time unit, potential difference between boundary and adjacent block and block conductivity, for individual boundary layers. Use these quantities to identify why each layer and boundary is behaving as it is. Note that in multi-phase mode a boundary has a single phase value determined by the previous \*ELTARGET \*POTENTIAL specification, but current flows in its various layers may have different phase values. A \*CURRENT constraint effects only the magnitude of the layer sum of those complex currents, which is why this quantity appears in the *Operating conditions* report.
7. The following **sector** statistics are available: electrical heating rate and accumulation.

### Graphical Display in RESULTS

In RESULTS you have a choice of units for display of electrical quantities. For example, electrical potential may be displayed in V, kV or mV.

1. The following **per-block electrical results** are available via the subkeywords of \*OUTSRF \*GRID:
  - Those per-block quantities available in the text file.
  - Bulk electrical conductivity in individual directions, if anisotropic.
  - Vector plots of current density and current. Current density is recommended since it is independent of grid block size.
2. **Electrical boundary results** are available as special histories: potential, current and accumulated charge for both boundaries and individual boundary segments.
3. Other **special histories** are available:
  - Rate and accumulation of electrical heat generated Total grid
  - Resistance between two boundaries, meaningful only if there are at most two different boundary potentials.
  - Potential gradient between two grid blocks
  - Quantity \*Y for the water component is vapour pressure divided by total pressure, and so is a good indicator of how close a block is to the bubble point.
  - Any quantity (except vector plot) available via \*OUTSRF \*GRID is available for a single grid block via subkeyword \*BLOCKVAR, as well as subkeywords \*MINVAR, \*MAXVAR and \*AVGVAR.
4. The following **sector** statistics are available: electrical heating rate and accumulation.

---

## G.4 Templates

Most of these templates have extensive output enabled to both the text output and SR2, and some have the current density field available as a vector plot in RESULTS 3D. All have a significant list of special histories, for example, total heat rate and accumulation, and boundary potentials, currents and accumulated charge.

### **ELEC1: Constraint Changes, \*ELWCOMPTAB and \*ELSCOMPTAB**

This template has recurrent data in which the downhole electrode potential changes several times during the run. Also, the list of blocks associated with the electrode is changed several times. You can see in a cross-section view of the block electric potential in RESULTS that the change in electrode block assignments changes the electric field shape near the well.

This template also has composition-dependent electrical conductivities for both the water and solid phases. There are two water components: original in-place water and injected water with enhanced conductivity. Two different solid components are found coating the rock matrix in separate regions of the reservoir. Note that you need a chemical reaction present in order to satisfy mass conservation of the solid components, but the reaction has a zero rate.

### **ELEC2: Electrical Heating with 3D Cartesian Grid**

In this 13\*10\*4 Cartesian grid the electrode is at one corner and the ground is at an opposite edge, and sector output allows you to track the heat generated in the 4 layers separately.

### **ELEC3: Metal Electrode and Constraint Switching**

This template models primary production with a radial grid. The electrode interval at the wellbore changes with time, and ground is the top of the overburden. The innermost radial block is used to model a metal electrode. The 10-ft pay zone, the 67-ft overburden and the metal electrode each have a different set of electrical properties. The water electrical conductivity varies with temperature. Electrode potential target starts at 220 V but is immediately cut back due to the 5 kW maximum total power constraint. Later the potential is manually decreased in steps to 120 V so that the power falls below its maximum. This can be seen by viewing the total power and boundary potentials in RESULTS Graph.

The change of conductivity can be viewed via a plot of total resistance between the electrode and ground. Some potential gradients also are written to the SR2.

### **ELEC4: No-Flash Option**

This template is similar to ELEC3 but the fluid flow is larger and the No-Flash option is enabled. The data requests that the temperature in any grid block not exceed that block's water flashing temperature minus 50 degrees.

The electrode starts on 220 V, but hits the 20 kW power limit as the region around the wellbore heats and the electrical conductivity increases. As blocks near the wellbore approach the water flashing temperature, the no-flash constraint kicks in and reduces the power even further. In the last half of the run the temperature and power are very steady, indicating the pseudo-steady process of in-flowing fluid cooling the wellbore. This can be seen very well by plotting with RESULTS Graph the total power together with temperature of block (1,1,6).

## **ELEC5: Hybrid Grid**

The regions around two wells in a Cartesian grid are modelled with hybrid grids. The run starts on maximum potential constraint, switches to maximum total power constraint and then to maximum current, as shown by the corresponding special histories.

## **ELEC6: Multi-Phases in Single-Phase Segments**

This template is a single-well case that uses different phases in each of the single-phase time segments. This template changes the phase angle between time segments in order to verify the correct handling of multi-phase current as compared to single-phase current. The phase schedule is

0 – 1 days	220 V at 90 deg
1 – 2 days	200 V at 60 deg
2 – 3 days	180 V at 150 deg
3 – 4 days	160 V at 240 deg
4 – 10 days	120 V at 330 deg

If this data is re-run in single-phase mode, the heating result will be the same.

## **ELEC7: Three-Phase Triangular Configuration**

This template tests and illustrates the use of a three-phase configuration for electrical heating. Three electrodes are placed at the vertices of an equilateral triangle, all with potentials at 220 V but differing in phase by 120 deg. Plots viewed in Results 3D show triangular symmetry for all "magnitude" results including heating, voltage and current density. Real and imaginary potentials and currents have some symmetry but not triangular symmetry.

The run starts on specified potentials, each electrode with its own phase. Later, the maximum power constraint is reached, after which the electrode current constraints are used.

Note that Electrode 1 with phase angle 0 has some imaginary component current during the \*POTENTIAL and \*POWER constraint operation, but it has no imaginary current component while it is on \*CURRENT constraint since the specified current uses the phase specified by the \*POTENTIAL constraint.

---

## G.5 Input Data

The electric heating option is enabled by keyword \*ELECHEAT in the Other Reservoir Properties data section. The electrical heating keywords are organized into the following groupings, one grouping per manual page, in the following data sections.

### Other Reservoir Properties

1. \*ELECHEAT enables the electrical heating option. \*ELECTYPE and \*ELTYPE access the property set (rock type) option. \*VOLTOL, \*VOLSHF and \*EHEATCYC control convergence.
2. \*ELCONTAB, \*ELWCOMPTAB and \*ELSCOMPTAB specify electrical conductivities that vary with set, temperature, phase and composition.
3. \*ECONDWI, \*ECONDWJ, \*ECONDWK and \*TEMMULT specify electrical conductivities that vary by block and temperature (obsolete).

### Well and Recurrent Data

4. \*ELBOUND and \*ELTARGET specify electrical boundary conditions and operating constraints.

### Input/Output Control

5. \*OUTPRN \*GRID subkeywords ELCONDUCT, etc., specify grid dump output to the “.out” text file.
6. \*OUTSRF \*GRID subkeywords ELCONDUCT, etc., specify grid dump output to the SR2 graphics file. In addition, \*OUTSRF \*SPECIAL subkeywords ELHEAT, etc., are available for history plots.
7. The EXPLANATION for keyword \*INUNIT documents the electrical units.

### **Restrictions**

The electric heating option may be used with any grid, component, rock property and fluid well configuration, with the following exceptions:

1. The nine-point, natural fracture and discretized wellbore grid options are not allowed.
2. You may not use the \*RW 0 option of keyword \*GRID \*RADIAL with an electrical boundary in the -I, -J or +J direction, since this would give a radius of 0 to the inner reservoir boundary normally associated with the wellbore.
3. A zero-porosity heat-conducting block conducts electrical current only if rock electrical conductivity is assigned a non-zero value.
4. The \*ISOTHERMAL formulation option is not allowed.
5. Adaptive-implicit (\*AIM) options are not recommended or supported.
6. Dynamic (\*DYNAGRID) and recurrent gridding options are not allowed.

---

## G.6 References

1. Hiebert, A.D., "Numerical Simulation of the Electrical Pre-heat and Steam Drive Bitumen Recovery Process for the Athabasca Oil Sands," Ph.D. Thesis, Dept. of Electrical Engineering, University of Alberta, 1986.
2. Killough, J.E., Gonzalez, J.A., "A Fully-Implicit Model for Electrically Enhanced Oil Recovery," SPE 15605, presented at the 61<sup>st</sup> Annual Technical Conference and Exhibition, New Orleans, Oct. 5-8, 1986.

# Keyword Index

## A

ADHEAT 951  
ADMAXT 483  
ADRT 483  
ADSCOMP 480  
ADSLANG 480  
ADSPHBLK 483  
ADSROCK 483  
ADSTABLE 480  
ADSTYPE 483  
AIM 542  
AIMSET 965  
ALFDR 287  
ALITHO 313  
ALL 79  
ALLELEM 72  
ALTER 816  
ANNULUS 71  
ANNULUSWAL 217, 975  
APPOR-METHOD 866  
AQCOMP 255  
AQFRCOMP 331  
AQFUNC 263  
AQGEOM 255  
AQLEAK 255  
AQMETHOD 255  
AQPROP 255  
AQSTAT 114  
AQUIFER 255  
AQVISC 255  
ASATUR 313  
ATORTU 313  
AUTOCOOLER 946  
AUTODRILL 757  
AUTOHEATER 946  
AVG 355  
AVISC 359

## B

BCDOMAIN 705  
BCOEF 606  
BHPDEPTH 812  
BHPGRAD 814  
BINARY\_DATA 80  
BIOTSCOEF 586  
BKRCGW 458  
BKROCW 458  
BKRWIRO 458  
BKRWR0 458  
BLOCKAGE 372  
BPCGMAX 458  
BPCWMAX 458  
BSGCON 458  
BSGR 458  
BSOIRG 458  
BSOIRW 458  
BSORG 458  
BSORW 458  
BSWCON 458  
BSWCRT 458  
BSWIRG 458  
BSWR 458  
BSWRG 458  
BVG 355  
BVISC 359

## C

CALIB\_POR 715  
CASEID 103  
CASING 217, 975  
CHECKONLY 102  
CHECKRB 552  
CIRCWELL 217, 975  
CMM 329  
COHESEXPN 622  
COHESHARD 622  
COHESION 586, 599, 606, 615  
COMMENT 62  
COMPACT\_VAR 287  
COMPNAME 319  
CON 73  
CONC\_SLD 504  
CONVERGE 525  
CONVERT-TO-CORNER-POINT  
    154  
COORD 179  
CORNERS 181  
CORNER-TOL 184  
CP 346

CPEPAC 279  
CPERMCOR 287  
CPG1 335  
CPG2 335  
CPG3 335  
CPG4 335  
CPL1 335  
CPL2 335  
CPL3 335  
CPL4 335  
CPOR 274  
CPORPD 274  
CPT 346  
CPTPOR 274  
CRB 287  
CRD 279  
CRM 287  
CRP 284  
CT1 346  
CT2 346  
CTD 279  
CTP 284  
CTPOR 274  
CTPPAC 279  
CVFANGLE 622  
CYC\_GROUP 822  
CYCPRT\_END 944  
CYCPRT\_START 944

## D

DATE 731  
DATUMDEPTH 505  
DEPTH 163  
DEPTH-TOP 170  
DFRICANGLE 606  
DGOC 496  
DI 157  
DIFFI\_GAS 471  
DIFFI\_OIL 471  
DIFFI\_WAT 471  
DIFFJ\_GAS 471  
DIFFJ\_OIL 471  
DIFFJ\_WAT 471  
DIFFK\_GAS 471  
DIFFK\_OIL 471  
DIFFK\_WAT 471  
DIFRAC 213  
DILANGLE 586  
DILATION 279  
DIM 99  
DIP 172  
DISPI\_GAS 478  
DISPI\_OIL 478

DISPI\_WAT 478  
DISPJ\_GAS 478  
DISPJ\_OIL 478  
DISPJ\_WAT 478  
DISPK\_GAS 478  
DISPK\_OIL 478  
DISPK\_WAT 478  
DISPLACTOL 654  
DJ 159  
DJFRAC 213  
DK 161  
DKFRAC 213  
DLOADBC 686  
DLOADBC2D 686  
DLOADBC3D 690  
DLOADIJK 690  
DNMIXCOMP 350  
DNMIXENDP 350  
DNMIXFUNC 350  
DPLANES 560  
DRILLQ 862  
DRUCKER 596  
DTMAX 519  
DTMIN 519  
DTOP 165  
DTRAPN 423  
DTRAPW 423  
DTWELL 736  
DTYPE 561  
DUALPERM 204  
DUALPOR 203  
DWOC 496  
DW-RES-UPSTREAM 534  
DYNAGRID 966  
DYNGRDFREQ 120

## E

EACT 377  
EACT\_TAB 377  
ECOEF 606  
ECONDWI 316  
ECONDWJ 316  
ECONDWK 316  
EHEATCYC 310  
ELASTMOD 586, 599, 604, 622, 628  
ELASTOMOD 615  
ELBOUND 977  
ELCONTAB 313  
ELECHEAT 310  
ELECTYPE 310  
ELSCOMPTAB 313  
ELTARGET 977

ELTYPE 310  
ELWCOMPTAB 313  
END-GRID 272  
EPCAP 420  
EPCGO 461  
EPCOMPACT 284  
EPCOW 461  
EPGCP 420  
EPOIL 420  
EPOMF 420  
EPSALT 420  
EPSCUTOFF 622  
EPSURF 420  
EQUALSI 82  
EQUILIBRATE 755  
EV 335  
EXPCG 461  
EXPCW 461  
EXPNI 606  
EXPN2 606

## F

FAULT 247  
FAULTARRAY 249  
FILENAMES 95  
FILM\_COND 217, 975  
FLOIL 420  
FLSALT 420  
FMCAP 420  
FMGCP 420  
FMMOB 420  
FMOIL 420  
FMOMF 420  
FMSALT 420  
FMSURF 420  
FONE 287  
FORCETOL 654  
FORMINFRAC 215  
FPVOLM 650  
FR 279  
FRACANGLE 647  
FRACTURE 69  
FRANGSOFT 622  
FRATIO 606  
FREQFAC 374  
FREQFACP 374  
FRFRAC 215  
FRICANGLE 586, 599, 606, 615,  
    634  
FRICANGMN 606  
FRICANGMX 606  
FRICHARDEN 622  
FTWO 287

## G

GAMMA 604  
GAPPOR 864  
GASD-ZCOEF 353  
GASD-Z-MIN 353  
GASLIQKV 325  
GASSYLIQ 346  
GASSYSLD 341  
GAUSSPNT 654  
GCAPD 599  
GCAPLOC 599  
GCAPMAT 599  
GCAPMODEL 598  
GCAPR 599  
GCAPTEN 599  
GCAPW 599  
GCFATOR 711  
GCINCRMT 599  
GCIOFF 877  
GCONCYCLE 942  
GCONCYCLE\_END 929, 930, 939,  
    941, 943, 945  
GCONCYCLE\_START 929  
GCONCYCR\_END 940  
GCONCYCR\_START 940  
GCONI 835  
GCONIINCR 925  
GCONIMULT 925  
GCONM 846  
GCONP 830  
GCONPINCR 921  
GCONPMULT 921  
GCOUPLING 708  
GCPOFF 877  
GCRV 423  
GCUPDATE 713  
GDI 574  
GDJ 574  
GDK 574  
GENPLAST 622  
GEODEPTH 574  
GEODOMAIN 703  
GEOGRID 574  
GEOM3D 573  
GEOMECH 572  
GEOMETRY 787  
GEORBLOCK 669  
GEOROCK 585  
GEOSOLID 592  
GEOTYPE 585  
GEXPONENTN 604  
GFRACBLK 643  
GLIFT 826  
GLOADBC 695

GLOADBC3D 697  
GMCREEP 613  
GNULL 574  
GOUTSRF 581  
GPATM 606  
GPERMBB 643  
GPERMES 640  
GPERMLC 640  
GPERMTS 640  
GPERMVL 640  
GPHYDDEP 856  
GPOLY 574  
GPTABLE 854  
GPTOLMUL 712  
GRID 150  
GROTAIEI 574  
GROTAIEJ 574  
GROTAIEK 574  
GROUP 737  
GROUPALQ 858  
GRPORTAB 590  
GRTEMTAB 588  
GTRANSLI 574  
GTRANSLJ 574  
GTRANSLK 574  
GUIDEI 874  
GUIDEP 874  
GULBULKMOD 604  
GULOGINT 640  
GVISCOR 355

## H

HARDEN 586  
HEAD-METHOD 746  
HEATR 946  
HEATSLAVE 953  
HFPROP 255  
HLOSSPROP 306  
HLOSST 306  
HLOSSSTDIFF 306  
HTWELL 955  
HVAPR 335  
HVR 335  
HYS\_DRAING 433  
HYS\_DRAINW 433  
HYS\_IMBIBG 433  
HYS\_IMBIBW 433  
HYS\_KRG 433  
HYS\_KRO 433  
HYS\_KRW 433  
HYS\_LEVEL 433  
HYS\_PCOG 433  
HYS\_PCOW 433

HYS\_REVG 433  
HYS\_REVW 433  
HYS\_TOLG 433  
HYS\_TOLW 433

## I

ICE 390  
IDEALGAS 330  
IFTTABLE 418  
IJK 74  
IN\_PR\_SHUT 823  
INCOMP 771  
INCOMPGL 826  
INIT\_FROM\_IMEX 511  
INIT\_FROM\_RESTART 507  
INITFANGLE 622  
INITIAL 487  
INITREGION 488  
INJ\_C\_SWT 823  
INJECTOR 757  
INT 86  
INTCOMP 417  
INTERPOWER 574  
INTERRUPT 143  
INTLIN 418  
INTLOG 418  
INTYPE 488  
INUNIT 104  
ISECTOR 268  
ISOTHERMAL 520  
ITERMAX 541  
ITERMAXG 657  
ITUBE1 883  
IVAR 76

## J

JDUMG 657  
JVAR 77

## K

K\_SURF 331  
KDIR 150  
KL\_SURF 331  
KRCGW 454  
KRINTERP 423  
KRINTRP 423  
KRNOPR 963  
KROCW 454  
KRPRDET 963  
KRPRGRID 963

KRRESET 963  
KRSWITCH 963  
KRTEMTAB 457  
KRTYPE 412  
KRTYPE\_CTRGAS 415  
KRTYPE\_CTROIL 415  
KRTYPE\_CTRWAT 415  
KRTYPE\_VERT 412  
KRWIRO 454  
KRWRO 454  
KV1 323  
KV2 323  
KV3 323  
KV4 323  
KV5 323  
KVAR 78  
KVKEYCOMP 325  
KVTABLE 325  
KVTABLIM 325

## L

LAMINAR 217, 975  
LAYERGRAD 809  
LAYERIJK 803  
LAYERXYZ 800  
LEP-DIAMETER 806  
LEP-DISCHARGE-COEFF 806  
LEP-DISCHARGE-COEFF-CNST 806  
LEP-WELL 806  
LIQLIQKV 325  
LIQPHASE 344  
LIST 61

## M

MANIFOLD 850  
MASSBASIS 107  
MASSDEN 346  
MATBALTOL 525  
MATRIX 68  
MAXERROR 108  
MAXLAYPRE 547  
MAXPRES 544  
MAXSTEPS 518  
MAXTEMP 544  
MCCMODEL 634  
MCETHA 634  
MCOCR 634  
MCOEF 606  
MCOMINDX 634  
MCONNG 660  
MCPREHD 634

MCRCSL 634  
MCSWINDX 634  
MDICLU\_PG 660  
MDSPI\_GAS 476  
MDSPI\_OIL 476  
MDSPI\_WAT 476  
MDSPJ\_GAS 476  
MDSPJ\_OIL 476  
MDSPJ\_WAT 476  
MDSPK\_GAS 476  
MDSPK\_OIL 476  
MDSPK\_WAT 476  
MFRAC\_GAS 499  
MFRAC\_OIL 499  
MFRAC\_WAT 499  
MINC 209  
MINPRES 544  
MINTEMP 544  
MOD 83  
MODEL 319  
MODELSHUT 755  
MOHRCOUL 596  
MOLDEN 346  
MOLDIFF\_DEP 474  
MOLVOL 346  
MONITOR 782  
MPLNE 660  
MRC-RESET 820  
MTVEL 385  
MXCNRPT 781

## N

NB 606  
NCOUPLING 717  
NCUTS 548  
NE 606  
NETGROSS 236  
NETPAY 234  
NEWTONCYC 531  
NINCS 654  
NINEPOINT 155  
NINEPTH 155  
NITERGEO 654  
NLINEAR 603  
NODE4 654  
NODE8 654  
NODE9 654  
NOLIST 61  
NOLISTLIM 144  
NORM 523  
NORTH 537  
NORTHG 657  
NTB 606

NTE 606  
NULL 202  
NULL-PERF 749  
NUMERICAL 517  
NUMSET 521  
NUMTYPE 521

## O

O2CONC 381  
O2PP 381  
OCRV 423  
OILPHASE 344  
ON-TIME 879  
OPEN 757  
OPERATE 774  
ORTHOGG 657  
OUTPRN 114  
OUTSOLVR 142  
OUTSRF 123  
OUTUNIT 104

## P

PARTCLSIZE 114  
PARTOL-TOL 184  
PAUSE 734  
PAYDEPTH 168  
PBASE 279  
PBC 499  
PC\_3RD\_PHASE 461  
PCGEND 454  
PCGWEND 454  
PCON-MANIFOLD 852  
PCRIT 330  
PCWEND 454  
PDEGAA 553  
PDEGAB 554  
PDILA 279  
PEAKANGLE 622  
PEAKCOHES 622  
PERF 789  
PERFV 799  
PERMCK 291  
PERMEXP 291  
PERMI 230  
PERMJ 230  
PERMK 230  
PERMSCALE 385  
PERMSLD 291  
PERMTAB 291  
PERMTABLOG 291  
PERMULI 291

PERMULJ 291  
PERMULK 291  
PFRAC 961  
PFRACF 961  
PGDILA 619  
PGPACT 619  
PGPDMAX 619  
PHWELLBORE 761  
PINCHOUTARRAY 243  
PINCHOUT-TOL 245  
PINJW 769  
PIVOT 540  
PIVOTG 657  
PLOADBC 680  
PLOADBC3D 684  
PLSTRAINY 584  
PNPROSL 551  
PNTHRDS 549  
POISSRATIO 586, 599, 604, 615,  
622, 628, 634  
POR 228  
PORFORM 274  
PORFT 483  
PORINTERP 274  
PORMAX 274  
POROSTOL 717  
PORRATMAX 279  
PPACT 279  
PPATTERN 555  
PPLASTIC 284  
PR\_IN\_SHUT 823  
PRECABG 657  
PRECC 536  
PRECCG 657  
PRES 495  
PRESCBC 670  
PRESCBC3D 676  
PRESSTOL 717  
PRINT\_REF 140  
PRINTGEO 654  
PRIOR-FORM 870  
PRNTORIEN 140  
PROD\_C\_SWT 823  
PRODUCER 757  
PRPOR 274  
PRSR 331  
PSURF 331  
PTRANSI 961  
PTRANSIJ- 961  
PTRANSIJ+ 961  
PTRANSIK- 961  
PTRANSIK+ 961  
PTRANSJ 961

PTRANSK 961  
PTUBE1 883  
PVCUTOFF 264  
PVTOSCMAX 546

## Q

QUAL 769

## R

RANGE 185, 217, 975  
RANGECHECK 59  
RCONBK 574  
RCONBT 574  
RCONF 574  
RCONLF 574  
RCONRT 574  
RCONT 574  
REFBLOCK 489  
REFDEPTH 489  
REFINE 185  
REFPRES 489  
REGIME 217, 975  
RELROUGH 217, 975  
RENTH 377  
REPORTING-GROUP 744  
RESUCOHES 622  
RESTART 109  
RESTIME 109  
REWIND 110  
RG 70  
RIGIDNULL 699  
RIGIDTOP 702  
ROCKCP 295  
ROCKCP\_SHL 295  
ROCKFLUID 411  
ROCKTYPE 273  
RORDER 381  
RPHASE 381  
RPLTABD 648  
RPT 412  
RPWTABD 648  
RRFT 483  
RTEMLOWR 377  
RTEMUPR 377  
RTYPE 412  
RUN 730  
RXCMPFAC 381  
RXCRITCON 381  
RXEQBAK 387  
RXEQFOR 387

## S

SA1 628  
SAMINFO 828  
SBETA 628  
SCONNECT 251  
SCRV 423  
SD\_HCMULT 304  
SD\_REINF 469  
SDBIOTSCF 592  
SDCOHES 592  
SDEGREE 539  
SDEGREEG 657  
SDELASTMOD 592  
SDFRICANG 592  
SDILANG 592  
SDPOISSON 592  
SDTHEXPCF 592  
SECTOR 265  
SECTORARRAY 267  
SECTORNAMES 268  
SETHA1 628  
SG 496  
SGAMMA 628  
SGCON 454  
SGLIM 963  
SGR 454  
SHAPE 211  
SHEAR\_FAC 145  
SHEARMOD 634  
SHEARTAB 371  
SHEARTHICK 367  
SHEARTHIN 367  
SHUTIN 757  
SITERPG 657  
SKAPA1 628  
SKAPA2 628  
SLT 430  
SM 628  
SMALL-RATES 535  
SN 628  
SNG 628  
SO 496  
SOIRG 454  
SOIRW 454  
SOLID\_CP 341  
SOLID\_DEN 341  
SOLIDMIN 372  
SOLVER 550  
SOLVERG 657  
SORDER 538  
SORDERG 657  
SORG 454

SORW 454  
SPATM 628  
SPECGRAV 695  
SPEGRAV 697  
SRFASCII 123  
SSMODEL 628  
STEN 628  
STIFFCOM1 653  
STIFFCOM2 653  
STIFFINIT 653  
STIFFTANG 653  
STOP 731  
STOPROD 374  
STOREAC 374  
STRESI 661, 664  
STRESJ 661, 664  
STRESK 661, 664  
STRESS 661  
STRESS3D 664  
STRESSALL 661  
STRESSGRAD 661  
STRESSGRAD3D 664  
STRESSH 661  
STRESSHIJ 664  
STRESSHIK 664  
STRESSHIJK 664  
STRESSTOL 717  
SUBDOMAIN 205  
SURFLASH 331  
SW 496  
SWCON 454  
SWCRIT 454  
SWIRG 454  
SWR 454  
SWG 454  
SWT 427  
  
**T**  
TARGET 818  
TCRIT 330  
TDMAX 650  
TDMIN 650  
TEMLIM 963  
TEMMULT 316  
TEMP 495  
TEMR 331  
TFORM 520  
THCONANTAB 298  
THCONG 298  
THCONMIX 298  
THCONO 298  
THCONR 298  
  
**U**  
THCONR\_SHL 298  
THCONS 298  
THCONTAB 298  
THCONW 298  
THEXPCOEF 639  
THTYPE 273  
TIME 731  
TINJW 769  
TITLE1 103  
TITLE2 103  
TITLE3 103  
TMPSET 946  
TORTIKE\_VG 355  
TRANLI 240  
TRANLJ 240  
TRANLK 240  
TRANSD 207  
TRANSF 253  
TRANSI 237  
TRANSIENT 217, 828, 975  
TRANSIJ- 308  
TRANSIJ+ 308  
TRANSIK- 308  
TRANSIK+ 308  
TRANSJ 237  
TRANSK 237  
TRANSLATE 63  
TRANSMF 242  
TRANSWB 960  
TRANZONE 489  
TRESCA 596  
TRIGGER 890  
TSURF 331  
TUBING 71  
TUBINSUL 217, 975

UREXPN2 606  
URNB 606  
URNE 606  
URNTB 606  
URNTE 606

## V

VAMOD 194  
VATYPE 194  
VERTICAL 489  
VISCTABLE 359  
VISCTYPE 354  
VISFLOWR 615  
VISINIT 615  
VISPARA 615  
VISPOWER 615  
VISSCHEME 615  
VISTEP 615  
VISTIME 615  
VOLCONST 274  
VOLMOD 232  
VOLSHF 310  
VOLTOL 310  
VONMISES 596  
VSMIXCOMP 365  
VSMIXENDP 365  
VSMIXFUNC 365  
VSTYPE 354

## W

WATPENTH 335  
WATPHASE 344  
WBZ 217  
WBZADJ 217  
WCRV 423  
WELL 741  
WELLALQ 860  
WELLCORE 71, 217  
WELLCORE-REC 975  
WELLINFO 217, 975  
WELLINIT 753  
WELLWALL 217, 975  
WLISOPEN 759  
WLISHTUT 759  
WMKCOMP 844  
WMKUPTO 845  
WOC\_SW 496  
WPRN 112  
WRADIUS 652  
WRECYMASK 843

WRST 110  
WSRF 120  
WTINCR 917  
WTMULT 913

## X

XCORN 177  
XDR 123  
XFLOW-MODEL 751  
XNACL 359

## Y

YCORN 177  
YLDSTRESS 586, 615

## Z

ZCORN 175