

# **A NEW GENERATION CHEMICAL FLOODING SIMULATOR**

Final Report for the Period  
Sept. 2001 - Aug. 2004

Semi-Annual Report for the Period  
April 1, 2004 – August 30, 2004

by  
Gary A. Pope, Kamy Sepehrnoori, and Mojdeh Delshad

January 2005

Work Performed under Contract No. DE-FC-26-00BC15314

Sue Mehlhoff, Project Manager  
U.S. Dept of Energy  
National Petroleum Technology Office  
One West Third Street, Suite 1400  
Tulsa, OK 74103-3159

Prepared by  
Center for Petroleum and Geosystems Engineering  
The University of Texas at Austin  
Austin, TX 78712

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## **ABSTRACT**

The premise of this research is that a general-purpose reservoir simulator for several improved oil recovery processes can and should be developed so that high-resolution simulations of a variety of very large and difficult problems can be achieved using state-of-the-art algorithms and computers. Such a simulator is not currently available to the industry. The goal of this proposed research is to develop a new-generation chemical flooding simulator that is capable of efficiently and accurately simulating oil reservoirs with at least a million gridblocks in less than one day on massively parallel computers. Task 1 is the formulation and development of solution scheme, Task 2 is the implementation of the chemical module, and Task 3 is validation and application. In this final report, we will detail our progress on Tasks 1 through 3 of the project.

# TABLE OF CONTENTS

DISCLAIMER .....	ii
ABSTRACT .....	iii
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
INTRODUCTION .....	1
EXECUTIVE SUMMARY .....	4
EXPERIMENTAL .....	5
RESULTS AND DISCUSSIONS .....	6
Task 1: Formulation and Development of Solution Scheme .....	6
Mass Conservation Equation .....	6
Phase Behavior and Equilibrium Calculations .....	8
Phase Stability Analysis .....	8
Flash Calculation .....	9
Phase Identification and Tracking .....	13
Constraints and Constitutive Equations .....	15
Physical Property Models .....	16
Well Model .....	19
Overall Computation Procedure of the Simulator .....	22
Executive Routines in EOSCOMP .....	24
Description of the Solver .....	25
Solution Approach in Chemical Module .....	26
Non-Orthogonal Grid .....	26
Approximate Equation .....	30
Automatic Time Step Control .....	37
Linear and Nonlinear Iterative Solvers .....	38
Enhancements in Well Model .....	39
Generation of Derivatives using Automatic Differentiation .....	49
Task 2: Formulation and Implementation of Chemical Module .....	50
Explicit Chemical Module .....	50
Fully Implicit Chemical Module .....	57
Task 3: Validation and Application .....	61
Batch Flash of Ethane-Propylene Binary Mixture .....	61
Buckley Leverett 1-D Water Flood .....	63
Carbon Dioxide Sequestration Case .....	67
Six-Component Compositional Simulation Example .....	70
Chemical Module Tests—Explicit Formulation .....	79
2-D Surfactant/Polymer Simulation, Case 1 .....	79
3-D Surfactant/polymer simulation, Case 2 .....	84
3-D Surfactant/polymer simulation, Case 3 .....	88
Chemical Module Tests—Fully Implicit Formulation .....	90
Parallel Simulations .....	98
CONCLUSIONS .....	109
REFERENCES .....	111

NOMENCLATURE .....	115
APPENDIX A—Trapping Number Model Calculations Used in the Jacobian .....	120
APPENDIX B—Residuals and Derivation of Jacobian for Newton Iteration .....	122
Volume constraint.....	122
Material Balance Equations .....	122
Derivatives for Accumulation Term .....	128
Derivatives for Flux Term .....	128
Derivatives for Source/Sink Term .....	148
APPENDIX C—Derivations of the Equations in Appendix B.....	151
APPENDIX D—3-D Surfactant/Polymer Input Data.....	157

## LIST OF TABLES

Table 1.1—Initial and Injected Gas Composition Used in 3-D Simulations.....	47
Table 3.1—Summary of Input Data for Ethane-Propylene Mixture Batch-Flash Calculation .....	62
Table 3.2—Comparison of Phase Compositions From GPAS With UTCOMP and VLMU in Ethane Propylene Batch - Flash Test Case .....	63
Table 3.3—Comparison of Phase Molar Densities From GPAS With UTCOMP and VLMU in Ethane Propylene Batch - Flash Test Case.....	63
Table 3.4—Summary of Input Data for One-Dimensional Buckley Leverett Problem .....	65
Table 3.5—Summary of Input Data For Two Dimensional Carbon Dioxide Sequestration Problem .....	68
Table 3.6—Comparison of Mobile Water Phase and Gas Phase Compositions From GPAS With UTCOMP for Carbon Dioxide Sequestration Problem .....	70
Table 3.7—Absolute Differences in Mobile Water Phase and Gas Phase Compositions From GPAS With UTCOMP for Carbon Dioxide Sequestration Problem After 100 Days .....	70
Table 3.8—Summary of Component Critical Properties and Binary Interaction Coefficients for Two-Dimensional Six Component Simulation Example .....	71
Table 3.9—Summary of Input Data for Two-Dimensional Six Component Simulation Example.....	72
Table 3.10—Summary of Input Data for Three-Dimensional Six Component Simulation Example.....	78
Table 3.11—Input parameters used for Case 1 .....	80
Table 3.12—Input Parameters for Case 2 .....	85
Table 3.13—Input parameters for Case 3 .....	89

## LIST OF FIGURES

Figure 1.1—Physical and computational domains. ....	27
Figure 1.2—Metrics evaluation. ....	29
Figure 1.3—Elemental control volume.....	31
Figure 1.4—Control volume P and its neighbors. ....	33
Figure 1.5—Schematic of the grid used in Case 1 simulations. ....	35
Figure 1.6—Comparison of water production rate using corner point grid implementation and Cartesian grid. ....	35
Figure 1.7—Comparison of oil production rate using corner point grid implementation and Cartesian grid. ....	36
Figure 1.8—Schematic of the grid used in Case 2 simulation.....	36
Figure 1.9—Cumulative water production rate using corner point grid implementation using the grid shown in Fig. 1.8.....	37
Figure 1.10—Cumulative oil production using corner point grid implementation using the grid shown in Fig. 1.8.....	37
Figure 1.11—Oil production rate comparison between PETSc and TICAMA. ....	39
Figure 1.12—Comparison of simulation time for PETSc and TICAMA.....	39
Figure 1.13—Comparison of water production rate for water injection problem with constant pressure injector and constant rate producer. ....	44
Figure 1.14—Comparison of oil production rate for water injection problem with constant pressure injector and constant rate producer. ....	45
Figure 1.15—Comparison of reservoir pressure for water injection problem with constant pressure injector and constant rate producer. ....	45
Figure 1.16—Comparison of water production rate for water injection problem with constant injection rate and constant pressure producer. ....	46
Figure 1.17—Comparison of oil production rate for water injection problem with constant injection rate and constant pressure producer.....	46
Figure 1.18—Comparison of average reservoir pressure for water injection problem with constant injection rate and constant pressure producer. ....	47
Figure 1.19—Comparison of average reservoir pressure for gas injection problem with constant injection rate and constant pressure producer. ....	48
Figure 1.20—Comparison of oil production rate for gas injection problem with constant injection rate and constant pressure producer.....	48
Figure 1.21—Comparison of gas production rate for gas injection problem with constant injection rate and constant pressure producer.....	49
Figure 2.1—Flowchart of Newton method for fully implicit chemical flooding model.....	60
Figure 3.1—Oil/water relative permeability curves used in the Buckley Leverett problem. ....	64
Figure 3.2—Fractional flow curve of water in the Buckley Leverett problem. ....	66
Figure 3.3—Comparison of the water saturation profile from GPAS with the analytical solution and the result from UTCHEM at 0.1, 0.2 and 0.3 pore volumes injected for one-dimensional Buckley Leverett waterflooding problem. ....	66

Figure 3.4—Schematic reservoir geometry (quarter five spot) for the CO2 sequestration problem. ....	67
Figure 3.5—Comparison of the pressure profile along the x-axis obtained from GPAS with UTCOMP at the end of 100 days for CO2 sequestration problem. ....	69
Figure 3.6—Comparison of the gas saturation profile along the x-axis from GPAS with UTCOMP at the end of 100 days for the CO2 sequestration problem. ....	69
Figure 3.7—Comparison of the pressure profile along the x-axis from GPAS with UTCOMP at the end of 100 days for the six component problem. ....	73
Figure 3.8—Comparison of the oil saturation profile along the x-axis from GPAS with UTCOMP at the end of 100 days for the six component problem. ....	73
Figure 3.9—Comparison of the gas saturation profile along the x axis from GPAS with UTCOMP at the end of 100 days for the six component problem. ....	74
Figure 3.10—Schematic reservoir geometry for the three dimensional six component simulation case. ....	75
Figure 3.11—Comparison of the pressure profile along the x-axis from GPAS with UTCOMP for the six component case at the end of two years. ....	75
Figure 3.12—Comparison of the oil saturation profile along the x-axis from GPAS with UTCOMP for the six component case at the end of two years. ....	76
Figure 3.13—Comparison of the gas saturation profile along the x-axis from GPAS with UTCOMP for the six component case at the end of two years. ....	76
Figure 3.14—Comparison of the average reservoir pressure history from GPAS with UTCOMP for the six component case. ....	77
Figure 3.15—Comparison of the producer GOR (gas-oil ratio) history of GPAS with UTCOMP result for the six component case. ....	77
Figure 3.16—Comparison of produced surfactant concentrations for Case 1. ....	81
Figure 3.17—Comparison of produced tracer concentration for Case 1. ....	81
Figure 3.18—Comparison of produced polymer concentration for Case 1. ....	81
Figure 3.19—Polymer concentration (GPAS) at Td=0.5 PV. ....	82
Figure 3.20—Polymer concentration (UTCHEM) at Td=0.5 PV. ....	82
Figure 3.21—Surfactant concentration in fluid (GPAS) at Td=0.5 PV. ....	82
Figure 3.22—Surfactant concentration in fluid (UTCHEM) at Td=0.5 PV. ....	83
Figure 3.23—Oil saturation profile (UTCHEM) at Td=0.5 PV. ....	83
Figure 3.24—Oil saturation profile (GPAS) at Td=0.5 PV. ....	84
Figure 3.25—Comparison of oil production rate and cumulative oil production for Case 1. ....	84
Figure 3.26—Permeability profile. ....	85
Figure 3.27—Injection rate and cumulative fluid injected for Case 2. ....	86
Figure 3.28—Oil production rate and cumulative oil recovery for Case 2. ....	87
Figure 3.29—Average reservoir pressure for Case 2. ....	87
Figure 3.30—Surfactant concentration distribution at the end of chemical slug of 0.1 PV. ....	87
Figure 3.31—Oil saturation distribution during post water flush at 0.5 PV. ....	88
Figure 3.32—Well locations for Case 3. ....	88
Figure 3.33—Parallel performance speedup for Case 3. ....	89
Figure 3.34—Comparison of time spent in solver, Jacobian and MPI communication for Case 3. ....	90



Figure 3.35—Comparison of average reservoir pressure for 1-D surfactant/polymer case. ....	91
Figure 3.36—Comparison of water production rate for 1-D surfactant/polymer case. ....	91
Figure 3.37—Comparison of oil production rate for 1-D surfactant/polymer case. ....	91
Figure 3.38—Comparison of surfactant concentration produced for 1-D surfactant/polymer case. ....	92
Figure 3.39—Comparison of tracer concentration for 1-D surfactant/polymer case. ....	92
Figure 3.40—Comparison of polymer concentration for 1-D surfactant/polymer case. ....	92
Figure 3.41—Comparison of average reservoir pressure for 2-D surfactant/polymer case. ....	93
Figure 3.42—Comparison of water production rate for 2-D surfactant/polymer case. ....	94
Figure 3.43—Comparison of oil production rate for 2-D surfactant/polymer case. ....	94
Figure 3.44—Comparison of surfactant concentration for 2-D surfactant/polymer case. ....	94
Figure 3.45—Comparison of tracer concentration for 2-D surfactant/polymer case. ....	95
Figure 3.46—Comparison of polymer concentration for 2-D surfactant/polymer case. ....	95
Figure 3.47—Comparison of average reservoir pressure for 3-D surfactant/polymer case. ....	96
Figure 3.48—Comparison of water production rate for 3-D surfactant/polymer case. ....	96
Figure 3.49—Comparison of oil production rate for 3-D surfactant/polymer case. ....	97
Figure 3.50—Comparison of surfactant concentration for 3-D surfactant/polymer case. ....	97
Figure 3.51—Comparison of tracer concentration for 3-D surfactant/polymer case. ....	97
Figure 3.52—Comparison of polymer concentration for 3-D surfactant/polymer case. ....	98
Figure 3.53—GPAS execution time and speedup plots for the case of 77x256x10 using single processor per node. ....	100
Figure 3.54—Speedup curves of each section for the simulation case (77x256x10) using Myrinet as the interconnect. ....	101
Figure 3.55—Execution time ratio of Fast Ethernet to Myrinet for each sections of the larger simulation case (77x256x10). ....	101
Figure 3.56—Execution time ratio of single-processor to dual-processor for each sections of the simulation case (77x256x10) on the cluster with Myrinet. ....	102
Figure 3.57—Well locations for Case 3. ....	103
Figure 3.58—Average reservoir pressure for the parallel surfactant/polymer flood simulation (96x96x11). ....	103
Figure 3.59—Water production rate for the parallel surfactant/polymer flood simulation (96x96x11). ....	104

Figure 3.60—Oil production rate for the parallel surfactant/polymer flood simulation (96x96x11). .....	104
Figure 3.61—Cumulative oil recovery for the parallel surfactant/polymer flood simulation (96x96x11). .....	104
Figure 3.62—Time step for the parallel surfactant/polymer flood simulation (96x96x11). .....	105
Figure 3.63—Number of Newton iteration for the parallel surfactant/polymer flood simulation (96x96x11). .....	105
Figure 3.64—Average reservoir pressure for the parallel surfactant/polymer flood simulation (384x384x7). .....	106
Figure 3.65—Total water and oil production rates for the parallel surfactant/polymer flood simulation (384x384x7). .....	107
Figure 3.66—Cumulative oil recovery for the parallel surfactant/polymer flood simulation (384x384x7). .....	107
Figure 3.67—Time step values for the parallel surfactant/polymer flood simulation (384x384x7). .....	108
Figure 3.68—Number of Newton iterations for the parallel surfactant/polymer flood simulation (384x384x7). .....	108

# INTRODUCTION

Increased oil production using improved oil recovery processes requires numerical modeling of such processes to minimize the risk involved in development decisions. The oil industry is requiring much more detailed analyses with a greater demand for reservoir simulation with geological, physical, and chemical models of much more detail than the past. Reservoir simulation has become an increasingly widespread and important tool for analyzing and optimizing oil recovery projects.

Numerical simulation of large petroleum reservoirs with complex recovery processes is computationally challenging due to the problem size and detailed property calculations involved. This problem is compounded by the finer resolution needed to model such processes accurately. Traditionally, such simulations have been performed on workstations or high-end desktop computers. These computers restrict the problem size due to their addressable memory limit and simulation studies of the entire project life become time consuming. Parallel reservoir simulation especially on low cost, high performance computing clusters has alleviated these issues to a certain extent. Recent publications describe the development of such approaches and emphasize the necessity and advantages of using parallel processing (Dogru *et al.*, 2002; Habiballah *et al.*, 2003; Gai *et al.*, 2003; Zhang *et al.*, 2001, Wang *et al.*, 1997).

Compositional reservoir simulators which are based on equation of state (EOS) formulations do not handle the modeling of aqueous phase behavior and those which are designed for chemical flood modeling typically assume simplified hydrocarbon phase behavior. There is need to have a single reservoir simulator capable of combining both approaches to benefit from the advantages of both models. The overall objective of this research is to develop such technology using a computational framework that also allows parallel processing. The initial stage of development involved the formulation of a fully implicit, parallel, EOS compositional simulator (Wang *et al.*, 1997). The description of the framework approach used for modular code development and the application to gas injection is given in Wang *et al.*, 1999.

Here we report on the implementation of the chemical module to the existing EOS simulator, its validation and application to large-scale chemical flooding simulations. The formulation of the compositional model is first described. The assumptions for the chemical model and its formulation are described next. We use Hand's rule (Hand, 1939) to describe surfactant/oil/brine Type II(-) phase behavior. The trapping number model for relative permeability is implemented to capture the changes in residual saturations caused due to the lowered interfacial tension. The validation of the implementation against the explicit chemical flooding simulator UTCHEM is shown. Application to large-scale problems and tests showing the parallel performance of the simulator are described. With the capability of parallel processing, the general-purpose adaptive simulator (GPAS) can now be used to simulate chemical flooding on a larger scale than before.

The need for a reservoir simulation framework that supports and eases physical model development led to the development of the Integrated Parallel Accurate Reservoir Simulator (IPARS) at The University of Texas, Austin (Parashar *et al.*, 1997). The computational framework of IPARS is used to separate the physical reservoir model development from the code involving parallel processing, solvers and other auxiliary functions. The IPARS framework supports three-dimensional multiphase, multicomponent isothermal flow. It has encapsulated functions to perform the input processing, memory allocation and management, domain decomposition, well management, output generation and other functions like table lookups, interpolation, etc.

The framework natively supports parallel computation on both distributed and shared memory computers using message passing. On multiprocessor computers, the reservoir domain is divided over the y-direction into several subdomains equal to the number of processors allocated for the run. The computations associated with each of these subdomains are distributed to individual processors. An additional layer of gridblocks is added surrounding each subdomain assigned to a processor. The computational framework provides a routine that updates data in this communication layer to enable each processor to perform its calculations while communicating necessary data with the other processors using the message passing interface (MPI).

The primary source of the solvers used in the framework is that provided by the software package called PETSc (Portable Extensible Toolkit for Scientific Computation) (Balay *et al.*, 1997). Other solvers based on the Generalized Minimum Residual Method (GMRES), Line Successive Over-relaxation (LSOR) and Pre-conditioned Conjugate Gradient (PCG) methods have also been implemented as additional options in the framework. All of these solvers provide the users great flexibility and control of the numerical methods used for the solutions.

Finite difference formulations with a block-centered grid are used in the framework. The nonlinear difference equations are solved by either fully implicit or semi-implicit techniques. Both mass balance and volume balance is supported. Since the portability of the simulator is very important, FORTRAN 77 is used wherever possible. For memory management and user interaction, classical C is used. Commercial libraries are prohibited except in the graphics front end to the simulator. The simulator is formulated for a distributed memory, message passing machine. Free-format keyword input is used for direct data input. The simulator uses a single set of units to solve the partial differential equations. However, the user may choose any physically correct units for the input variables and they will then be converted to the default internal unit using appropriate conversion factor.

The overall structure of IPARS framework consists of 3 layers:

- Executive layer that consists of routines that direct the overall course of the simulation
- Work routines that are typically FORTRAN subroutines that perform grid element computations.

- Data-management layer that handles the distribution of grid across processing nodes, local storage allocation, dynamic reallocation and dynamic load balancing, and communication rescheduling. The data-management layer is also responsible for checkpoint/restart, input/output and visualization.

The fully implicit EOS compositional formulation has already been implemented into the IPARS framework and successfully tested. The Peng-Robinson EOS is used for hydrocarbon phase behavior calculations. The linear solvers from PETSc package are used for the solution of underlying linear equations. The framework provides hooks for implementing specific physical models. Such hooks bridge the models with the framework. There are several executive routines, and any communications between the processors and compositional model are performed in these routines. Many tests have been performed using the EOS compositional simulator on variety of computer platforms such as IBM SP and a cluster of PCs (Wang *et al.*, 1999; Uetani *et al.*, 2002).

The goal of this project was to add a chemical module to the existing compositional Peng-Robinson cubic equation of state (EOS). The simulator called GPAS, General Purpose Adaptive Reservoir Simulator, will then include the IPARS framework, and the compositional and chemical modules. We detail our progress on Tasks 1 through 3 throughout the three-year project. We have formulated the mass conservation equation and physical properties for chemical species such as tracers, polymer, surfactant and electrolytes. We implemented and validated the chemical module. The chemical module was added both using a hybrid approach and also a fully implicit formulation. We have verified and validated the formulation and the implementation in GPAS by making comparison with analytical solutions of the known problem, with other reservoir simulators such as UTCHEM, UTCOMP, and Eclipse.

We have also conducted a series of benchmarks by running the General Purpose Adaptive (GPAS) simulator on a Linux cluster and studied the scalability while using different interconnects. The results were very encouraging and indicated that GPAS performance scales linearly from one to 64 single processor nodes using a low latency, high-bandwidth such as Myrinet.

The progress in the last six months of the project was on addition of more flexible corner point grid geometry and also the implementation of the fully implicit surfactant and its properties. We performed a series of validation tests to compare the simulation results accuracy and efficiency of the explicit and fully implicit chemical modules.

## EXECUTIVE SUMMARY

The premise of this research is that a general-purpose reservoir simulator for several improved oil recovery processes can and should be developed so that high-resolution simulations of a variety of very large and difficult problems can be achieved using state-of-the-art algorithms and computers. Such a simulator is not currently available to the industry. The goal of this proposed research is to develop a new-generation chemical flooding simulator that is capable of efficiently and accurately simulating oil reservoirs with at least a million gridblocks in less than one day on massively parallel computers. Task 1 is the formulation and development of solution scheme, Task 2 is the implementation of the chemical module, and Task 3 is validation and application. We made significant progress on all three tasks and we were on schedule on both technical and budget. In this report, we will detail our progress on Tasks 1 through 3 of the project.

We report on the implementation of the chemical module to the existing EOS simulator, its validation and application to large-scale chemical flooding simulations. The formulation of the compositional model is first described. The assumptions for the chemical model and its formulation are described next. The aqueous species added as part of the chemical model are surfactant, polymer, electrolytes, and tracers. We use Hand's rule (Hand, 1939) to describe surfactant/oil/brine Type II(-) phase behavior. The trapping number model for relative permeability is implemented to capture the changes in residual saturations caused due to the lowered interfacial tension. The validation of the implementation against the explicit chemical flooding simulator UTCHEM is shown. Application to large-scale problems and tests showing the parallel performance of the simulator are described. We have implemented the chemical module using both explicit and fully implicit formulations. The explicit approach we used to couple the models is easy to implement, computationally efficient and extendable to many other interesting reservoir problems involving aqueous chemistry. With the fully implicit formulation we can take advantage of large time steps.

We have also added a corner point grid geometry option for modeling more complex reservoir geometries.

With the capability of parallel processing, the general-purpose adaptive simulator (GPAS) can now be used to simulate chemical flooding on a larger scale than before.

We have also conducted a series of benchmarks by running GPAS on Xeon-based Linux cluster and studied the scalability while using different interconnects for the cluster. The simulations were performed for a gas injection process in a reservoir with 197,120 gridblocks and total of 88 wells in a staggered line drive pattern. The speed up results indicated a very good performance. The GPAS showed the best performance as well as scalability (speedup) on a cluster with Myrinet.

## **EXPERIMENTAL**

This project does not include an experimental task.

## RESULTS AND DISCUSSIONS

### ***Task 1: Formulation and Development of Solution Scheme***

The effort on this task was directed towards the formulation of tracer, polymer, electrolytes, and surfactant species in GPAS. We implemented the aqueous species using both explicit and fully implicit methods. The explicit formulation was based on a hybrid approach where the mass conservation equations for hydrocarbon species are solved implicitly where the aqueous species mass balances are solved explicitly using an updated phase fluxes, saturations, and densities. To take advantage of the larger time steps with the fully implicit formulation to reduce the simulation time, we have developed a fully implicit module of chemical flood with the relevant physical properties. The advantage of the hybrid method is in the fast and easy implementation of the physical models.

For completeness, we first give a review of the mathematical formulation of the compositional model.

### **Mass Conservation Equation**

The assumptions made in developing the formulation are:

- Reservoir is isothermal.
- Darcy's law describes the multiphase flow of fluids through the porous media.
- Impermeable zones represented by the no-flow boundaries surround the reservoir.
- The injection and production of fluids are treated as source or sink terms.
- The rock is slightly compressible and immobile.
- Each hydrocarbon phase is composed of  $n_c$  hydrocarbon components, which may include the non-hydrocarbon components such as  $\text{CO}_2$ ,  $\text{N}_2$  or  $\text{H}_2\text{S}$ .
- Instantaneous local thermodynamic equilibrium between hydrocarbon phases.
- Negligible capillary pressure effects on hydrocarbon phase equilibrium.
- Water is slightly compressible and water viscosity is constant.

The general mass conservation equation for species  $i$  in a volume  $V$  can be expressed as

$$\begin{aligned} &\{\text{Rate of accumulation of } i \text{ in } V\} \\ &= \{\text{Rate of } i \text{ transported into } V\} \\ &\quad - \{\text{Rate of } i \text{ transported from } V\} \\ &\quad + \{\text{Rate of production of } i \text{ in } V\}, i = 1, \dots, N_c \end{aligned} \tag{1.1}$$

The differential form for the species conservation equation can be expressed as:



$$\frac{\partial W_i}{\partial t} + \vec{\nabla} \bullet \vec{N}_i - R_i = 0 \quad (1.2)$$

where  $W_i$  is the overall concentration of  $i$  in units of mass of  $i$  per unit bulk volume,  $\vec{N}_i$  is the flux vector of species  $i$  in units of mass of  $i$  per surface area-time and  $R_i$  is the mass rate of production in units of mass of  $i$  per bulk volume-time.

The mass balance equation can be expressed in terms of moles per unit time by defining each term of Eq. 1.2 in terms of the porous media and fluid properties such as porosity, permeability, density, saturations, compositions, rates etc. The accumulation term for a porous medium becomes

$$W_i = \phi \sum_{j=1}^{n_p} \xi_j S_j x_{ij} \quad (1.3)$$

where  $\phi$  is the porosity,  $\xi_j$  is the molar density of phase  $j$ ,  $S_j$  is the saturation of phase  $j$  and  $x_{ij}$  is the mole fraction of component  $i$  in phase  $j$ .

The flux vector of component  $i$  is a sum of the convective and the dispersive flux, and can be expressed as

$$\vec{N} = \sum_{j=1}^{n_p} \xi_j x_{ij} \vec{u}_j - \phi \xi_j S_j \vec{K}_{ij} \bullet \vec{\nabla} x_{ij} \quad (1.4)$$

where  $\vec{u}_j$  represents the superficial velocity or flux of phase  $j$ . The flux is evaluated using the Darcy's law for multiphase flow of fluids through porous media.

$$\vec{u}_j = -\vec{K} \lambda_{rj} (\vec{\nabla} P_j - \gamma_j \vec{\nabla} D) \quad (1.5)$$

Darcy's law is a fundamental relationship describing the flow of fluids in permeable media under laminar flow conditions. The differential form of Darcy's law can be used to treat multiphase unsteady state flow, non-uniform permeability, non-uniform pressure gradients. It is used to govern the transport of phases from one cell to another under the local pressure gradient, rock permeability, relative permeability and viscosity. Converting each of the terms in the mass balance equation to units of moles per unit time and expressing the flux using Darcy's law, the mass balance for each component  $i$  is the following partial differential equation:

$$\frac{\partial}{\partial t} \left( \phi \sum_{j=1}^{n_p} \xi_j S_j x_{ij} \right) + \vec{\nabla} \bullet \left[ \sum_{j=1}^{n_p} \xi_j \lambda_{rj} x_{ij} (\vec{\nabla} P_j - \gamma_j \vec{\nabla} D) + \phi \xi_j S_j \vec{K}_{ij} \bullet \vec{\nabla} x_{ij} \right] - \frac{q_i}{V_b} = 0 \quad (1.6)$$

Multiplying both sides of Eq. 1.6 by  $V_b$  gives

$$V_b \frac{\partial}{\partial t} \left( \phi \sum_{j=1}^{n_p} \xi_j S_j x_{ij} \right) - V_b \vec{\nabla} \cdot \left[ \sum_{j=1}^{n_p} \xi_j \lambda_j x_{ij} (\vec{\nabla} P_j - \gamma_j \vec{\nabla} D) + \phi \xi_j S_j \vec{K}_{ij} \cdot \vec{\nabla} x_{ij} \right] - q_i = 0 \quad (1.7)$$

for  $i = 1, \dots, n_c$

The above equation is written in terms of moles per unit time, in which  $q_i$  is the molar injection (positive) or production (negative) rate for component  $i$ . The mobility for phase  $j$  is defined as

$$\lambda_j = \frac{k_{rj}}{\mu_j} \quad (1.8)$$

The physical dispersion term has not yet been implemented in GPAS.

## Phase Behavior and Equilibrium Calculations

The phase equilibrium relationship determines the number, amounts and compositions of all the equilibrium phases.

The sequence of phase equilibrium calculations is as follows:

1. The number of phases in a gridblock is determined using the phase stability analysis.
2. After the number of phases is determined, the composition of each equilibrium phase is determined.
3. The phases in the gridblock are tracked for the next time step calculations.

## Phase Stability Analysis

The stability algorithm has not been implemented in Equation of State Compositional Model (EOSCOMP). One way of doing the phase stability analysis was given by Michelsen (1982). The Michelsen's approach could be used for the implementing the phase stability analysis in EOSCOMP as explained below.

A stability analysis on a mixture of overall hydrocarbon composition  $\bar{Z}$  is a search for a trial phase, taken from the original mixture that, when combined with the remainder of the mixture, gives a value of Gibbs free energy that is lower than a single-phase mixture of overall hydrocarbon composition,  $\bar{Z}$  (Michelsen, 1982; Trangenstein, 1987; Chang, 1990). If such a search is successful, an additional phase must be added to the phase equilibrium calculation. This condition is expressed mathematically as

$$\Delta G = \sum_{i=1}^{n_c} y_i [\mu_i(\bar{Y}) - \mu_i(\bar{Z})] \quad (1.9)$$

where  $\mu_i$  is the chemical potential of component  $i$  and  $y_i$  is the mole fraction of component  $i$  in the trial phase. Thus, if for any set of mole fractions the value of  $\Delta G$  at constant temperature and pressure is greater than zero, then the phase will be stable. If a composition can be found such that  $\Delta G < 0$ , the phase will be unstable.

The phase stability analysis is to solve the following set of nonlinear equations for the variables  $Y_i$

$$\ln Y_i + \ln \phi_i(\bar{Y}) - h_i = 0 \text{ for } i=1, \dots, n_c \quad (1.10)$$

where the mole fraction  $\bar{Y}$  and  $h_i$  is related to these variables by

$$y_i = \frac{Y_i}{\sum_{i=1}^{n_c} Y_i} \quad (1.11)$$

$$h_i = \ln Z_i + \ln \phi_i(\bar{Z}) \text{ for } i=1, \dots, n_c \quad (1.12)$$

## Flash Calculation

Once a mixture has been shown to split into more than one phase by the stability calculation, the flash involves calculation of the mole fraction and composition of each phase at the given temperature, pressure and overall composition of the fluid. The governing equations for the flash require equality of component fugacities and mass balance.

The equilibrium solution must satisfy three conditions:

- Mass conservation of each component in the mixture
- Chemical potentials for each component are equal in all phases
- Gibbs free energy at constant temperature and pressure is a minimum

Fugacity is calculated using the Peng Robinson equation-of-state. The phase composition constraint, which states that the sum of the mole fraction of all the components in a phase equals to one, and the Rachford-Rice equation for determining the phase amounts for two hydrocarbon phases are implicitly used in the solution of the fugacity equation. The Rachford-Rice equation is used to determine the phase compositions and amounts. This equation requires the values of the equilibrium ratios  $K_i$ , which are defined as the ratio of the mole fractions of component  $i$  in oil and gas phases, respectively. The  $K_i$  values are determined by the equality of component fugacities in each phase. The fugacity equality,

the Rachford-Rice equation and the Peng Robinson equation of state are described below in detail.

### Equality of the Component Fugacity

One of the criteria for phase equilibrium is the equality of the partial molar Gibbs free energies or the chemical potentials. Alternatively, this criterion can be expressed in terms of fugacity (Sandler, 1999): with the assumption of local thermodynamic equilibrium for the hydrocarbon phases, the criterion of phase equilibrium applies (Smith and Van Ness, 1975), namely

$$f_{ij} = f_{i\ell} \text{ for } i = 1, \dots, n_c \text{ and } j = 2, \dots, n_p (j \neq \ell) \quad (1.13)$$

where phase  $\ell$  has been chosen as a reference phase. The fugacity of a component in a phase is taken as a function of pressure and phase composition, at a given temperature,

$$f_{ij} = f_{ij}(P, \overline{x_j}) \text{ for } i = 1, \dots, n_c \text{ and } j = 2, \dots, n_p (j \neq \ell) \quad (1.14)$$

### *Composition Constraint*

The phase composition constraint is

$$\sum_{i=1}^{n_c} x_{ij} - 1 = 0 \quad (1.15)$$

where the mole fractions are defined as

$$x_{ij} = \frac{n_{ij}}{n_j} \text{ for } i = 1, \dots, n_c \text{ and } j = 2, \dots, n_p \quad (1.16)$$

### Rachford-Rice Equation

In a classical flash calculation, the amount and composition of each equilibrium phase is evaluated using a material-balance equation after each update of the K-value from the equation-of-state:

$$r(v) = \sum_{i=1}^{n_c} \frac{(K_i - 1)Z_i}{1 + v(K_i - 1)} = 0 \quad (1.17)$$

where  $v$  is the mole fraction of gas in absence of water,  $K_i$  is the equilibrium ratio,  $Z_i$  is the overall mole fraction of component  $i$  in the feed and  $r(v)$  is the residual of the Rachford-Rice equation.

The component mole fractions in the liquid and gas phases are then computed from the equations:

$$x_i = \frac{Z_i}{1 + v(K_i - 1)} \quad (1.18)$$

$$y_i = \frac{Z_i K_i}{1 + v(K_i - 1)} \quad (1.19)$$

The range for  $v$  is defined by

$$v_l = \frac{1}{1 - K_{\max}} < 0$$

$$v_r = \frac{1}{1 - K_{\min}} > 0 \quad (1.20)$$

Equation 1.17 is a monotonically decreasing function of  $v$  with asymptotes at  $v_l$  and  $v_r$ . Usually, a Newton iteration can efficiently solve Eqs. 1.18 and 1.19 for  $v$ . However, round-off errors occur when solving Eq. 1.17.

### Leibovici and Neoschil Equation

EOSCOMP solves the Rachford-Rice Equation. As was described above, round-off errors occur when solving the Rachford-Rice equation. To avoid the round off errors that occur when solving Eq. 1.17, the original Rachford-Rice equation can be changed into a form that is more nearly linear with respect to  $v$  as done by Leibovici and Neoschil (1992) and this approach could be implemented in EOSCOMP. The Leibovici and Neoschil equation is given by

$$\tilde{r}(v) = (v - v_l)(v_r - v) \sum_{i=1}^{n_c} \frac{(K_i - 1)Z_i}{1 + v(K_i - 1)} = 0 \quad (1.21)$$

The range for  $v$  is defined by

$$v_{\min} = \max \left( \frac{Z_i K_i - 1}{K_i - 1} \right) \text{ for } K_i > 1 \text{ and } Z_i < 0 \quad (1.22)$$

$$v_{\max} = \min \left( \frac{Z_i - 1}{K_i - 1} \right) \text{ for } K_i < 1 \text{ and } Z_i > 0 \quad (1.23)$$

Also, the Newton procedure is used to solve Eq. 1.21 for  $v$ .

### Equation of State

The Peng Robinson equation of state (Peng and Robinson, 1976) is

$$P = \frac{RT}{V - b} - \frac{a(T)}{V(V + b) + b(V - b)} \quad (1.24)$$

The parameters  $a$  and  $b$  for a pure component are computed from

$$a(T) = 0.45724 \frac{R^2 T_c^2}{P_c} \alpha(T) \quad (1.25)$$

$$\sqrt{\alpha} = 1 + \kappa \left( 1 - \sqrt{\frac{T}{T_c}} \right) \quad (1.26)$$

$$b = 0.07780 \frac{RT_c}{P_c} \quad (1.27)$$

$$\kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2 \text{ if } \omega < 0.49 \quad (1.28)$$

$$\kappa = 0.379640 + 1.485030 \omega - 0.164423 \omega^2 + 0.016666 \omega^3 \text{ if } \omega \geq 0.49 \quad (1.29)$$

For a multi component mixture, the mixing rules for the two parameters are

$$a = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} x_i x_j \sqrt{a_i a_j} (1 - k_{ij}) \quad (1.30)$$

$$b = \sum_{i=1}^{N_c} x_i b_i$$

where for component  $i$ , the  $a_i$  is computed from Eq. 1.25, and  $b_i$  is computed from Eq. 1.27. The constant,  $k_{ij}$  is called the binary interaction coefficient between components  $i$  and  $j$ .

The Peng Robinson Equation of state can be written in the form

$$Z^3 + \alpha Z^2 + \beta Z + \gamma = 0 \quad (1.31)$$

where  $Z = \frac{PV}{RT}$  is the compressibility factor, and the parameters are expressed as

$$\alpha = -1 + B \quad (1.32)$$

$$\beta = A - 3B^2 - 2B \quad (1.33)$$

$$\gamma = -AB + B^2 + B^3 \quad (1.34)$$

$$A = \frac{aP}{(RT)^2} \quad (1.35)$$

$$B = \frac{bP}{RT} \quad (1.36)$$

In GPAS, the equation-of-state calculations are done in the subroutine named XEOS, which contains the four subroutines EOSPURE, EOSMIX, EOSPHI and EOSPARTIAL. The equation-of-state parameters for each pure component are calculated in the subroutine EOSPURE. Mixture values are calculated in the subroutine EOSMIX. The fugacity coefficient is calculated from the equation-of-state in the subroutine EOSPHI and the equation-of-state related derivatives are computed in the EOSPARTIAL subroutine. The subroutine EOSCUB solves the Peng Robinson cubic equation-of-state and calculates the compressibility factor and its derivative. EOSCOMP requires the pure component critical temperature, critical pressure, critical volume, acentric factors, molecular weights and binary interaction coefficients to calculate the equation of state parameters. The volume shift parameter is not implemented in GPAS.

The main flash subroutine in GPAS is XFLASH. This subroutine performs the flash calculation at a given initial composition, temperature and pressure. The number of components, binary interaction coefficients and the equilibrium ratio values are also part of the input to this subroutine. An initial estimate of the equilibrium values is done in the subroutine STABL, which is then passed to the flash calculations. The flash subroutine calculates the liquid and vapor phase mole fractions, liquid and vapor compressibility factors, and also the negative residual of component  $i$  in cell  $k$ . The subroutine SOLVE solves the Rachford-Rice equation for finding the phase composition and the phase amounts.

## Phase Identification and Tracking

Phase identification deals with the labeling of a phase as oil, gas, or aqueous phase at the initial conditions and also when a new phase appears. After a phase has been identified, phase tracking does the labeling of a phase during the simulation. Labeling phases consistently is important because of the need to assign a consistent relative permeability to each phase during a numerical simulation. Perschke (1988) developed a method for the phase identification and tracking in which both phase mass density and phase composition are used. This is the procedure followed in GPAS. Once a phase has been

identified, it is tracked during simulation by comparing the mole fraction value of a selected or key component in the equilibrium phases at the new time step with the values at the old time step. The phases at the new time step are labeled such that the mole fraction values are closest to the values at the old time step.

The algorithm used in GPAS for naming a phase when the hydrocarbon mixture is a single phase is similar to that proposed by Gosset *et al.* (1986). The parameters A and B of a two-parameter cubic EOS are computed from

$$A = \frac{aP}{(RT)^2} = \Omega_a \frac{PT_c^2}{P_c T^2} \alpha \quad (1.37)$$

$$B = \frac{bP}{RT} = \Omega_b \frac{PT_c}{P_c T} \quad (1.38)$$

where

$$\Omega_a = 0.4572355299$$

$$\Omega_b = 0.077796074$$

Dividing Eq. 1.37 by Eq. 1.38 gives:

$$\frac{A}{B} = \frac{\Omega_a T_c}{\Omega_b T} \alpha \quad (1.39)$$

where  $\alpha$  is defined in Eq. 1.26.

A fluid is assumed to be in single-phase if  $T > T_c$ , which also implies  $\alpha \leq 1$ . From Eq. 1.39 this implies

$$\frac{A}{B} \leq \frac{\Omega_a}{\Omega_b} \quad (1.40)$$

or its molar volume to be greater than the critical molar volume,  $v > v_c$ , which implies

$$Z > \frac{BZ_c}{\Omega_b} \quad (1.41)$$

In GPAS, the subroutine EOS\_1PH identifies a single phase as oil or gas using the above method. Also, an option is provided in the code to identify a single phase by the



conventional method: The fluid is liquid when sum of  $Z_i K_i = 1$ , and the fluid is gas when the sum of  $Z_i / K_i = 1$ .

## Constraints and Constitutive Equations

### Volume Constraint

The volume constraint states that the pore volume in each of the cells must be filled completely by the total fluid volume. This is expressed in a mathematical form as

$$V_B \sum_{i=1}^{n_c} N_i \sum_{j=1}^{n_p} L_j \bar{v}_j - V_p = 0 \quad (1.42)$$

where  $N_i$  is the number of moles of each component  $i$  per unit bulk volume,  $L_j$  is the ratio of moles in phase  $j$  to the total number of moles in the mixture,  $\bar{v}_j$  is the molar volume of phase  $j$ ,  $V_B$  is the bulk volume and  $V_p$  is the pore volume of a cell.

### Saturation Constraint

The saturation constraint is

$$\sum_{j=1}^{n_p} S_j = 1 \quad (1.43)$$

### Phase Pressures

The phase pressure is related to the capillary pressure and a reference pressure,  $P_{ref}$ :

$$P_j = P_{ref} + P_{cref,j} \text{ for } j = 1, \dots, n_p \quad (1.44)$$

For  $n_p$  phases, there are  $(n_p-1)$  independent capillary equations. The capillary pressure is a function of phase saturations and compositions,

$$P_{cref,j} = P_{cref,j}(\vec{S}, \vec{x}) \text{ for } j = 1, \dots, n_p \quad (1.45)$$

### Porosity

The porosity is a function of pressure.

$$\phi = \phi_{ref} (1 + c_f (P - P_{ref})) \quad (1.46)$$

where  $\phi_{\text{ref}}$  is the porosity at the reference pressure,  $P_{\text{ref}}$  and  $\phi$  is calculated at the pressure  $P$ .

In GPAS, the porosity calculation is performed in the subroutine AQUEOUS. The AQUEOUS subroutine also calculates the aqueous phase molar and mass densities.

## Physical Property Models

In this section, the physical models implemented in GPAS to calculate the viscosities, interfacial tension, relative permeability, capillary pressure, phase molar density and the hydrocarbon solubility in water are described.

### Viscosity

The aqueous phase viscosity is constant and is specified as user input. The gas and oil viscosity computed using the Lohrenz et al. (1964) correlation. An option is also available to input the oil phase viscosity. All the phase viscosity calculations are performed in the subroutine VIS, which consists of the subroutines LVISC1 and LVISC2. LVISC1 subroutine computes the viscosity of pure components at low pressures while LVISC2 subroutine computes the viscosity of a mixture and its derivatives with respect to phase composition and pressure at high pressures.

### *Lohrenz et al. Correlation*

The Lohrenz *et al.* (1964) correlation combines several viscosity correlations as described below. The steps involved in calculating the phase viscosity are given below:

- Computation of the low-pressure, pure-component viscosity

$$\hat{\mu}_i = \frac{0.00034 T_{ri}^{0.94}}{\zeta_i} \text{ for } T_{ri} \leq 1.5$$

or

$$\hat{\mu}_i = \frac{0.0001776(4.58 T_{ri} - 1.67)^{5/8}}{\zeta_i} \text{ for } T_{ri} > 1.5 \quad (1.47)$$

where

$$\zeta_i = \frac{5.44 T_{ci}^{1/6}}{MW_i^{1/2} P_{ci}^{2/3}}$$

- Calculation of the low pressure viscosity

$$\mu_j^* = \frac{\sum_{i=1}^{n_c} x_{ij} \hat{\mu}_i \sqrt{MW_i}}{\sum_{i=1}^{n_c} x_{ij} \sqrt{MW_i}} \quad (1.48)$$

- The reduced phase molar density calculation

$$\xi_{jr} = \xi_j \sum_{i=1}^{n_c} x_{ij} V_{ci} \quad (1.49)$$

$$\eta_j = \frac{5.44 \left[ \sum_{i=1}^{n_c} x_{ij} T_{ci} \right]^{1/6}}{\left[ \sum_{i=1}^{n_c} x_{ij} MW_i \right]^{1/2} \left[ \sum_{i=1}^{n_c} x_{ij} P_{ci} \right]^{2/3}}$$

- The Phase viscosity calculation at the desired pressure

$$\mu_j = \mu_j^* + 0.000205 \frac{\xi_{jr}}{\eta_j} \text{ for } \xi_{jr} \leq 0.18$$

$$\mu_j = \frac{\mu_j^* + (\chi_j^4 - 1)}{10^4 \eta_j} \text{ for } \xi_{jr} > 0.18 \quad (1.50)$$

where

$$\chi_j = 1.023 + 0.23364 \xi_{jr} + 0.58533 \xi_{jr}^2 - 0.40758 \xi_{jr}^3 + 0.093324 \xi_{jr}^4$$

### Interfacial Tension

The interfacial tension between two hydrocarbon phases is calculated from the Macleod-Sudgen correlation as reported in Reid, Prausnitz and Poling (1987):

$$\sigma_{jl} = \left[ 0.016018 \sum_{i=1}^{n_c} \psi_i (\xi_j x_{ij} - \xi_l x_{il}) \right]^4 \quad (1.51)$$

where  $\psi_i$  is the parachor of component i.

In GPAS, the interfacial tension between gas and oil for gridblock k and the derivatives of the interfacial tension are calculated in the subroutine IFT.

### Relative Permeability

The two-phase relative permeability is given as tabular input. The three -phase relative permeability to water, oil and gas is calculated based on the two-phase relative permeability data using Stone's method I or II.

#### *Stone's Model II*

The three-phase oil relative permeability is calculated from the two-phase relative permeability using

$$k_{r2} = k_{r2}^0 \left[ \left( \frac{k_{r21}}{k_{r2}^0} + k_{r1} \right) \left( \frac{k_{r23}}{k_{r2}^0} + k_{r3} \right) - (k_{r1} + k_{r3}) \right] \quad (1.52)$$

where the oil/water and oil/gas two-phase relative permeability is obtained from the input tables.

The subroutine RELPERM3EOS computes the three-phase relative permeability to water, oil and gas based on the two-phase data.

### Capillary Pressure

The gas-oil and water-oil capillary pressure data are inputted as tables. The capillary pressure at any saturation is interpolated from the tables.

### Hydrocarbon Solubility in Water

The solubility of the hydrocarbons in the water is calculated, separately from the flash calculations, using Henry's law. The vaporization of water into hydrocarbon phases is ignored. When the solubility calculations are treated separately from the flash calculations, the component molar balance equations are expressed as

$$\frac{\partial(N_i^h + N_i^a)}{\partial t} - \bar{\nabla} \cdot \sum_{j=1}^{n_p} \xi_j \lambda_j x_{ij} (\bar{\nabla} P_j - \gamma_j \bar{\nabla} D) - \frac{q_i}{V_b} = 0 \quad (1.53)$$

where  $N_i^h$  is the moles of component i in the hydrocarbon phase,  $N_i^a$  is the moles of component i in the aqueous phase and i is each hydrocarbon component.

$N_i^h$  is selected as the primary variable instead of the total number of moles. The solubility of hydrocarbon components in the aqueous phase needs to be determined using the unknowns during the Newton iterations of the governing equations. This is done as follows. In step 2 of the solution procedure described later under this task, the solubility of each component in the aqueous phase is calculated using Henry's law.

$$x_i^a = \frac{f_i^a}{H_i} = \frac{N_i^a}{N_w^a + \sum_{i=1}^{n_c} N_i^a} \quad (1.54)$$

The mole fraction of water in the aqueous phase is

$$x_w^a = 1 - \sum_{i=1}^{n_c} x_i^a = \frac{N_w^a}{N_w^a + \sum_{i=1}^{n_c} N_i^a} \quad (1.55)$$

Dividing Eq. 1.54 by Eq. 1.55 yields

$$N_i^a = \frac{x_i^a}{x_w^a} N_w^a \quad (1.56)$$

Since  $N_i^a$  can be determined using the primary variables as shown in the above equation, it is apparent that the inclusion of the solubility calculations doesn't increase the number of primary variables.

## Well Model

The well model accounts for the source/sink terms in the mass conservation equations. A reservoir simulator needs a well model to translate historical production/injection data and the actual or desired field operating conditions and constraints into acceptable controls for the reservoir model. This requires that the operating constraints such as production and injection facility limits be imposed correctly as boundary conditions for the individual gridblock representing part or all of a well in the reservoir model. The well-management routine assigns user-specified well rates or pressures to individual producing blocks in the simulator at specified times. The application of a constant-pressure boundary condition will hold the bottom hole production or injection pressure in the well constant. A well is located at the center of the gridblock containing the well. The equations given below apply only for vertical wells.

The relationship between volumetric flow rate, flowing bottom hole pressure and gridblock pressure is expressed as

$$Q_j = PI_j (P_{wf} - P_j) \quad (1.57)$$

where  $PI_j$  is the productivity index for phase  $j$ . For one-dimensional (x-direction) and x-z cross-sectional simulations, the productivity index is given by (Chang, 1990):

$$PI_j = \frac{\sqrt{k_x k_y} \Delta y \Delta z \lambda_{rj}}{25.15 * 2\pi \left( \frac{\Delta x}{2} \right)} \quad (1.58)$$

For two-dimensional areal and three-dimensional simulations, the phase productivity index is computed as

$$PI_j = \frac{\sqrt{k_x k_y} \Delta z \lambda_{rj}}{25.15 \ln \left( \frac{r_o}{r_w} \right)} \quad (1.59)$$

where  $k_x$  and  $k_y$  (md) are permeability in the x and y directions,  $\Delta x$ ,  $\Delta y$  and  $\Delta z$  are gridblock sizes (ft) in x, y and z directions,  $\lambda_{rj}$  is relative mobility in  $cp^{-1}$ ,  $Q_j$  is the well flow rate of phase  $j$  in  $ft^3/day$  and pressure is in psi. In Eq. 1.59, the radius  $r_o$  is given by Peaceman's equation:

$$r_o = 0.28 \frac{\left[ \left( \frac{k_x}{k_y} \right)^{1/2} \Delta x^2 + \left( \frac{k_x}{k_y} \right)^{1/2} \Delta y^2 \right]}{\left( \frac{k_y}{k_x} \right)^{1/4} + \left( \frac{k_x}{k_y} \right)^{1/4}} \quad (1.60)$$

### Constant Bottomhole Flowing Pressure Injector

The bottomhole flowing pressure at elevation  $z$  is calculated from this reference pressure as follows:

$$(P_{wf})_z = P_{bot} - \gamma_{inj} (z - z_{bot}) \quad (1.61)$$

where  $\gamma_{inj}$  is the specific weight of the injected fluid at the well pressure. The bottomhole reference depth for each well can be assigned in the input file.

The component flow rates for layer  $z$  are computed by

$$(q_i)_z = \left[1 - (f_1)_{inj}\right] (z_i)_{inj} (q_t)_z \text{ for } i = 1, \dots, n_c \quad (1.62)$$

$$(q_{n_c+1})_z = (f_1)_{inj} (q_t)_z \quad (1.63)$$

and

$$(q_t)_z = \frac{(Q_t)_z}{(v_t)_{inj}} \quad (1.64)$$

where

$$(Q_t)_z = \sum_{j=1}^{n_p} (PI_j)_z \left[ (P_{wf})_z - (P_j)_z \right] \quad (1.65)$$

and

$$(v_t)_{inj} = \left( \frac{f_1}{\xi_1} \right)_{inj} + \left[ \left( 1 - (f_1)_{inj} \right) \right] \left[ \left( \frac{L_2}{\xi_2} \right)_{inj} + \left( \frac{L_3}{\xi_3} \right)_{inj} \right] \quad (1.66)$$

where  $(\xi_j)_{inj}$  is the molar density of phase  $j$ ,  $(L_j)_{inj}$  is a ratio of moles in hydrocarbon phase  $j$  to the total number of hydrocarbon moles in the injection fluid and  $(f_1)_{inj}$  is the molar fraction of water component in the injection stream. The values of  $(\xi_j)_{inj}$  and  $(L_j)_{inj}$  for  $j = 2$  and  $3$  are determined by flash calculations at the time the well conditions are specified. The molar density for the aqueous phase, which is assumed to be slightly compressible, is calculated using

$$\xi_1 = \xi_1^0 \left[ 1 + c_1 (P - P_1^0) \right] \quad (1.67)$$

where  $\xi_1^0$  is the aqueous phase molar density at the reference pressure  $P_1^0$ .

### Constant Bottomhole Flowing Pressure Producer

The flowing bottomhole pressures for a layer  $z$  is calculated using

$$(P_{wf})_z = P_{bot} - \gamma_z (z - z_{bot}) \quad (1.68)$$

The layer component flow rate is found by

$$(q_i)_z = \sum_{j=2}^{n_p} (\xi_j x_{ij} PI_j)_z (P_{wf} - P_j)_z \text{ for } i = 1, \dots, n_c \quad (1.69)$$

and

$$(q_{n_c+1})_z = (\xi_l PI_l)_z (P_{wf} - P_l)_z \quad (1.70)$$

In GPAS, the main well model calculations are done in the subroutine XWELL, which calls the subroutines WELLRATE and PRDWDEN. The WELLRATE subroutine calculates the molar flow rates and volumetric flow rates of each component in each layer and for each well. At present, only the constant bottomhole flowing pressure constraint is implemented. The productivity index is calculated in the subroutine IWELL. The wellbore fluid density for production wells required to calculate the flowing well production pressure at each layer is computed in the subroutine PRDWDEN.

## Overall Computation Procedure of the Simulator

### Primary and secondary variables

The independent variables for solving the governing equations are  $P, N_1, N_2, \dots, N_{n_c}, \ln K_1, \ln K_2, \dots, \ln K_{n_c}$ . The equilibrium ratio,  $K_i$  is defined as

$$K_i = \frac{y_i}{x_i} \quad (1.71)$$

where  $x_i$  and  $y_i$  are the mole fractions of component  $i$  in oil and gas phases, respectively.

The set of the independent variables is further classified into the primary and secondary variables. The primary variables are  $P, N_1, N_2, \dots, N_{n_c}$  because they are coupled between adjacent cells through the component mass-balance equations and need to be solved simultaneously. The  $\ln K$  values are the secondary variables because they are calculated from the equality of the component fugacities at given pressure, temperature and total moles of all components.

### Solution Procedure

A fully implicit solution method is used to solve the governing equations. The equations are nonlinear and must be solved iteratively. A Newton procedure is used in which the system of nonlinear equations is approximated by a system of linear equations. The



linearization is performed by using the Jacobian Matrix of the governing equations. The Jacobian refers to the matrix whose elements are the derivatives of the governing equations with respect to the independent variables.

The sequence of steps involved in the solution of the governing equations for the independent variables over a timestep include:

1. *Initialization in Each Gridblock:* The Pressure, overall composition and temperature of the fluids in each gridblock are specified. The initialization and calculation of the initial fluid in place is done in the subroutine INFLUID0. This subroutine is the main driver for the computation of the initial fluid in place.
2. *Phase identification and Physical Properties Calculation:* The flash calculations are performed in each gridblock and the phase saturations, compositions and densities are determined. The phases are then identified as gas, oil or aqueous phase. Phase viscosities and relative permeabilities are subsequently computed. The flash calculations are performed in the subroutine XFLASH and currently are limited to two phases. The Rachford-Rice equation that determines the phase fractions is coded in the subroutine SOLVE. All the fluid physical properties like the fluid viscosity, relative permeability and phase density calculations are determined in the PROP subroutine. The PROP subroutine calls separate routines to determine the different physical properties. It calls the subroutine LVISC2 to calculate the phase viscosity using the Lorenz coefficient, the subroutine RELPERM3EOS to calculate the three-phase relative permeability using Stone's model or calls the LOOKUP subroutine to interpret the two-phase relative permeability from the relative permeability tables.
3. *Governing Equations Linearization:* All the governing equations are linearized in terms of the independent variables and the elements of the Jacobian are calculated. The subroutine JACOBIAN that in turn calls PREROW generates the Jacobian for the linear system. PREROW is the main subroutine that calls the other subroutines, each with a specific task of computing the derivatives of separate equations and terms. The subroutine JACCUM calculates the derivatives of the accumulation term of the component balance. The subroutine JACO2 calculates the derivatives related to transmissibility terms. The subroutine JMASS calculates the derivatives related to the component mass balance in X, Y and Z directions. The derivatives of the source/sink terms are calculated in JSOURCE.
4. *Jacobian Factorization and Reduction of the Linear Systems:* A row elimination is performed to reduce the size of the linear system from  $2n_c + 1$  to  $n_c$  for each gridblock. To achieve this, the linearized phase-equilibrium relations and the linearized volume constraint are used to eliminate the secondary variables and one of the overall component moles from the linearized component mass balance equations. The subroutine EOS\_JACO forms the Jacobian for row elimination for two-phase cells.

5. *Solution of the Reduced System of the Linear Equations for the Primary Variables:* The reduced system of linear equations is simultaneously solved for pressure and the overall moles of  $n_c - 1$  components per unit bulk volume for all the cells.
6. *Secondary Variables Calculation:* A back substitution method is employed to compute the secondary variables  $\ln K$  and overall moles of the component eliminated in Step 4 using the factorized Jacobian. The phase-stability analysis is then carried out for all the gridblocks using the newly updated pressure and overall component moles.
7. *Updating Phase Densities and Viscosities, Determination of Single-Phase State and Estimation of Phase Relative Permeability:* The subroutine XUPDATE updates the phase composition and the phase properties as phase density, viscosity, relative permeability and determination of single phase in each cell. The main subroutine in the compositional model EOSCOMP is XSTEP. XSTEP calls XDELTA, which in turn calls the XUPDATE subroutine.
8. *Check for Convergence:* The residuals of the linear system obtained in Step 3 are used to determine convergence. If a tolerance is exceeded, the elements of the Jacobian and the residuals of the governing equations are then updated and another Newton iteration is performed by returning to Step 4. If the tolerance is met, a new timestep is then started by returning to Step 3. The subroutine XSTEP is the contact routine between the IPARS framework code and the EOSCOMP compositional model code. The residuals are checked in XSTEP and if the tolerance is met, a new timestep is started, else another Newton iteration is performed.

## Executive Routines in EOSCOMP

The executive routines in the EOSCOMP model are as follows:

- XISDAT All the initial scalar data are read by this subroutine. These include the physical properties of each component, default number of iterations, convergence tolerances for a variety of calculations, output flags, operation specific flags and chemical property data. No grid-element arrays can be reference in this subroutine.
- XARRAY This subroutine allocates memory for all the grid element arrays.
- XIADAT The entire grid element array input as the pressure, water saturation, feed composition is read in and written out to a file.
- XIVDAT. Performs the model initialization before time iteration. The PETSc linear solver is also initialized.

XSTEP	The main subroutine that performs all the calculations over a timestep.
XQUIT	Exits from the simulation when it meets the maximum time, production limits, or if an error occurs.

The communication between processors for the compositional model is performed in the executive subroutines. There is no argument attached to these calls. Those variables associated with grids are passed into these routines through pointers that are stored in common block. These common blocks are included as header files in the subroutines. The executive routines call the work routines to perform all the calculations. The grid dimensions and variables are passed into these work routines through a C routine called CALLWORK, which is handled by the framework. The CALLWORK function passes the variables as an index argument list to the function being called.

## Description of the Solver

PETSc (Balay *et al.*, 1997; Wang *et al.*, 1999) is a large suite of parallel, general-purpose, object-oriented solvers for the scalable solution of partial differential equations discretized using implicit and semi-implicit methods. PETSc is implemented in C, and is usable from C, Fortran, and C<sup>++</sup>. It uses MPI for communication across processors. GPAS uses the linear solver component of PETSc to solve the linearized Newton system of equations and uses the parallel data formats provided by PETSc to store the Jacobian and the vectors.

The linear solver components of PETSc provides a unified interface to various Krylov methods, such as conjugate gradient (CG), generalized minimal residual (GMRES), biconjugate gradient, etc. and also to various parallel preconditioners such as Jacobi, block preconditioners like block Jacobi, domain decomposition preconditioners like additive Schwartz. GPAS uses the biconjugate gradient stabilized approach as the Krylov method and block Jacobi preconditioner, with point block incomplete factorization (ILU) on the subdomain blocks. The point block refers to treating all the variables associated with a single gridblock as a single unit. The number of subdomain blocks for block Jacobi is chosen to match the number of processors used, so that each processor gets a complete subdomain of the problem and does a single local incomplete factorization on the Jacobian corresponding to this subdomain.

For three-phase flow, the compositional model EOSCOMP generates  $2n_c + 1$  equations per gridblock causing the Jacobian to have a point-block structure and a point-block sparse storage format is used to store the matrix. These  $2n_c + 1$  equations do not result in complete coupling of all the variables across gridblocks. This causes the Jacobian to have some  $n_c + 1$  point-block locations with zero values. Thus a block size of  $n_c + 1$  is chosen for this matrix type eliminating the need to store the  $(n_c + 1) (n_c + 1)$  blocks

with zero values. The usage of the point-block sparse matrix storage leads to the improvement in the performance of the matrix routines.

## Solution Approach in Chemical Module

The chemical species were added using two different numerical methods. The species included in the chemical module were any number of conservative and partitioning tracers, electrolytes, polymer, and surfactant. In the first method, we solved the mass balance equations for the aqueous species explicitly. We refer to this formulation as the hybrid method. The chemical module was linked to the equation-of-state compositional model EOSCOMP in an explicit manner. After EOSCOMP solves for the pressures, saturations, and compositions of the non-aqueous species components for a particular time step and the convergence for the mass balance equations is attained, the chemical subroutine imports the required input from the host EOSCOMP and solves for the aqueous species mass balance equation to find the concentration at a given point in space and time. This decoupled approach is more computationally efficient than solving all of the equations simultaneously in EOSCOMP.

We, then, formulated and implemented the chemical species in a fully implicit method to take advantage of the larger time steps attainable in the fully implicit formulation. The details of these procedures are given in Task 2.

## Non-Orthogonal Grid

In the last six months of the project, we have implemented a cornerpoint non-orthogonal grid option to model more realistic reservoir geometries and be able to include the curved reservoir boundaries and impermeable barriers such as shales and faults.

The mass balance equation for i-th component assuming diagonal permeability tensor can be written in a Cartesian system as

$$\begin{aligned} \frac{\partial(\phi N_i)}{\partial t} - \frac{\partial}{\partial x} \left( \sum_{j=1}^{np} \xi_j x_{ij} \lambda_j K_{xx} \frac{\partial \Phi_j}{\partial x} \right) - \frac{\partial}{\partial y} \left( \sum_{j=1}^{np} \xi_j x_{ij} \lambda_j K_{yy} \frac{\partial \Phi_j}{\partial y} \right) \\ - \frac{\partial}{\partial z} \left( \sum_{j=1}^{np} \xi_j x_{ij} \lambda_j K_{zz} \frac{\partial \Phi_j}{\partial z} \right) - q_i = 0 \end{aligned} \quad (1.72)$$

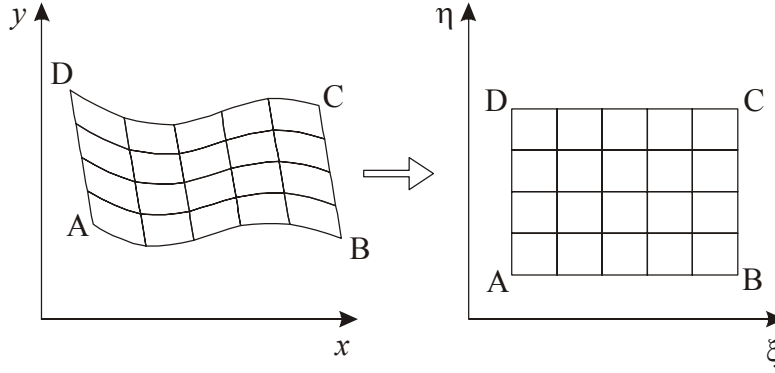
where  $\Phi_j$  is the potential of j phase and is given by

$$\Phi_j = P_j - \gamma_j D \quad (1.73)$$

In Eq. 1.73  $P_j$  denotes the pressure of j phase and  $D$  is depth which is positive in downward direction. Equation 1.72 can be written in a boundary fitted coordinate system using the following transformation

$$\xi = \xi(x, y, z) ; \eta = \eta(x, y, z) ; \gamma = \gamma(x, y, z) \quad (1.74)$$

Using the above mentioned relations Eq. 1.72 and its counterparts can be solved in a rectangular domain that is usually called computational domain, Thompson *et al.* (1985) and Maliska (2004). Figure 1.1 shows a physical domain and computational domain for a two-dimensional case.



**Figure 1.1—Physical and computational domains.**

We define some of the terms in Eq. 1.72 as

$$E = \xi_j x_{ij} \lambda_j K_{xx} \frac{\partial \Phi_j}{\partial x} = \Gamma_{xij} \frac{\partial \Phi_j}{\partial x} ;$$

$$F = \xi_j x_{ij} \lambda_j K_{yy} \frac{\partial \Phi_j}{\partial y} = \Gamma_{yij} \frac{\partial \Phi_j}{\partial y} ; \quad (1.75)$$

$$G = \xi_j x_{ij} \lambda_j K_{zz} \frac{\partial \Phi_j}{\partial z} = \Gamma_{zij} \frac{\partial \Phi_j}{\partial z}$$

Applying the rule chain, the gradients of E, F, and G in x, y, and z directions the following equations are rendered.

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial E}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial E}{\partial \gamma} \frac{\partial \gamma}{\partial x}$$

$$\frac{\partial F}{\partial y} = \frac{\partial F}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial F}{\partial \eta} \frac{\partial \eta}{\partial y} + \frac{\partial F}{\partial \gamma} \frac{\partial \gamma}{\partial y} \quad (1.76)$$

$$\frac{\partial G}{\partial z} = \frac{\partial G}{\partial \xi} \frac{\partial \xi}{\partial z} + \frac{\partial G}{\partial \eta} \frac{\partial \eta}{\partial z} + \frac{\partial G}{\partial \gamma} \frac{\partial \gamma}{\partial z}$$

Dividing each term of Eq. 1.76 by  $J_t$ , the Jacobian of the transformation, and replacing the result in Eq. 1.72, and adding and subtracting terms of type  $E \frac{\partial}{\partial \xi} \left( \frac{\xi_x}{J_t} \right), F \frac{\partial}{\partial \xi} \left( \frac{\xi_y}{J_t} \right), G \frac{\partial}{\partial \xi} \left( \frac{\xi_z}{J_t} \right)$  and after simplifications, the following conservative equation is obtained:

$$\begin{aligned}
& \frac{\partial}{\partial t} \left( \frac{\phi N_i}{J_t} \right) - \frac{\partial}{\partial \xi} \left( \frac{\xi_x E + \xi_y F + \xi_z G}{J_t} \right) - \frac{\partial}{\partial \eta} \left( \frac{\eta_x E + \eta_y F + \eta_z G}{J_t} \right) - \frac{\partial}{\partial \gamma} \left( \frac{\gamma_x E + \gamma_y F + \gamma_z G}{J_t} \right) \\
& - E \left[ \frac{\partial}{\partial \xi} \left( \frac{\xi_x}{J_t} \right) + \frac{\partial}{\partial \eta} \left( \frac{\eta_x}{J_t} \right) + \frac{\partial}{\partial \gamma} \left( \frac{\gamma_x}{J_t} \right) \right] - F \left[ \frac{\partial}{\partial \xi} \left( \frac{\xi_y}{J_t} \right) + \frac{\partial}{\partial \eta} \left( \frac{\eta_y}{J_t} \right) + \frac{\partial}{\partial \gamma} \left( \frac{\gamma_y}{J_t} \right) \right] \\
& - G \left[ \frac{\partial}{\partial \xi} \left( \frac{\xi_z}{J_t} \right) + \frac{\partial}{\partial \eta} \left( \frac{\eta_z}{J_t} \right) + \frac{\partial}{\partial \gamma} \left( \frac{\gamma_z}{J_t} \right) \right] - \frac{q_i}{J_t} = 0
\end{aligned} \tag{1.77}$$

It is easy to demonstrate that the terms in blankets in Eq. 1.77 are equal to zero. Details can be found in Maliska (2004). Applying the chain rule to E, F, and G and inserting the results into Eq. 1.77, we obtain

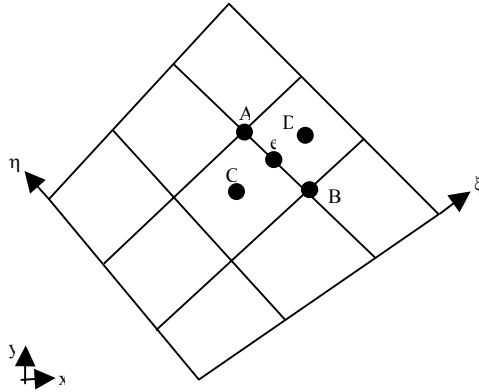
$$\begin{aligned}
& \frac{\partial}{\partial t} \left( \frac{\phi N_i}{J_t} \right) - \frac{\partial}{\partial \xi} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \xi_x^2 \Gamma_{xij} + \xi_y^2 \Gamma_{yij} + \xi_z^2 \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \xi} \right] - \frac{\partial}{\partial \xi} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \xi_x \eta_x \Gamma_{xij} + \xi_y \eta_y \Gamma_{yij} + \xi_z \eta_z \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \eta} \right] \\
& - \frac{\partial}{\partial \xi} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \xi_x \gamma_x \Gamma_{xij} + \xi_y \gamma_y \Gamma_{yij} + \xi_z \gamma_z \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \gamma} \right] - \frac{\partial}{\partial \eta} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \xi_x \eta_x \Gamma_{xij} + \xi_y \eta_y \Gamma_{yij} + \xi_z \eta_z \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \xi} \right] \\
& - \frac{\partial}{\partial \eta} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \eta_x^2 \Gamma_{xij} + \eta_y^2 \Gamma_{yij} + \eta_z^2 \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \eta} \right] - \frac{\partial}{\partial \eta} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \eta_x \gamma_x \Gamma_{xij} + \eta_y \gamma_y \Gamma_{yij} + \eta_z \gamma_z \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \gamma} \right] \\
& - \frac{\partial}{\partial \gamma} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \xi_x \gamma_x \Gamma_{xij} + \xi_y \gamma_y \Gamma_{yij} + \xi_z \gamma_z \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \xi} \right] - \frac{\partial}{\partial \gamma} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \eta_x \gamma_x \Gamma_{xij} + \eta_y \gamma_y \Gamma_{yij} + \eta_z \gamma_z \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \gamma} \right] \\
& - \frac{\partial}{\partial \gamma} \left[ \sum_{j=1}^{np} \frac{1}{J_t} \left( \gamma_x^2 \Gamma_{xij} + \gamma_y^2 \Gamma_{yij} + \gamma_z^2 \Gamma_{zij} \right) \frac{\partial \Phi_j}{\partial \gamma} \right] - \frac{q_i}{J_t} = 0
\end{aligned} \tag{1.78}$$

where the direct metrics and the Jacobian ( $J_t$ ) of the transformation can be evaluated as a function of the inverse metrics by

$$\begin{aligned}
\frac{\xi_x^2}{J_t} &= J_t (y_\eta z_\gamma - y_\gamma z_\eta)^2 ; \quad \frac{\xi_y^2}{J_t} = J_t (x_\eta z_\gamma - x_\gamma z_\eta)^2 ; \quad \frac{\xi_z^2}{J_t} = J_t (x_\eta y_\gamma - x_\gamma y_\eta)^2 \\
\frac{\xi_x \eta_x}{J_t} &= -J_t (y_\eta z_\gamma - y_\gamma z_\eta) (y_\xi z_\gamma - y_\gamma z_\xi) ; \quad \frac{\xi_y \eta_y}{J_t} = -J_t (x_\eta z_\gamma - x_\gamma z_\eta) (x_\xi z_\gamma - x_\gamma z_\xi) \\
\frac{\xi_z \eta_z}{J_t} &= -J_t (x_\eta y_\gamma - x_\gamma y_\eta) (x_\xi y_\gamma - x_\gamma y_\xi) ; \quad \frac{\eta_x^2}{J_t} = J_t (y_\xi z_\gamma - y_\gamma z_\xi)^2 ; \\
\frac{\eta_y^2}{J_t} &= J_t (x_\xi z_\gamma - x_\gamma z_\xi)^2 \\
\frac{\eta_z^2}{J_t} &= J_t (x_\xi y_\gamma - x_\gamma y_\xi)^2 ; \quad \frac{\xi_x \gamma_x}{J_t} = J_t (y_\eta z_\gamma - y_\gamma z_\eta) (y_\xi z_\eta - y_\eta z_\xi) \\
\frac{\xi_y \gamma_y}{J_t} &= J_t (x_\eta z_\gamma - x_\gamma z_\eta) (x_\xi z_\eta - x_\eta z_\xi) ; \quad \frac{\xi_z \gamma_z}{J_t} = J_t (x_\eta y_\gamma - x_\gamma y_\eta) (x_\xi y_\eta - x_\eta y_\xi) \\
\frac{\gamma_x^2}{J_t} &= J_t (y_\xi z_\eta - y_\eta z_\xi)^2 ; \quad \frac{\gamma_y^2}{J_t} = J_t (x_\xi z_\eta - x_\eta z_\xi)^2 ; \quad \frac{\gamma_z^2}{J_t} = J_t (x_\xi y_\eta - x_\eta y_\xi)^2 \\
\frac{\eta_x \gamma_x}{J_t} &= J_t (y_\xi z_\gamma - y_\gamma z_\xi) (y_\xi z_\eta - y_\eta z_\xi) ; \quad \frac{\eta_y \gamma_y}{J_t} = -J_t (x_\xi z_\gamma - x_\gamma z_\xi) (x_\xi z_\eta - x_\eta z_\xi) \\
\frac{\eta_z \gamma_z}{J_t} &= -J_t (x_\xi y_\gamma - x_\gamma y_\xi) (x_\xi y_\eta - x_\eta y_\xi) \\
J_t &= \left( x_\xi (y_\eta z_\gamma - y_\gamma z_\eta) - x_\eta (y_\xi z_\gamma - y_\gamma z_\xi) + x_\gamma (y_\xi z_\eta - y_\eta z_\xi) \right)^{-1}
\end{aligned} \tag{1.79}$$

After the mesh generation, the expressions in Eq. 1.79 can be easily evaluated using the finite difference method. For instance, for the 2D case shown in Fig. 1.2,  $x_{\xi e}$ ,  $y_{\xi e}$ ,  $x_{\eta e}$ , and  $y_{\eta e}$  are given by

$$\left. \frac{\partial x}{\partial \xi} \right|_e = \frac{x_D - x_C}{\Delta \xi} ; \quad \left. \frac{\partial y}{\partial \xi} \right|_e = \frac{y_D - y_C}{\Delta \xi} ; \quad \left. \frac{\partial x}{\partial \eta} \right|_e = \frac{x_A - x_B}{\Delta \eta} ; \quad \left. \frac{\partial y}{\partial \eta} \right|_e = \frac{y_A - y_B}{\Delta \eta} \tag{1.80}$$



**Figure 1.2—Metrics evaluation.**

The coordinates at points C and D are obtained by interpolating the coordinates at the four neighbor corners. According to Maliska (2004) whenever coordinates are not available at the specific position, for example points C and D in Fig. 1.2, an interpolation procedure should be done instead of using an interpolation in metrics. A similar procedure is employed to calculate the metrics in others positions of the domain.

### Approximate Equation

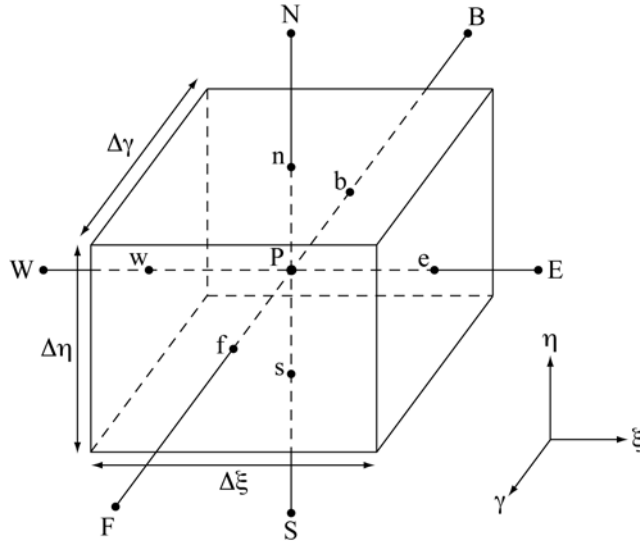
Integrating Eq. 1.78 in time and space (control volume of Fig. 1.3) and considering a fully implicit formulation for all terms, except the derivatives of the cross terms, and dividing each term by  $\Delta t$ , the following equation is obtained

$$\begin{aligned}
& \left( \frac{\phi N_i}{J_t} \right)_P^{n+1} \frac{\Delta V}{\Delta t} - \left( \frac{\phi N_i}{J_t} \right)_P^o \frac{\Delta V}{\Delta t} - \sum_{j=1}^{np} D_{11ij,e}^{n+1} \frac{\partial \Phi_j}{\partial \xi} \bigg|_e^{n+1} \Delta \eta \Delta \gamma \\
& + \sum_{j=1}^{np} D_{11ij,w}^{n+1} \frac{\partial \Phi_j}{\partial \xi} \bigg|_w^{n+1} \Delta \eta \Delta \gamma - \sum_{j=1}^{np} D_{12ij,e}^{n+1} \frac{\partial \Phi_j}{\partial \eta} \bigg|_e^n \Delta \eta \Delta \gamma + \sum_{j=1}^{np} D_{12ij,w}^{n+1} \frac{\partial \Phi_j}{\partial \eta} \bigg|_w^n \Delta \eta \Delta \gamma \\
& - \sum_{j=1}^{np} D_{13ij,e}^{n+1} \frac{\partial \Phi_j}{\partial \gamma} \bigg|_e^n \Delta \eta \Delta \gamma + \sum_{j=1}^{np} D_{13ij,w}^{n+1} \frac{\partial \Phi_j}{\partial \gamma} \bigg|_w^n \Delta \eta \Delta \gamma - \sum_{j=1}^{np} D_{21ij,n}^{n+1} \frac{\partial \Phi_j}{\partial \xi} \bigg|_n^n \Delta \xi \Delta \gamma \\
& + \sum_{j=1}^{np} D_{21ij,s}^{n+1} \frac{\partial \Phi_j}{\partial \xi} \bigg|_s^n \Delta \xi \Delta \gamma - \sum_{j=1}^{np} D_{22ij,n}^{n+1} \frac{\partial \Phi_j}{\partial \eta} \bigg|_n^{n+1} \Delta \xi \Delta \gamma + \sum_{j=1}^{np} D_{22ij,s}^{n+1} \frac{\partial \Phi_j}{\partial \eta} \bigg|_s^{n+1} \Delta \xi \Delta \gamma \\
& - \sum_{j=1}^{np} D_{23ij,n}^{n+1} \frac{\partial \Phi_j}{\partial \gamma} \bigg|_n^n \Delta \xi \Delta \gamma + \sum_{j=1}^{np} D_{23ij,s}^{n+1} \frac{\partial \Phi_j}{\partial \gamma} \bigg|_s^n \Delta \xi \Delta \gamma - \sum_{j=1}^{np} D_{31ij,f}^{n+1} \frac{\partial \Phi_j}{\partial \xi} \bigg|_f^n \Delta \xi \Delta \eta \\
& + \sum_{j=1}^{np} D_{31ij,b}^{n+1} \frac{\partial \Phi_j}{\partial \xi} \bigg|_b^n \Delta \xi \Delta \eta - \sum_{j=1}^{np} D_{32ij,f}^{n+1} \frac{\partial \Phi_j}{\partial \eta} \bigg|_f^n \Delta \xi \Delta \eta + \sum_{j=1}^{np} D_{32ij,b}^{n+1} \frac{\partial \Phi_j}{\partial \eta} \bigg|_b^n \Delta \xi \Delta \eta \\
& - \sum_{j=1}^{np} D_{33ij,f}^{n+1} \frac{\partial \Phi_j}{\partial \gamma} \bigg|_f^{n+1} \Delta \xi \Delta \eta + \sum_{j=1}^{np} D_{33ij,b}^{n+1} \frac{\partial \Phi_j}{\partial \gamma} \bigg|_b^{n+1} \Delta \xi \Delta \eta - \frac{q_{i,P}^{n+1} \Delta V}{J_{t,P}} = 0
\end{aligned} \tag{1.81}$$

where  $J_{t,P}$  means that the Jacobian of the transformation is evaluated at point P. The tensor D involves geometric information and fluid properties. For example, i denote component, and j phase. It should be observed that each component of this tensor is given by



$$\begin{aligned}
D_{11ij} &= \frac{1}{J_t} \left( \xi_x^2 \Gamma_{xij} + \xi_y^2 \Gamma_{yij} + \xi_z^2 \Gamma_{zij} \right) \\
D_{12ij} &= \frac{1}{J_t} \left( \xi_x \eta_x \Gamma_{xij} + \xi_y \eta_y \Gamma_{yij} + \xi_z \eta_z \Gamma_{zij} \right) \\
D_{13ij} &= \frac{1}{J_t} \left( \xi_x \gamma_x \Gamma_{xij} + \xi_y \gamma_y \Gamma_{yij} + \xi_z \gamma_z \Gamma_{zij} \right) \\
D_{21ij} &= D_{12ij} = \frac{1}{J_t} \left( \xi_x \gamma_x \Gamma_{xij} + \xi_y \gamma_y \Gamma_{yij} + \xi_z \gamma_z \Gamma_{zij} \right) \\
D_{22ij} &= \frac{1}{J_t} \left( \eta_x^2 \Gamma_{xij} + \eta_y^2 \Gamma_{yij} + \eta_z^2 \Gamma_{zij} \right) \\
D_{23ij} &= \frac{1}{J_t} \left( \eta_x \gamma_x \Gamma_{xij} + \eta_y \gamma_y \Gamma_{yij} + \eta_z \gamma_z \Gamma_{zij} \right) \\
D_{31ij} &= D_{13ij} = \frac{1}{J_t} \left( \xi_x \gamma_x \Gamma_{xij} + \xi_y \gamma_y \Gamma_{yij} + \xi_z \gamma_z \Gamma_{zij} \right) \\
D_{32ij} &= D_{23ij} = \frac{1}{J_t} \left( \eta_x \gamma_x \Gamma_{xij} + \eta_y \gamma_y \Gamma_{yij} + \eta_z \gamma_z \Gamma_{zij} \right) \\
D_{33ij} &= \frac{1}{J_t} \left( \gamma_x^2 \Gamma_{xij} + \gamma_y^2 \Gamma_{yij} + \gamma_z^2 \Gamma_{zij} \right)
\end{aligned} \tag{1.82}$$



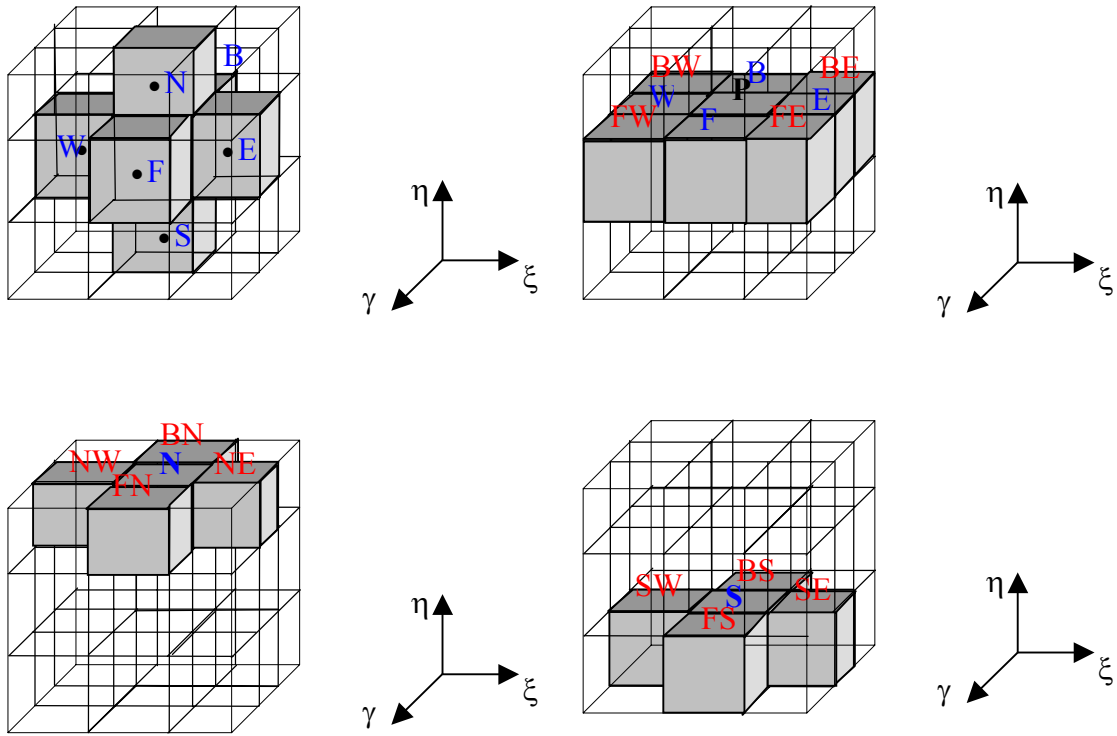
**Figure 1.3—Elemental control volume.**

It is worth to note that all physical properties in each interface are evaluated in time level  $n+1$ , while the cross derivatives are evaluated in time level  $n$ . Even though the convergence rate can be reduced when the mesh is much distorted, this procedure renders a Jacobian matrix structure similar to those of Cartesian or orthogonal corner point grids, but is much more flexible since it can be used for domains with highly irregular shapes.

The derivatives in Eq. 1.81 are evaluated using a central difference scheme. Figure 1.4 shows all the neighbors volumes for the control volume of Fig. 1.3. Using Fig. 1.4, the derivatives in each face of volume P are given by

$$\begin{aligned}
\left. \frac{\partial \Phi_j}{\partial \xi} \right|_e &= \frac{\Phi_{jE} - \Phi_{jP}}{\Delta \xi} ; \quad \left. \frac{\partial \Phi_j}{\partial \xi} \right|_w = \frac{\Phi_{jP} - \Phi_{jW}}{\Delta \xi} ; \\
\left. \frac{\partial \Phi_j}{\partial \eta} \right|_n &= \frac{\Phi_{jN} - \Phi_{jP}}{\Delta \eta} ; \quad \left. \frac{\partial \Phi_j}{\partial \eta} \right|_s = \frac{\Phi_{jP} - \Phi_{jS}}{\Delta \eta} ; \\
\left. \frac{\partial \Phi_j}{\partial \gamma} \right|_f &= \frac{\Phi_{jF} - \Phi_{jP}}{\Delta \gamma} ; \quad \left. \frac{\partial \Phi_j}{\partial \gamma} \right|_b = \frac{\Phi_{jP} - \Phi_{jB}}{\Delta \gamma}
\end{aligned} \tag{1.83}$$

$$\begin{aligned}
\left. \frac{\partial \Phi_j}{\partial \eta} \right|_e &= \frac{\Phi_{jN} + \Phi_{jNE} - \Phi_{jS} - \Phi_{jSE}}{4 \Delta \eta} ; \quad \left. \frac{\partial \Phi_j}{\partial \eta} \right|_w = \frac{\Phi_{jN} + \Phi_{jNW} - \Phi_{jS} - \Phi_{jSW}}{4 \Delta \eta} \\
\left. \frac{\partial \Phi_j}{\partial \gamma} \right|_e &= \frac{\Phi_{jF} + \Phi_{jFE} - \Phi_{jB} - \Phi_{jBE}}{4 \Delta \gamma} ; \quad \left. \frac{\partial \Phi_j}{\partial \gamma} \right|_w = \frac{\Phi_{jF} + \Phi_{jFW} - \Phi_{jB} - \Phi_{jBW}}{4 \Delta \gamma} \\
\left. \frac{\partial \Phi_j}{\partial \xi} \right|_n &= \frac{\Phi_{jE} + \Phi_{jNE} - \Phi_{jW} - \Phi_{jNW}}{4 \Delta \xi} ; \quad \left. \frac{\partial \Phi_j}{\partial \xi} \right|_s = \frac{\Phi_{jE} + \Phi_{jSE} - \Phi_{jW} - \Phi_{jSW}}{4 \Delta \xi} \\
\left. \frac{\partial \Phi_j}{\partial \gamma} \right|_n &= \frac{\Phi_{jF} + \Phi_{jFN} - \Phi_{jB} - \Phi_{jBN}}{4 \Delta \gamma} ; \quad \left. \frac{\partial \Phi_j}{\partial \gamma} \right|_s = \frac{\Phi_{jF} + \Phi_{jFS} - \Phi_{jB} - \Phi_{jBS}}{4 \Delta \gamma} \\
\left. \frac{\partial \Phi_j}{\partial \xi} \right|_f &= \frac{\Phi_{jE} + \Phi_{jFE} - \Phi_{jW} - \Phi_{jFW}}{4 \Delta \xi} ; \quad \left. \frac{\partial \Phi_j}{\partial \xi} \right|_b = \frac{\Phi_{jE} + \Phi_{jBE} - \Phi_{jW} - \Phi_{jBW}}{4 \Delta \xi} \\
\left. \frac{\partial \Phi_j}{\partial \eta} \right|_f &= \frac{\Phi_{jN} + \Phi_{jFN} - \Phi_{jS} - \Phi_{jFS}}{4 \Delta \eta} ; \quad \left. \frac{\partial \Phi_j}{\partial \eta} \right|_b = \frac{\Phi_{jN} + \Phi_{jBN} - \Phi_{jS} - \Phi_{jBS}}{4 \Delta \eta}
\end{aligned} \tag{1.84}$$



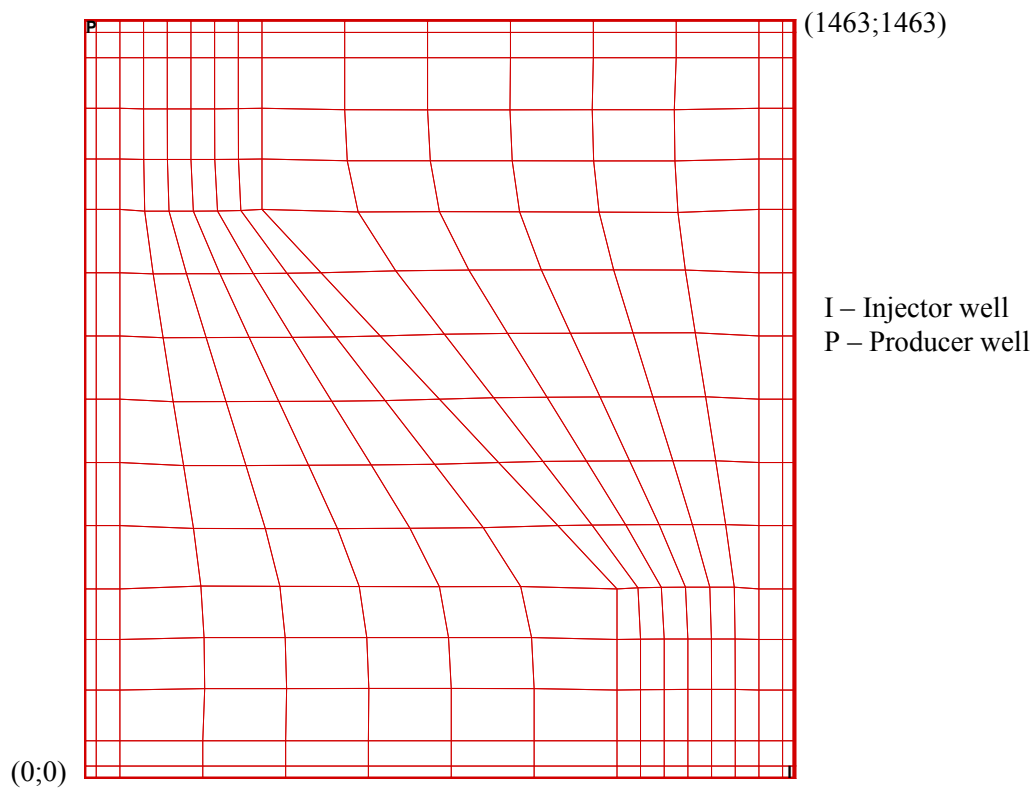
**Figure 1.4—Control volume P and its neighbors.**

Inserting the derivatives of Eqs. 1.83 and 1.84 into Eq. 1.81 result in the following residual equation for the  $i^{\text{th}}$  component

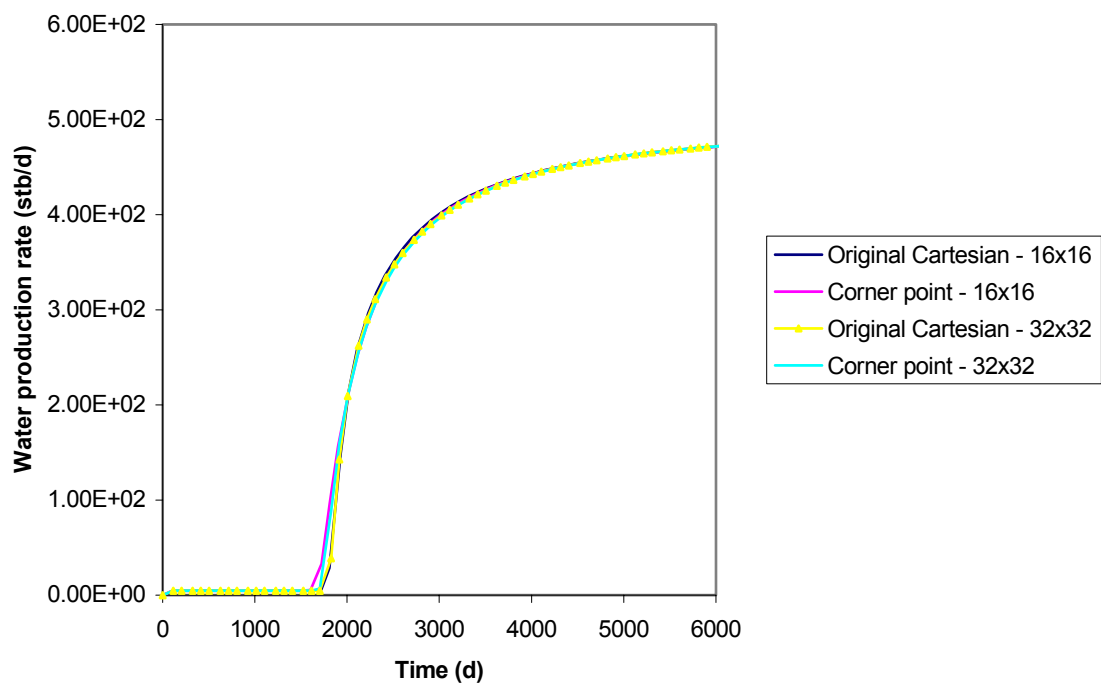
$$\begin{aligned}
R_{iP} = & \left( \frac{\phi N_i}{J} \right)_P^{n+1} \frac{\Delta V}{\Delta t} - \left( \frac{\phi N_i}{J} \right)_P^o \frac{\Delta V}{\Delta t} - \sum_{j=1}^{np} D_{1lij,e}^{n+1} \left( \frac{\Phi_{jE}^{n+1} - \Phi_{jP}^{n+1}}{\Delta \xi} \right) \Delta \eta \Delta \gamma \\
& + \sum_{j=1}^{np} D_{1lij,w}^{n+1} \left( \frac{\Phi_{jP}^{n+1} - \Phi_{jW}^{n+1}}{\Delta \xi} \right) \Delta \eta \Delta \gamma - \sum_{j=1}^{np} D_{12ij,e}^{n+1} \left( \frac{\Phi_{jN}^n + \Phi_{jNE}^n - \Phi_{jS}^n - \Phi_{jSE}^n}{4 \Delta \eta} \right) \Delta \eta \Delta \gamma \\
& + \sum_{j=1}^{np} D_{12ij,w}^{n+1} \left( \frac{\Phi_{jN}^n + \Phi_{jNW}^n - \Phi_{jS}^n - \Phi_{jSW}^n}{4 \Delta \eta} \right) \Delta \eta \Delta \gamma - \sum_{j=1}^{np} D_{13ij,e}^{n+1} \left( \frac{\Phi_{jF}^n + \Phi_{jFE}^n - \Phi_{jB}^n - \Phi_{jBE}^n}{4 \Delta \gamma} \right) \Delta \eta \Delta \gamma \\
& + \sum_{j=1}^{np} D_{13ij,w}^{n+1} \left( \frac{\Phi_{jF}^n + \Phi_{jFW}^n - \Phi_{jB}^n - \Phi_{jBW}^n}{4 \Delta \gamma} \right) \Delta \eta \Delta \gamma - \sum_{j=1}^{np} D_{2lij,n}^{n+1} \left( \frac{\Phi_{jE}^n + \Phi_{jNE}^n - \Phi_{jW}^n - \Phi_{jNW}^n}{4 \Delta \xi} \right) \Delta \xi \Delta \gamma \\
& + \sum_{j=1}^{np} D_{2lij,s}^{n+1} \left( \frac{\Phi_{jE}^n + \Phi_{jSE}^n - \Phi_{jW}^n - \Phi_{jSW}^n}{4 \Delta \xi} \right) \Delta \xi \Delta \gamma - \sum_{j=1}^{np} D_{22ij,n}^{n+1} \left( \frac{\Phi_{jN}^{n+1} - \Phi_{jP}^{n+1}}{\Delta \eta} \right) \Delta \xi \Delta \gamma \\
& + \sum_{j=1}^{np} D_{22ij,s}^{n+1} \left( \frac{\Phi_{jP}^{n+1} - \Phi_{jS}^{n+1}}{\Delta \eta} \right) \Delta \xi \Delta \gamma - \sum_{j=1}^{np} D_{23ij,n}^{n+1} \left( \frac{\Phi_{jF}^n + \Phi_{jFN}^n - \Phi_{jB}^n - \Phi_{jBN}^n}{4 \Delta \gamma} \right) \Delta \xi \Delta \gamma \\
& + \sum_{j=1}^{np} D_{23ij,s}^{n+1} \left( \frac{\Phi_{jF}^n + \Phi_{jFS}^n - \Phi_{jB}^n - \Phi_{jBS}^n}{4 \Delta \gamma} \right) \Delta \xi \Delta \gamma - \sum_{j=1}^{np} D_{3lij,f}^{n+1} \left( \frac{\Phi_{jE}^n + \Phi_{jFE}^n - \Phi_{jW}^n - \Phi_{jFW}^n}{4 \Delta \xi} \right) \Delta \xi \Delta \eta \\
& + \sum_{j=1}^{np} D_{3lij,b}^{n+1} \left( \frac{\Phi_{jE}^n + \Phi_{jBE}^n - \Phi_{jW}^n - \Phi_{jBW}^n}{4 \Delta \xi} \right) \Delta \xi \Delta \eta - \sum_{j=1}^{np} D_{32ij,f}^{n+1} \left( \frac{\Phi_{jN}^n + \Phi_{jFN}^n - \Phi_{jS}^n - \Phi_{jFS}^n}{4 \Delta \eta} \right) \Delta \xi \Delta \eta \\
& + \sum_{j=1}^{np} D_{32ij,b}^{n+1} \left( \frac{\Phi_{jN}^n + \Phi_{jBN}^n - \Phi_{jS}^n - \Phi_{jBS}^n}{4 \Delta \eta} \right) \Delta \xi \Delta \eta - \sum_{j=1}^{np} D_{33ij,f}^{n+1} \left( \frac{\Phi_{jF}^{n+1} - \Phi_{jP}^{n+1}}{\Delta \gamma} \right) \Delta \xi \Delta \eta \\
& + \sum_{j=1}^{np} D_{33ij,b}^{n+1} \left( \frac{\Phi_{jF}^{n+1} - \Phi_{jP}^{n+1}}{\Delta \gamma} \right) \Delta \xi \Delta \eta - \frac{q_{i,P}^{n+1} \Delta V}{J_{tP}} = 0
\end{aligned} \tag{1.85}$$

The formulation and implementation of the corner point grid geometry in GPAS was verified by first comparing the results of a Cartesian grid with that shown in Fig. 1.5. The simulation is based on a quarter of a five spot pattern waterflood with the wells in the two corners of the simulation domain. The domain was discretized using 16x16 and 32x32 number of gridblocks in x and y directions, respectively. Water was injected in an initial oil saturated reservoir at the residual water saturation. The water production rates are given in Fig. 1.6. Figure 1.7 shows the comparison of oil production rate for these simulations. The results are very similar for the Cartesian and corner point grid configuration.

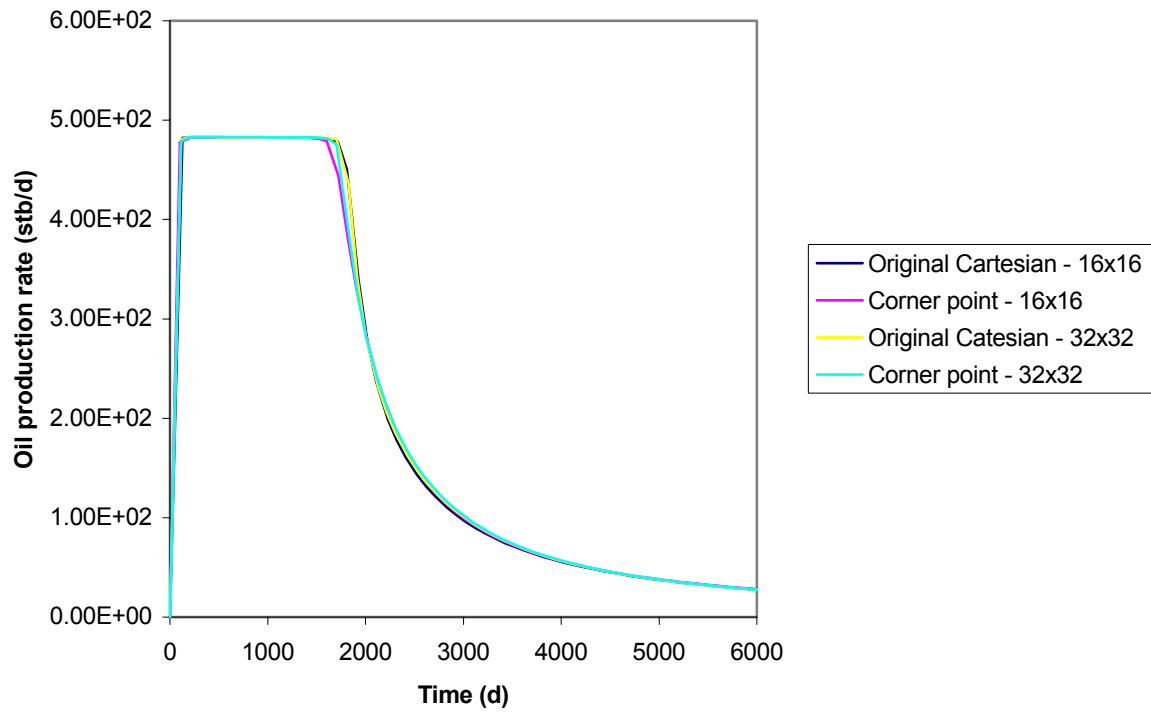
The second test was a waterflood simulation in the grid specified in Fig. 1.8. There are one injector and two production wells on a line drive pattern. Figures 1.9 and 1.10 show the cumulative water and oil production from the two producing wells.



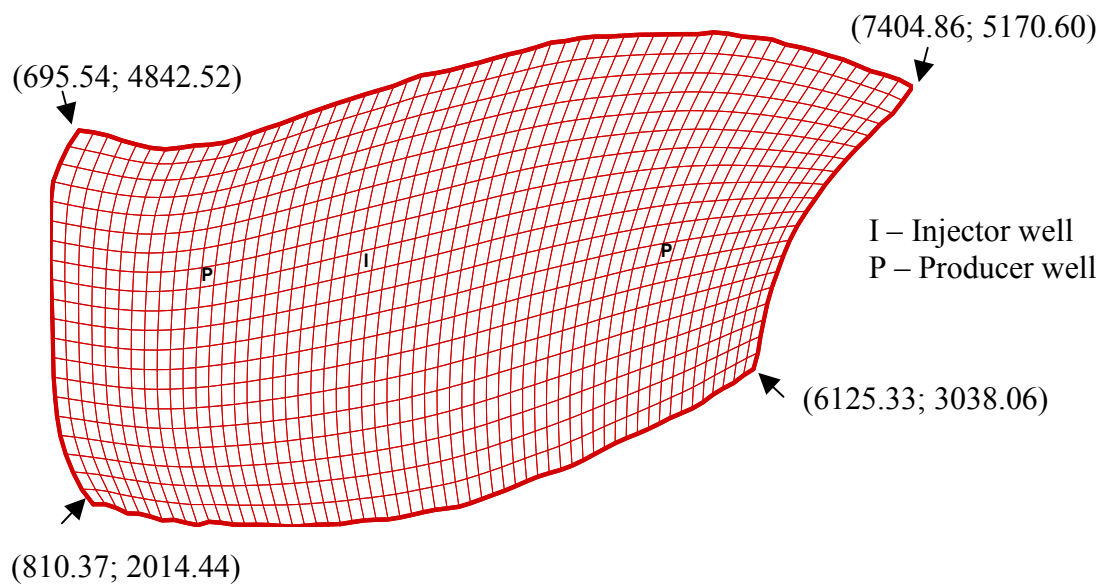
**Figure 1.5—Schematic of the grid used in Case 1 simulations.**



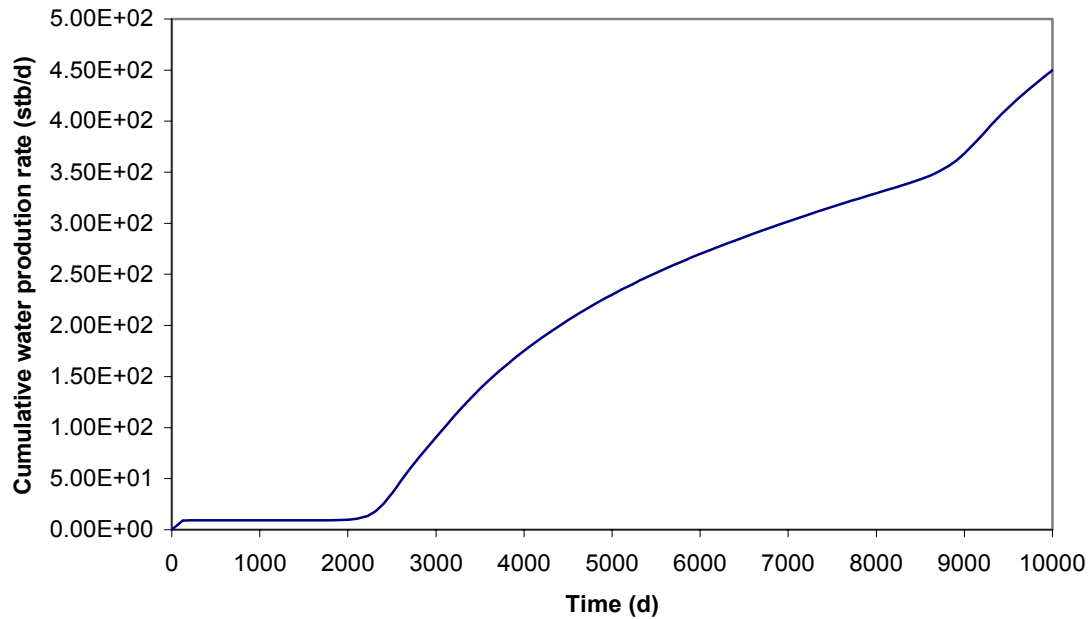
**Figure 1.6—Comparison of water production rate using corner point grid implementation and Cartesian grid.**



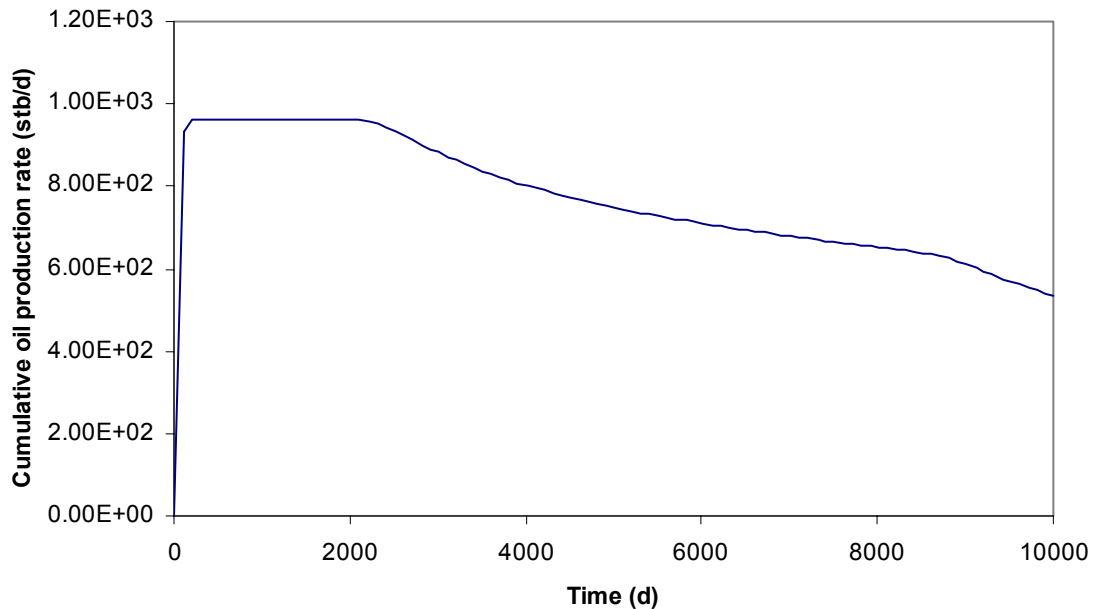
**Figure 1.7—Comparison of oil production rate using corner point grid implementation and Cartesian grid.**



**Figure 1.8—Schematic of the grid used in Case 2 simulation.**



**Figure 1.9—Cumulative water production rate using corner point grid implementation using the grid shown in Fig. 1.8.**



**Figure 1.10—Cumulative oil production using corner point grid implementation using the grid shown in Fig. 1.8.**

### **Automatic Time Step Control**

A new automatic time step control was added which is more stable and more efficient compared to the one already available in the code based on the number of Newtonian iterations. The new implementation is based on the maximum change in pressure, phase saturation, and molar concentration of each species. The formulation is described as below

$$\Delta t^{n+1} = \Delta t^n \left[ \frac{(1 + \omega) \eta_i}{\delta_i + \omega \eta_i} \right]_{\min \text{ over all gridblocks}} \quad (1.86)$$

where  $\omega$  is the tuning factor and is between a value of zero and one,  $\eta_i$  is the maximum relative change for each variable as an input parameter, and  $\delta_i$  is the computed relative change of the variable in gridblock  $i$  during the time step  $\Delta t$ . The value in the bracket in Eq. 1.86 will be different depending on the variables pressure, saturation, and species molar concentration used and the minimum will be used in the time step selection. Additional input parameters added for the time step control option 2 (TIME\_CONTROL = 2) are listed as below.

TUNE	tuning factor with a value between 0 and 1
DCMAX	maximum relative change of molar concentration
DPMAX	maximum relative change of pressure
DSPMAX	maximum relative change of saturation
DELTIM	initial time step in days
DTIMMAX	maximum time step, days
DTIMMIN	minimum time step, days

Several simulations were performed to test and verify the time step routine and its implementation in GPAS.

### Linear and Nonlinear Iterative Solvers

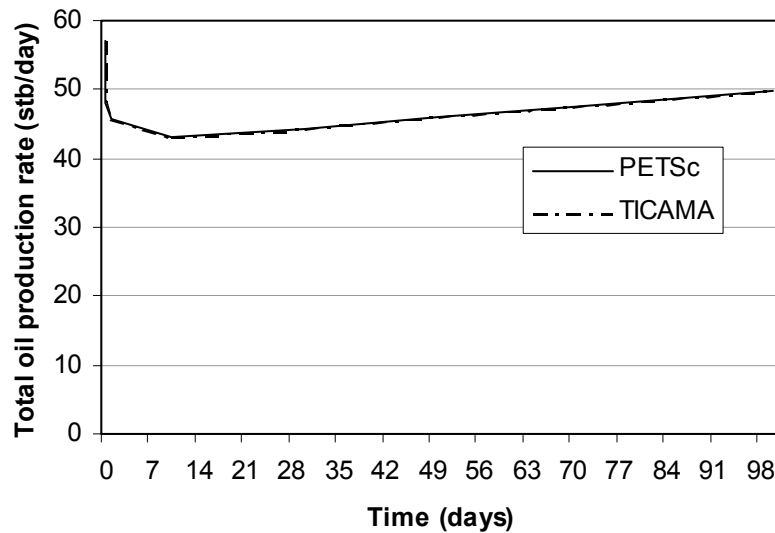
The solver package included in GPAS as a default solver is PETSC developed at Argonne National Laboratory (Balay et al., 1997). The linear solver from PETSC has been tested with the EOS compositional simulations for gas flooding and the results indicated that these solvers are very efficient and robust. Others solvers that were implemented are

TICAMA	generalized minimal residual method, GMRES
LSOR	line successive over relaxation method
PCG	Preconditioned conjugate gradient method

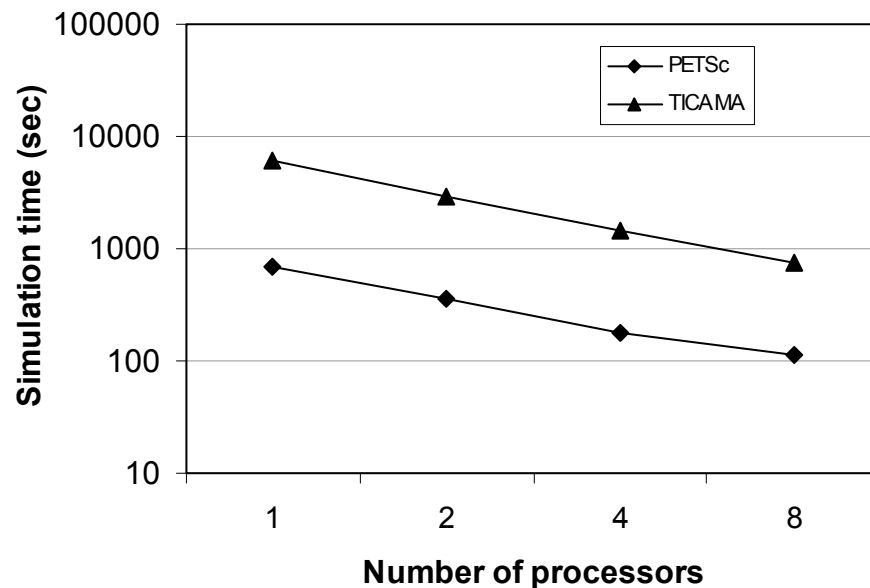
Several input files were set up and tested with each solver. The results indicated that the TICAMA was the fastest and most robust solver among those tested as given above.



Most of the data sets had convergence problem with both LSOR and PCG. The oil production rate for a 3-D, three-component compositional gas problem is compared for simulations using PETSc and TICAMA. Figure 1.11 shows identical results for the two solvers. The total number of gridblocks was 28,672 (16x224x8). The simulation was run on a single processor. The simulation time is compared in Fig. 1.12 which shows about 10 times faster simulation time using the PETSc.



**Figure 1.11—Oil production rate comparison between PETSc and TICAMA.**



**Figure 1.12—Comparison of simulation time for PETSc and TICAMA.**

### Enhancements in Well Model

To increase the flexibility of well operations in GPAS, we have added new options for injection and production well constraints. These include

- Constant water volumetric rate injection well
- Constant gas volumetric rate injection well
- Constant oil volumetric rate production well

The formulation and the procedure to implement the new well constraints are outlined as follows:

A. Define molar rate of each components in well grid block  $((q_i)_z$ , for species  $i = 1, n_c+1$ , where  $n_c$  is number of hydrocarbon component and  $n_c+1$  means water component). The molar rate  $(q_i)_z$  are defined as

1. Constant flowing bottomhole pressure injection well

$$(q_i)_z = [1 - (f_l)_{inj}](z_i)_{inj}(q_t)_z \text{ for } i = 1, n_c \quad (1.87)$$

$$(q_{n_c+1})_z = (f_l)_{inj}(q_t)_z \text{ for } z = z_t, z_b \quad (1.88)$$

Where  $z_t$  is the top perforation layer and  $z_b$  is the bottom perforation layer.

2. Constant volumetric rate injection well

The gas injection rate  $(Q_g)$  in Mscf/D, the water injection rate  $(Q_w)$  in STB/D, and the hydrocarbon composition of the injected fluid  $(z_i)_{inj}$ , are specified. The volumetric rates are converted to molar flow rates using the following equations:

$$q_i = 2.6357Q_g(z_i)_{inj} \text{ for } i = 1, n_c \quad (1.89)$$

and

$$q_{n_c+1} = 19.4658Q_w \quad (1.90)$$

After computing the molar flow rates, the allocation scheme used for constant molar rate injection wells is then employed to distribute the molar injection rates into each layer based on the productivity index (PI).

$$(q_i)_z = \frac{q_i(PI_t)_z}{\sum_{m=z_t}^{z_b} (PI_t)_m} \text{ for } i = 1, n_c, n_c + 1 \text{ and } z = z_t, z_b \quad (1.91)$$

3. Constant flowing bottomhole pressure production well

$$(q_i)_z = \sum_{j=2}^{np} (\xi_j x_{ij} PI_j)_z (P_{wf} - P_j)_z \text{ for } i = 1, n_c \text{ and } z = z_t \text{ to } z_b \quad (1.92)$$

and

$$(q_{nc+1})_z = (\xi_1 PI_1)_z (P_{wf} - P_1)_z \text{ for } z = z_t \text{ to } z_b \quad (1.93)$$

#### 4. Constant oil volumetric rate production well

The oil rate production in STB/D,  $Q_o$  is specified in this option. A flash calculation is done at separator conditions to determine the molar fraction of oil phase,  $(L_o)_{prod}$ , in the produced hydrocarbon fluid using the overall hydrocarbon composition computed by

$$(z_i)_{prod} = \frac{\sum_{m=z_t}^{z_b} \sum_{j=2}^{np} (\xi_j x_{ij} PI_j)_m}{\sum_{m=z_t}^{z_b} \sum_{j=2}^{np} (\xi_j PI_j)_m} \text{ for } i = 1, n_c \quad (1.94)$$

The total molar flow rate is then calculated using

$$q_t = \frac{5.6146 Q_o}{(L_o)_{prod} (v_o)_{prod}} \frac{\sum_{m=z_t}^{z_b} \sum_{j=1}^{np} (\xi_j PI_j)_m}{\sum_{m=z_t}^{z_b} \sum_{j=2}^{np} (\xi_j PI_j)_m} \quad (1.95)$$

where  $(v_o)_{prod}$  is the molar volume of the oil phase at the separator conditions. The same allocation scheme for constant molar rate production wells is used to compute the layer component rates.

When the total molar production rate ( $q_t$ ) is specified, the total production rate for each layer is calculated using:

$$(q_t)_z = \frac{q_t \sum_{j=1}^{np} (\xi_j PI_j)_z}{\sum_{m=z_t}^{z_b} \sum_{j=1}^{np} (\xi_j PI_j)_m} \text{ for } z = z_t, z_b \quad (1.96)$$

The layer component rates are found by

$$(q_i)_z = \frac{(q_t)_z \sum_{j=2}^{np} (\xi_j x_{ij} PI_j)_z}{\sum_{j=1}^{np} (\xi_j PI_j)_z} \text{ for } i = 1, n_c \text{ and } z = z_t \text{ to } z_b \quad (1.97)$$

and

$$(q_{nc+1})_z = \frac{(q_t)_z (\xi_1 PI_1)_z}{\sum_{j=1}^{np} (\xi_j PI_j)_z} \text{ for } z = z_t \text{ to } z_b \quad (1.98)$$

The  $(q_i)_z$  are added to residuals of species conservation equations for each components.

$$ARM(i) = ARM(i) + (q_i)_z \text{ for } i = 1, n_c \quad (1.99)$$

$$ARW = ARW + (q_{nc+1})_z \quad (1.100)$$

- B. Derive the derivatives of  $(q_i)_z$  w.r.t. independent variables  $(\ln K_1, \dots, \ln K_{nc}, N_1, \dots, N_{nc}, P, N_w)$ .
- C. Modify the elements of the Jacobian matrix. The derivatives in Step B are subtracted from the existing Jacobian for the species mass conservation equations.

For hydrocarbon components:

$$AMK(i, j) = AMK(i, j) - \frac{\partial (q_i)_z}{\partial \ln K_j} \text{ for } i, j = 1, \dots, n_c \quad (1.101)$$

$$AMNP(i, j) = AMNP(i, j) - \frac{\partial (q_i)_z}{\partial N_j} \text{ for } i, j = 1, \dots, n_c \quad (1.102)$$

$$AMNP(i, j+1) = AMNP(i, j+1) - \frac{\partial (q_i)_z}{\partial P} \quad (1.103)$$

$$AMW(i) = AMW(i) - \frac{\partial (q_i)_z}{\partial N_w} \quad (1.104)$$

For water components:

$$AWKN(j) = AWKN(j) - \frac{\partial(q_{nc+1})_z}{\partial \ln K_j} \text{ for } j = 1, n_c \quad (1.105)$$

$$AWKN(j) = AWKN(j) - \frac{\partial(q_{nc+1})_z}{\partial N_j} \text{ for } j = n_c + 1, 2n_c \quad (1.106)$$

$$AWW(1) = AWW(1) - \frac{\partial(q_{nc+1})_z}{\partial N_w} \quad (1.107)$$

$$AWW(2) = AWW(2) - \frac{\partial(q_{nc+1})_z}{\partial P} \quad (1.108)$$

Where AMW and AWKN are new arrays created.

D. Add codes to change well type. When residuals are converged, if bottomhole pressure in a constant volumetric rate injection well (constant oil volumetric rate production well) is larger (smaller) than given pressure limit, the well type is changed to constant flowing bottomhole pressure injection well (constant flowing bottomhole pressure production well) and Newtonian iteration is repeated.

E. Create new keyword input variables for the well models.

1. KWELL(IW)

- a. Constant flowing bottomhole pressure injection well
- b. Constant volumetric rate injection well (newly added)
- c. Constant flowing bottomhole pressure production well
- d. Constant oil volumetric rate production well (newly added)
- e. IW: well index

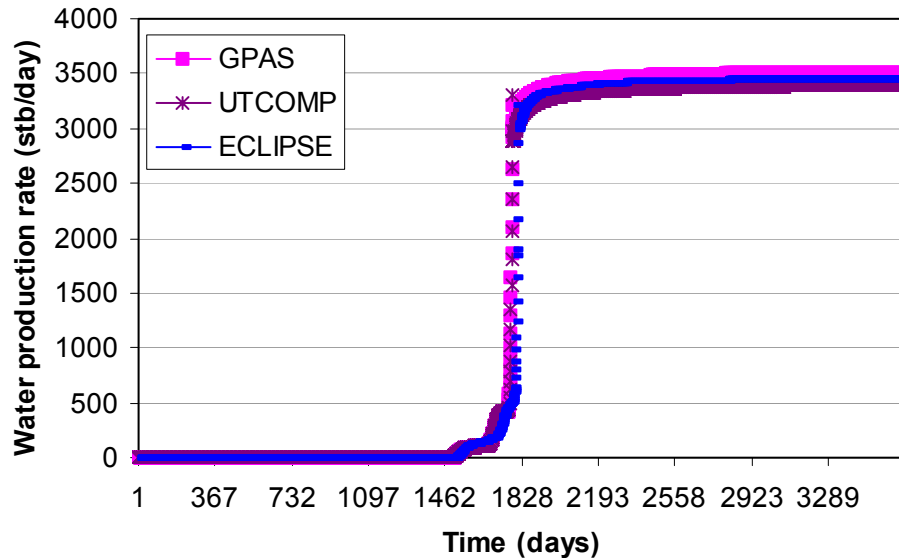
2. PRLIMIT(IW): Upper limit for bottomhole pressure in psi. If the pressure exceeds this input value, the well will operate under rate constrained to switch the well to a rate.

3. WELLPQ(IW)

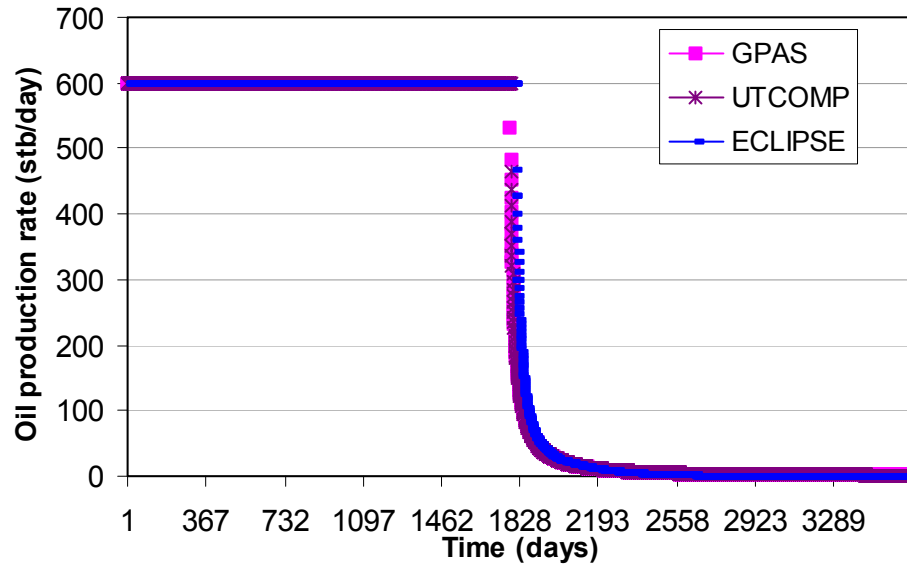
If KWELL(IW)

- a. Read bottomhole pressure (psi)
- b. Read injected volumetric rate (Mscf/day: gas, stb/day: water)
- c. Read bottomhole pressure (psi)
- d. Read produced volumetric rate (stb/day)

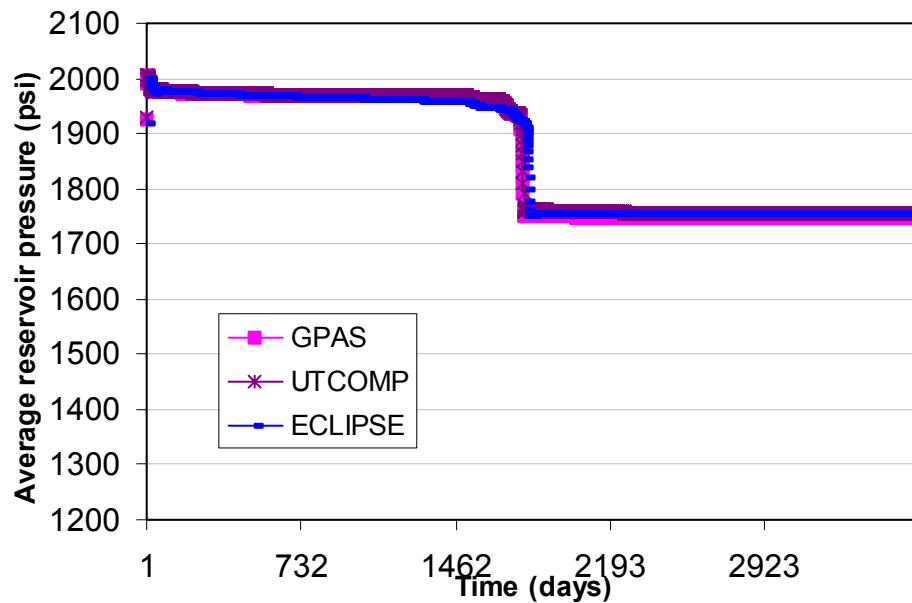
We have made several runs using the new well constraints and favorably compared with the results of the commercial simulator from Geoquest, Eclipse 300 and The University of Texas compositional gas simulator, UTCOMP. First, we ran water injection cases in a reservoir initially saturated with oil and water with a uniform water saturation of 0.1. The permeability was uniform at 500 md and porosity of 0.25. The simulation domain was 560 ft x 560 ft x 100 ft and was discretized in 7 x 7 x 3 gridblocks. The first case was with constant flowing bottomhole injection pressure of 2000 psi and constant oil production rate of 600 bbls/day. Figures 1.13 through 1.15 compare water production rate, oil production rate, and average reservoir pressure for the three simulators. The results are almost identical among the three simulators. The next waterflood simulation was done with constant water injection rate of 1000 stb/day and constant production pressure of 1500 psi. Figures 1.16 through 1.18 compare the water and oil production rates and average reservoir pressure obtained from GPAS, UTCOMP, and Eclipse 300. The results are very similar.



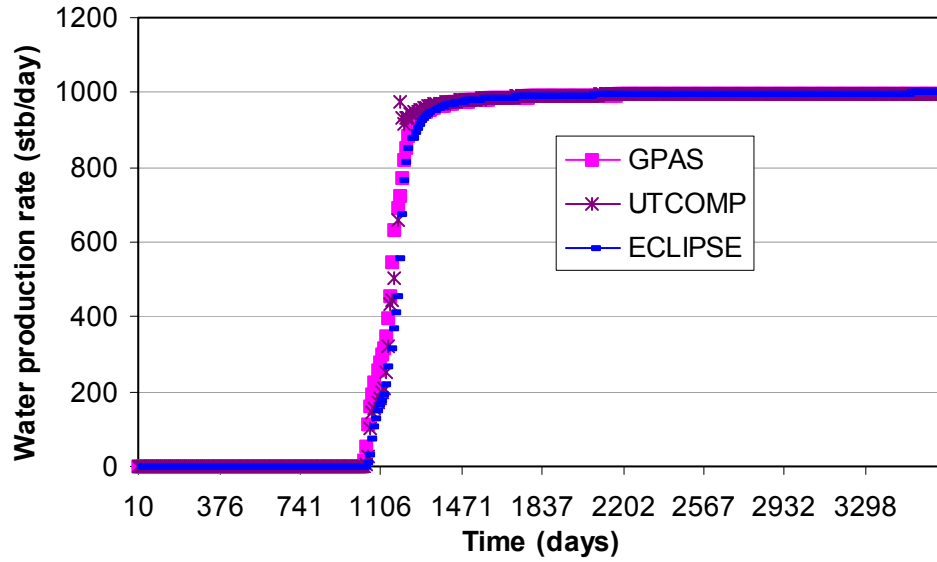
**Figure 1.13—Comparison of water production rate for water injection problem with constant pressure injector and constant rate producer.**



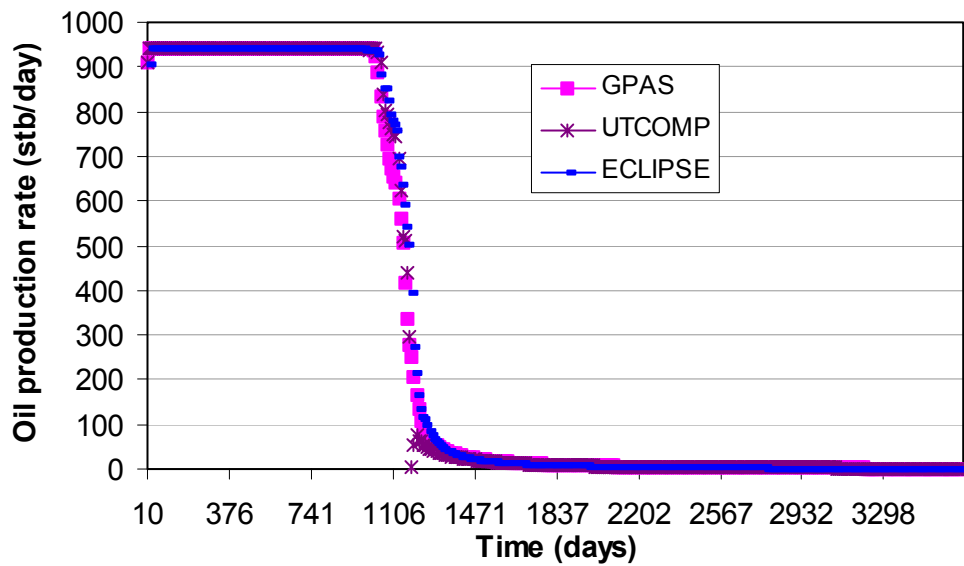
**Figure 1.14—Comparison of oil production rate for water injection problem with constant pressure injector and constant rate producer.**



**Figure 1.15—Comparison of reservoir pressure for water injection problem with constant pressure injector and constant rate producer.**

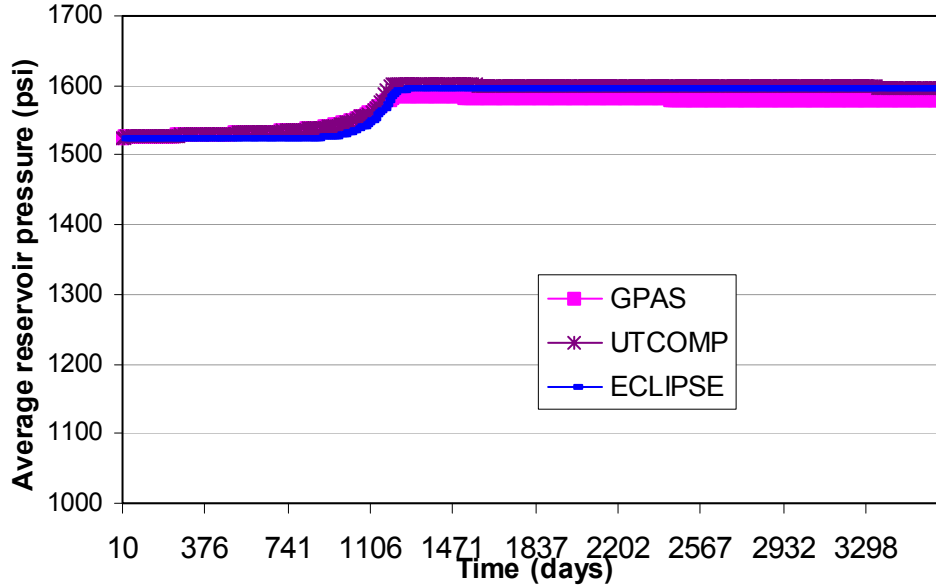


**Figure 1.16—Comparison of water production rate for water injection problem with constant injection rate and constant pressure producer.**



**Figure 1.17—Comparison of oil production rate for water injection problem with constant injection rate and constant pressure producer.**



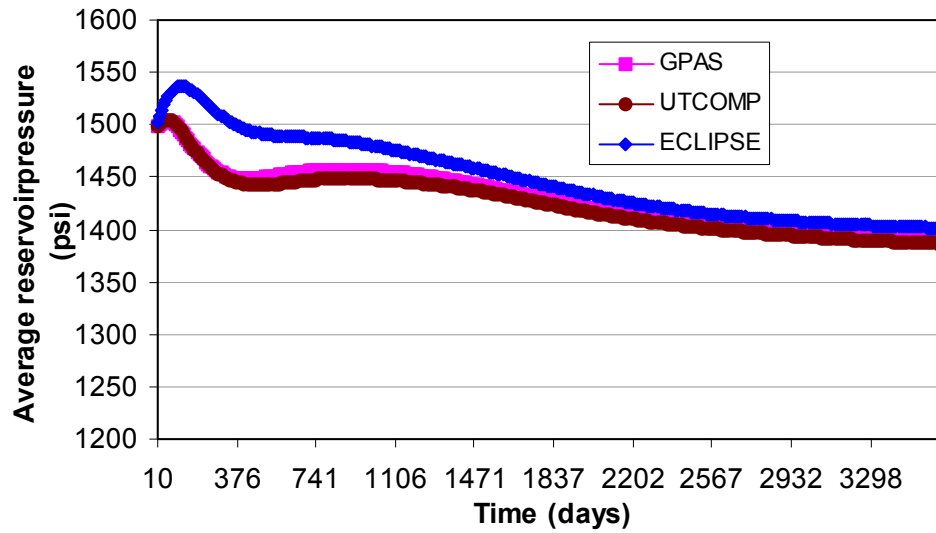


**Figure 1.18—Comparison of average reservoir pressure for water injection problem with constant injection rate and constant pressure producer.**

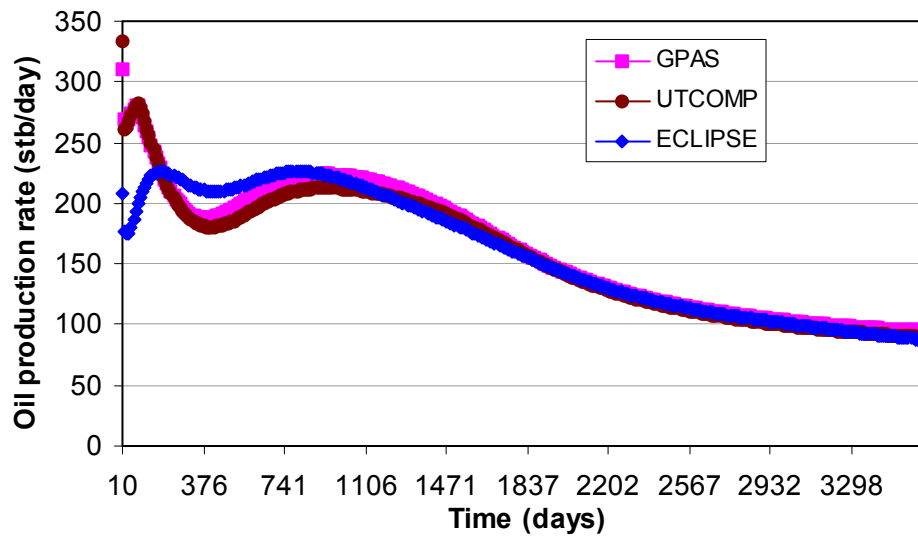
Other validation test was for a constant rate gas injection at 1000 Mscf/d with constant pressure production well at 1300 psi. The simulation domain was 560 ft x 560 ft x 100 ft discretized in 7 x 7 x 3 gridblocks. The permeability and porosity were uniform at 10 md and 0.25 respectively. The initial and injected gas compositions are given in Table 1.1. Average reservoir pressure, gas and oil production rates are compared for the three simulators in Figs. 1.19 through 1.21. The results of UTCOMP and GPAS are very similar. However, there is a slight difference between Eclipse and GPAS results at early time in average reservoir pressure and oil production rates.

**Table 1.1—Initial and Injected Gas Composition Used in 3-D Simulations**

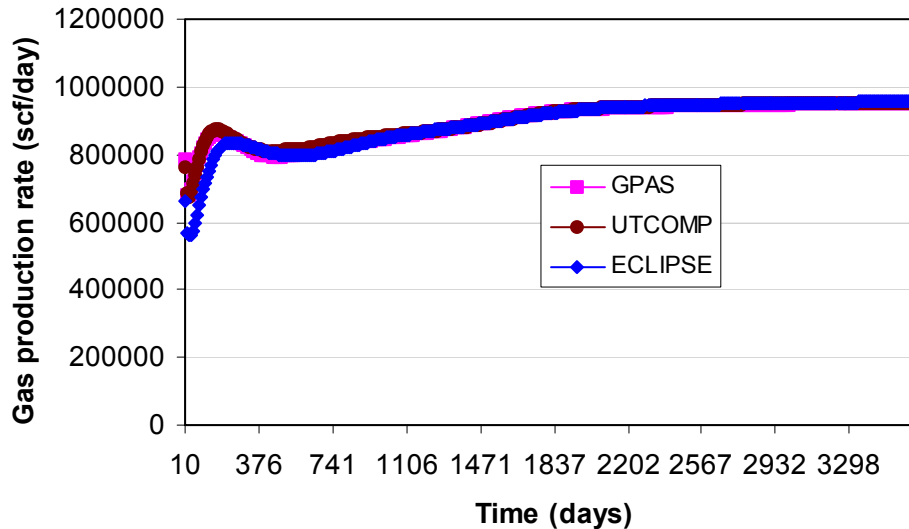
Components	Initial Composition (mole fraction)	Injected Composition (mole fraction)
C1	0.50	0.77
C3	0.03	0.20
C6	0.07	0.01
C10	0.20	0.01
C15	0.15	0.005
C20	0.05	0.005



**Figure 1.19—Comparison of average reservoir pressure for gas injection problem with constant injection rate and constant pressure producer.**



**Figure 1.20—Comparison of oil production rate for gas injection problem with constant injection rate and constant pressure producer.**



**Figure 1.21—Comparison of gas production rate for gas injection problem with constant injection rate and constant pressure producer.**

## Generation of Derivatives using Automatic Differentiation

During last half a decade, an alternative way of generating derivatives and Jacobian without losing accuracy and speed is gaining grounds in scientific/engineering community. This method is called Algorithmic Differentiation (AD), and we made an attempt in using its implementation for FORTRAN 77 codes called ADIFOR, a tool developed over the years to automate the process of generating derivatives and Jacobian construction.

ADIFOR is a tool for the automatic differentiation of Fortran 77 programs and co-developed by ANL and Rice University. Given a Fortran 77 source code and a configuration file including user's specification of dependent and independent variables, ADIFOR will generate an augmented derivative code that computes the partial derivatives of all of the specified dependent variables with respect to all of the specified independent variables.

ADIFOR performs the derivatives in the following manner. First, ADIFOR traces the code and generates a trace file that clearly maps the relationships between variables. After the trace file is generated for a given source code, it applies the chain-rule to every single statement in the code until the full derivative is generated.

We made a significant effort to incorporate ADIFOR in GPAS but due to the incompatibility of many non-standard FORTRAN statements in GPAS with ADIFOR, we decided to pursue this route in future once there are software utilities available to automatically replace the non-standard FORTRAN statements.

As the results of difficulties implementing ADIFOR in GPAS, GPAS currently employs hand-derived derivative code to calculate derivatives of residuals with respect to the primary unknowns, Jacobian entries. Hand-derived derivatives always provide faster

convergence and more accurate results compared to numerical alternatives such as divided differences. On the other hand, generating derivatives by hand is tedious.

## ***Task 2: Formulation and Implementation of Chemical Module***

Here we report on the formulation, implementation, and validation of aqueous components added to GPAS. Henceforth, the aqueous species refers to a species present in trace quantities in the aqueous phase and this includes a tracer, electrolytes, polymer or surfactant. As discussed earlier, we have implemented the chemical species using explicit and fully implicit methods.

### **Explicit Chemical Module**

In the implementation of the chemical module, the key assumption is that chemical species such as tracers, polymer, surfactant and electrolytes occupy negligible volume and do not affect the EOS model governing equations, which is reasonable for typical chemical enhanced oil recovery processes and other applications such as aqueous tracers. The chemical species mass balance is performed at the end of every time-step after the convergence of the EOS model solution. The phase fluxes and saturations derived from the EOS model at the end of each time-step are used in the explicit chemical species mass balance calculations. The computation of the mass balance and subsequent phase behavior is thus separated from the EOS model computation and coupled only through the modification of the physical properties such as relative permeability and aqueous phase viscosity due to surfactant phase behavior. This splitting of the mass balance calculations and coupling scheme make the implementation of aqueous component model in an existing compositional simulator formulation easy and efficient. More details can be found in Nalla, 2002 and John, 2003.

### **Mass Balance Equations**

The conservation equations for chemical species  $i$  with convective transport and adsorption in a permeable medium can be written as follows:

$$\frac{\partial}{\partial t}(\phi \tilde{C}_i \rho_i) + \bar{V} \cdot \left( \sum_{j=1}^3 C_{ij} \rho_i \bar{u}_j \right) = R_i \quad (2.1)$$

We perform the mass balance in this simplified form because it is solved explicitly using the phase properties and fluxes calculated by the EOS model for the same timestep. The direct calculation also helps because all the chemical phase property calculations are based on either mass or volume fractions and chemical species injection is also typically specified in mass or volume fractions.

We assume that the chemical species are transported in the aqueous phase only (except for partitioning tracers) and neglect physical dispersion and chemical reactions. Note that in presence of surfactant the aqueous phase becomes the microemulsion phase and is labeled as  $j = 3$  in the subsequent chemical phase behavior calculations.

For a tracer  $i$  that partitions between an aqueous phase and an oleic phase, we have

$$n_i = V_b \phi (\omega_{i1} \rho_1 S_1 + \omega_{i2} \rho_2 S_2) \quad (2.2)$$

where  $n_i$  is the total mass of tracer  $i$ .  $\omega_{i1}$  and  $\omega_{i2}$  are the tracer mass fractions in aqueous and oleic phases respectively. The partition coefficient is defined as

$$K_{T_i} = \frac{\omega_{i2}}{\omega_{i1}} \quad (2.3)$$

Substituting this in Eq. 2.2 and solving for  $\omega_{i1}$  and  $\omega_{i2}$  gives

$$n_i = V_b \phi (\omega_{i1} \rho_1 S_1 + K_{T_i} \omega_{i1} \rho_2 S_2) \quad (2.4)$$

$$\omega_{i1} = \frac{n_i}{V_b \phi (\rho_1 S_1 + K_{T_i} \rho_2 S_2)} \quad (2.5)$$

$$\omega_{i2} = K_{T_i} \omega_{i1} \quad (2.6)$$

When the input tracer concentrations are defined on a mass per volume basis, the above equations are modified by

$$C_{i1} = \omega_{i1} \rho_1 \quad (2.7)$$

$$C_{i2} = \omega_{i2} \rho_2 \quad (2.8)$$

$$K_{T_i} = \frac{C_{i2}}{C_{i1}} \quad (2.9)$$

The  $K_T$  values in Eqs. 2.3 and 2.9 differ numerically by a factor equal to the oil-water density ratio. The tracer partition coefficients are assumed to be constant.

### Adsorption

A Langmuir-type isotherm is used to model polymer ( $i=4$ ) and surfactant ( $i=3$ ) adsorption as a function of concentration and salinity (Hirasaki and Pope, 1974)

$$\hat{C}_i = \min \left( \tilde{C}_i, \frac{a_i (\tilde{C}_i - \hat{C}_i)}{1 + b_i (\tilde{C}_i - \hat{C}_i)} \right) \quad (2.10)$$

where  $a_i = (a_{i1} + a_{i2}C_{SE})$  and  $\tilde{C}_i$  is the aqueous concentration of species  $i$  normalized by the water concentration  $C_1$ . The adsorption model parameters  $a_{i1}$ ,  $a_{i2}$ , and  $b_i$  are found by matching surfactant or polymer adsorption data.  $C_{SE}$  is the effective salinity as the salt concentration in the aqueous phase. Equation 2.10 is rearranged to quadratic form and solved to get the amount of species adsorbed. The implementation assumes adsorption to be irreversible with respect to both salinity and concentration. The minimum is taken to ensure that the adsorbed concentration is not greater than the total concentration.

### Permeability Reduction and Inaccessible Pore Volume

The permeability reduction due to polymer is modeled by the permeability reduction factor,  $R_k$ , defined as the ratio of the effective permeability of the aqueous phase to the effective permeability of the aqueous phase containing polymer:

$$R_k = 1.0 + \frac{(R_{k,max} - 1.0)b_{rk}C_{ij}}{1.0 + b_{rk}C_{ij}} \quad (2.11)$$

where  $R_{k,max}$  is calculated based on the rock properties and salinity.  $j$  refers to the polymer rich aqueous phase and  $b_{rk}$  is the input parameter. The effective porosity for the polymer is modeled as

$$\phi_{polymer} = \phi_{actual}E_{phi} \quad (2.12)$$

where  $E_{phi}$  is the factor needed to account for the fact that polymers typically flow in the pores at a higher velocity than small molecules, the so called inaccessible pore volume effect.

### Oil/Brine/Surfactant Phase Behavior

The surfactant/oil/water phase behavior is based on Winsor, 1954; Reed and Healy, 1977; Nelson and Pope, 1978; Prouvost *et al.*, 1985; and Camilleri *et al.*, 1987. The volumetric concentrations of the three components are used as the coordinates on a ternary diagram. Although salinity, presence of alcohol and divalent cations influence the phase behavior significantly, we model the phase behavior as being affected by salinity alone and assume the absence of divalent cations and alcohols in this implementation.

If the surfactant concentration is below the Critical Micelle Concentration (CMC), the two phases are an aqueous phase containing all the surfactant, water, and electrolytes and a pure excess oil phase. At low salinity, an excess oil phase that is essentially pure oil and a microemulsion phase that contains water plus electrolytes, surfactant, and some solubilized oil exist. The tie lines (distribution curves) at low salinity have negative slope. This type of phase environment is called Winsor Type I, or Type II(-). At intermediate salinity, the mixture separates into three phases. These phases are excess oil and water phases and a microemulsion phase. This phase environment is called Winsor

Type III, or just Type III. At high salinity, an excess water phase and a microemulsion phase containing most of the surfactant and oil, and some solubilized water exist. This type of phase environment is called Winsor Type II, or alternatively Type II(+). Other variables besides electrolyte concentrations, e.g. alcohol type and concentration, the equivalent alkane carbon number of the oil or solvent and changes in temperature or pressure also cause a phase environment shift from one type of phase behavior to another type but these effects are not modeled here.

The surfactant/oil/water phase behavior is thus represented as a function of effective salinity once the binodal curve and tie lines are described. The oil concentrations used in the calculations are obtained from the EOS model at the end of each converged time step. Since they belong to the EOS model, they cannot be changed during the explicit chemical phase behavior calculations. This coupling forces the assumption of negligible changes in oil saturation due to solubilization. The surfactant phase behavior is therefore limited to Type II(-) behavior with microemulsion and oil as the two phases in equilibrium in this implementation with the following assumptions:

- isothermal conditions,
- corner plait point formulation where the excess oil phase is pure oil phase,
- binodal curve formulation based on Hand's rule (Hand, 1939),
- a symmetric binodal curve,
- no effect of divalent cations such as calcium or alcohol on phase behavior,
- the phase behavior is independent of the polymer concentration,
- no gas is present, and
- no effect of pressure on microemulsion phase behavior.

Hand's rule is based on the empirical observation that equilibrium phase concentration ratios are straight lines on a log-log scale. The binodal curve is described by

$$\frac{C_{3j}}{C_{2j}} = A \left( \frac{C_{3j}}{C_{1j}} \right)^{-1} \quad \text{for } j = 2, 3 \quad (2.13)$$

The parameter A is related to the height of binodal curve as follows:

$$A_s = \left( \frac{2C_{3\max,s}}{1 - C_{3\max,s}} \right)^2 \quad \text{for } s = 0, 1 \quad (2.14)$$

where  $s = 0, 1$  represents the zero and optimal salinities. Linear interpolation is used to determine the A parameter for intermediate salinities.

$$A = (A_0 - A_1) \left( 1 - \frac{C_{SE}}{C_{SEOP}} \right) + A_1 \quad \text{for } C_{SE} \leq C_{SEOP} \quad (2.15)$$

where the effective salinity  $C_{SE}$  is the salt concentration in the aqueous phase and the optimum salinity  $C_{SEOP}$  is calculated from the lower and upper effective salinities as

$$C_{SEOP} = \frac{C_{SEL} + C_{SEU}}{2} \quad (2.16)$$

Since all the component concentrations are in volume fractions, they must add up to one.

$$\sum_{k=1}^3 C_{kj} = 1 \text{ for } j = 2, 3 \quad (2.17)$$

For Type II(-),  $C_{SE} \leq C_{SEL}$  and the surfactant solubilization ratio in the aqueous phase is given by:

$$R_{31} = \frac{C_3^f}{C_1} \quad (2.18)$$

where  $C_3^f = \tilde{C}_3 - \hat{C}_3$  is the concentration of surfactant in the microemulsion after adjusting the total concentration for adsorbed surfactant. The solubilization ratio in oleic phase is

$$R_{32} = \frac{A}{R_{31}} \quad (2.19)$$

The microemulsion phase compositions are then computed as follows:

- For the water component

$$C_{13} = \frac{R_{32}}{R_{31} + R_{31}R_{32} + R_{32}} \quad (2.20)$$

- For the oil component

$$C_{23} = 1 - (1 + R_{31}) C_{13} \quad (2.21)$$

- For the surfactant component

$$C_{33} = 1 - C_{13} - C_{23} \quad (2.22)$$

The oil solubilization ratio is



$$R_{23} = \frac{C_{23}}{C_{33}} \quad (2.23)$$

### Interfacial Tension

Huh (Huh, 1979) proposed that the interfacial tension and solubilization ratio are related by the following function

$$\sigma_{23} = \frac{c}{R_{23}^2} \quad (2.24)$$

We use this equation with Hirasaki's correction factor (Hirasaki, 1981) so that it reduces to the water-oil interfacial tension as the surfactant concentration approaches zero.

For  $C_3 \geq \text{Critical Micelle Concentration (CMC)}$ :

$$\sigma_{23} = \sigma_{ow} e^{-a R_{23}} + \frac{c F_2}{R_{23}^2} \left( 1 - e^{-a R_{23}^3} \right) \quad (2.25)$$

$$F_2 = \frac{1 - e^{-\sqrt{\sum_{k=1}^3 (C_{k2} - C_{k3})^2}}}{1 - e^{-\sqrt{2}}} \quad (2.26)$$

where  $a$  and  $c$  are input parameters which match the model to the measured data.

When  $C_3 < \text{CMC}$ , there is no surfactant influence so

$$\sigma_{23} = \sigma_{ow} \quad (2.27)$$

where  $\sigma_{ow}$  is the oil-water interfacial tension.

### Viscosity

The aqueous phase viscosity at a fixed shear rate is a function of polymer concentration and salinity and this dependency is modeled using the Flory-Huggins type equation (Flory, 1953):

$$\mu_p = \mu_w \left( 1 + \left( A_{p1} C_{4\ell} + A_{p2} C_{4\ell}^2 + A_{p3} C_{4\ell}^3 \right) C_{SE}^{Sp} \right) \quad (2.28)$$

In this simplified implementation, we have neglected shear rate dependence of the polymer viscosity. The change in viscosity caused by surfactants is modeled by the following equation:

$$\mu_3 = C_{13}\mu_p e^{\alpha_1(C_{23}+C_{33})} + C_{23}\mu_o e^{\alpha_2(C_{13}+C_{33})} + C_{33}\alpha_3 e^{(\alpha_4 C_{13} + \alpha_5 C_{23})} \quad (2.29)$$

The pure water and oil viscosities are  $\mu_w$  and  $\mu_o$ .

### Relative Permeability

The relative permeability models implemented in the code are in the form of both lookup tables and Corey-type functions. Low interfacial tension leads to decreased capillary forces and mobilization of residual oil. This mobilization effect needs to be accurately captured in the relative permeability model. The trapping number is a dimensionless number that includes both the gravity/buoyancy and viscous forces on the trapped globule (Jin, 1995; Pope *et al.*, 2000).

$$N_{Tj} = \frac{\left| \bar{k} \bullet \left( \nabla \Phi_{j'} + g(\rho_{j'} - \rho_j) \nabla D \right) \right|}{\sigma_{jj'}} \quad (2.30)$$

The change in residual saturations is modeled as a function of the trapping number as shown below

$$S_{jr} = \min \left( S_j, S_{jr}^{\text{High}} + \frac{S_{jr}^{\text{Low}} - S_{jr}^{\text{High}}}{1 + T_j N_{Tj}^{\tau_j}} \right) \quad (2.31)$$

$S_{jr}^{\text{High}}$  and  $S_{jr}^{\text{Low}}$  are the residual saturations for phase  $j$  at high and low trapping numbers. This correlation was derived based on the experimental data for n-decane (Delshad, 1990).

The endpoints and exponents of the relative permeability curve also change as the residual saturations change at high trapping numbers (Chatzis and Morrow, 1981; Morrow *et al.*, 1985; Delshad *et al.*, 1986). These changes are computed by linear interpolation between the given input values at low and high trapping numbers:

$$k_{rj}^o = k_{rj}^{o\text{Low}} + \frac{S_{j'r}^{\text{Low}} - S_{j'r}}{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}} \left( k_{rj}^{o\text{High}} - k_{rj}^{o\text{Low}} \right) \quad (2.32)$$

$$n_j = n_j^{\text{Low}} + \frac{S_{j'r}^{\text{Low}} - S_{j'r}}{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}} \left( n_j^{\text{High}} - n_j^{\text{Low}} \right) \quad (2.33)$$

The trapping number model is implemented in a fully implicit form in the EOS compositional model. The main driver causing increased trapping number comes from the chemical model calculations (lowering of IFT). To build the Jacobian matrix for the fully implicit EOS model formulation we need the following partial derivatives for the relative permeability:

$$\frac{\partial k_{rj}}{\partial \ln K_i} \text{ for } i = 1, \dots, n_c, \frac{\partial k_{rj}}{\partial N_i} \text{ for } i = 1, \dots, n_c, n_w \text{ and } \frac{\partial k_{rj}}{\partial p}$$

A Corey-type function for relative permeability is used with the trapping number model. Without the trapping number, the relative permeability is a function of saturation only. The trapping number changes the relationship because it depends on the pressure, interfacial tension and densities. Also, the endpoint relative permeability and exponents are not constant and change with the trapping number. The derivatives in this case can be calculated applying the chain rule. The details are in Appendix A.

## Fully Implicit Chemical Module

The previous section was on the formulation, implementation, and validation of the surfactant, polymer, and tracer species in GPAS. The formulation was based on a hybrid approach and not fully implicit. In the hybrid approach, the material conservation equations for hydrocarbon are solved implicitly where the aqueous species material balances are solved explicitly using an updated phase fluxes, saturations, and densities. Although the hybrid approach has proved useful for some cases, the simulation results can be incorrect because it neglects oil amount dissolved in aqueous phase and surfactant volume in volume constraint equation. To overcome these limitations and take advantage of the larger time steps with the fully implicit formulation to reduce the simulation time, we have developed a fully implicit module of chemical flooding with the relevant physical properties. This section of the report is dedicated to the fully implicit implementation of the chemical module.

### Governing Equations

#### *Volume Constraint*

The sum of volumes of volume occupying components, that is, water, surfactant, and hydrocarbon components should add up to the pore volume. This volume constraint can be written as Eqs. 2.34. Volume of surfactant is neglected in hybrid approach, but it is considered in fully implicit method.

$$\frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}}{\xi_{surf}} + \frac{\sum_{i=1}^{n_c} N_i}{\xi_o} = 1.0 \quad (2.34)$$

where

$$\xi_i = \xi_i^0 (1 + c_i^0 (P - P^0)) \text{ for } i = \text{H}_2\text{O and surfactant}$$

### ***Material Balance Equations***

A hydrocarbon component can be dissolved in aqueous phase and there is no gas phase under Type II(-) surfactant phase environment. Therefore, material balance equation for the hydrocarbon component is as follows.

$$V_b \frac{\partial}{\partial t} (\phi N_i) - V_b \bar{\nabla} \cdot \left( \xi_2 \frac{\bar{k}k_{r2}}{\mu_2} x_{i2} (\nabla P - \gamma_2 \nabla D) + \xi_3 \frac{\bar{k}k_{r3}}{\mu_3} x_{i3} (\nabla P - \gamma_3 \nabla D) \right) = q_i \text{ for HC components} \quad (2.35)$$

However, aqueous components such as water, surfactant, polymer, salt, and tracer are assumed to be dissolved in only aqueous phase. In this case, the material balance equations for the aqueous component are

$$V_b \frac{\partial}{\partial t} (\phi N_i) - V_b \bar{\nabla} \cdot \left( \xi_3 \frac{\bar{k}k_{r3}}{\mu_3} x_{i3} (\nabla P - \gamma_3 \nabla D) \right) = q_i \text{ for aqueous components} \quad (2.36)$$

### ***Phase Behavior and Physical Properties***

The surfactant/brine/oil phase behavior formulation is the same as the one used in the explicit method. Surfactant related properties such as viscosity, adsorption, interfacial tension, and trapping number and its effect of relative permeabilities are the same as those used in the explicit chemical module. The polymer related properties such as viscosity, inaccessible pore volume, viscosity, permeability reduction and adsorption are also the same as those describe in the explicit chemical module.

### **Solution Approach**

Equations 2.34 to 2.36 are solved simultaneously using the fully implicit finite difference scheme. After the finite differencing of Equations 2.34 to 2.36 (the differentiated forms are described in Appendix B), we have  $(n_c + n_a + 2) \times n_b$  non-linear equations with the same number of unknowns  $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n_b})$ , where  $\bar{x}_i = (N_1, \dots, N_{n_c}, N_{n_c+1}, \dots, N_{n_c+n_a}, P, N_{\text{H}_2\text{O}})$  for  $i^{\text{th}}$  gridblock) at new time level when considering all the gridblocks. To solve the non-linear equations, Newton method is used. Residual vector for the method ( $\bar{R}$ ) consists of all the governing equations differentiated, and the unknown vector is  $\bar{x} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n_b})$ .  $\bar{x}$  is considered as a solution vector at new time level generally when it satisfies  $\bar{R}(\bar{x}) \approx \bar{0}$ . To search for the solution vector,  $\bar{x}$  is updated by

$\vec{x}^{\text{new}} = \vec{x}^{\text{old}} + \Delta \vec{x}$  where  $\Delta \vec{x}$  is obtained from  $\bar{J}(\vec{x}^{\text{old}}) \Delta \vec{x} = -\vec{R}(\vec{x}^{\text{old}})$  which can be rewritten as follows.

$$\begin{pmatrix} \bar{J}_{1,1} & \bar{J}_{1,2} & \dots & \bar{J}_{1,n_b} \\ \bar{J}_{2,1} & \bar{J}_{2,2} & \dots & \bar{J}_{2,n_b} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{J}_{n_b,1} & \bar{J}_{n_b,2} & \dots & \bar{J}_{n_b,n_b} \end{pmatrix} \begin{pmatrix} \Delta \bar{x}_1 \\ \Delta \bar{x}_2 \\ \vdots \\ \Delta \bar{x}_{n_b} \end{pmatrix} = - \begin{pmatrix} \bar{R}_1 \\ \bar{R}_2 \\ \vdots \\ \bar{R}_{n_b} \end{pmatrix} \quad (2.37)$$

where

$$\bar{J}_{i,j} = \begin{pmatrix} \frac{\partial R_{v_i}}{\partial N_{1,j}} & \dots & \frac{\partial R_{v_i}}{\partial N_{n_c,j}} & \frac{\partial R_{v_i}}{\partial N_{n_c+1,j}} & \dots & \frac{\partial R_{v_i}}{\partial N_{n_c+n_a,j}} & \frac{\partial R_{v_i}}{\partial P_j} & \frac{\partial R_{v_i}}{\partial N_{H_2O,j}} \\ \frac{\partial R_{m_{1,i}}}{\partial N_{1,j}} & \dots & \frac{\partial R_{m_{1,i}}}{\partial N_{n_c,j}} & \frac{\partial R_{m_{1,i}}}{\partial N_{n_c+1,j}} & \dots & \frac{\partial R_{m_{1,i}}}{\partial N_{n_c+n_a,j}} & \frac{\partial R_{m_{1,i}}}{\partial P_j} & \frac{\partial R_{m_{1,i}}}{\partial N_{H_2O,j}} \\ \vdots & & \vdots & & & \vdots & & \vdots \\ \frac{\partial R_{m_{n_c,i}}}{\partial N_{1,j}} & \dots & \frac{\partial R_{m_{n_c,i}}}{\partial N_{n_c,j}} & \frac{\partial R_{m_{n_c,i}}}{\partial N_{n_c+1,j}} & \dots & \frac{\partial R_{m_{n_c,i}}}{\partial N_{n_c+n_a,j}} & \frac{\partial R_{m_{n_c,i}}}{\partial P_j} & \frac{\partial R_{m_{n_c,i}}}{\partial N_{H_2O,j}} \\ \frac{\partial R_{m_{n_c+1,i}}}{\partial N_{1,j}} & \dots & \frac{\partial R_{m_{n_c+1,i}}}{\partial N_{n_c,j}} & \frac{\partial R_{m_{n_c+1,i}}}{\partial N_{n_c+1,j}} & \dots & \frac{\partial R_{m_{n_c+1,i}}}{\partial N_{n_c+n_a,j}} & \frac{\partial R_{m_{n_c+1,i}}}{\partial P_j} & \frac{\partial R_{m_{n_c+1,i}}}{\partial N_{H_2O,j}} \\ \vdots & & \vdots & & & \vdots & & \vdots \\ \frac{\partial R_{m_{n_c+n_a,i}}}{\partial N_{1,j}} & \dots & \frac{\partial R_{m_{n_c+n_a,i}}}{\partial N_{n_c,j}} & \frac{\partial R_{m_{n_c+n_a,i}}}{\partial N_{n_c+1,j}} & \dots & \frac{\partial R_{m_{n_c+n_a,i}}}{\partial N_{n_c+n_a,j}} & \frac{\partial R_{m_{n_c+n_a,i}}}{\partial P_j} & \frac{\partial R_{m_{n_c+n_a,i}}}{\partial N_{H_2O,j}} \\ \frac{\partial R_{m_{H_2O,i}}}{\partial N_{1,j}} & \dots & \frac{\partial R_{m_{H_2O,i}}}{\partial N_{n_c,j}} & \frac{\partial R_{m_{H_2O,i}}}{\partial N_{n_c+1,j}} & \dots & \frac{\partial R_{m_{H_2O,i}}}{\partial N_{n_c+n_a,j}} & \frac{\partial R_{m_{H_2O,i}}}{\partial P_j} & \frac{\partial R_{m_{H_2O,i}}}{\partial N_{H_2O,j}} \end{pmatrix}$$

$\Delta \bar{x}_i = (\Delta N_1, \dots, \Delta N_{n_c}, \Delta N_{n_c+1}, \dots, \Delta N_{n_c+n_a}, \Delta P, \Delta N_{H_2O})$  for  $i^{\text{th}}$  gridblock

$\bar{R}_i = (R_v, R_{m_1}, \dots, R_{m_{n_c}}, R_{m_{n_c+1}}, \dots, R_{m_{n_c+n_a}}, R_{m_{H_2O}})$  for  $i^{\text{th}}$  gridblock

Figure 2.1 shows the flowchart to search a solution vector at new time level (n+1) using Newton method. Bold letters are the names of the Fortran subroutines used in GPAS. A new criterion to check the convergence has been implemented in addition to  $\bar{R}(\vec{x}) \approx \vec{0}$ . The convergence is based on the pressure ( $\Delta P$ ) and saturation changes ( $\Delta S_j$ ) being ‘sufficiently small’ for a given change of  $\vec{x}$  by Newton iteration. This criterion is preferred especially when large saturation changes may occur for very small changes in composition during Newton iteration.

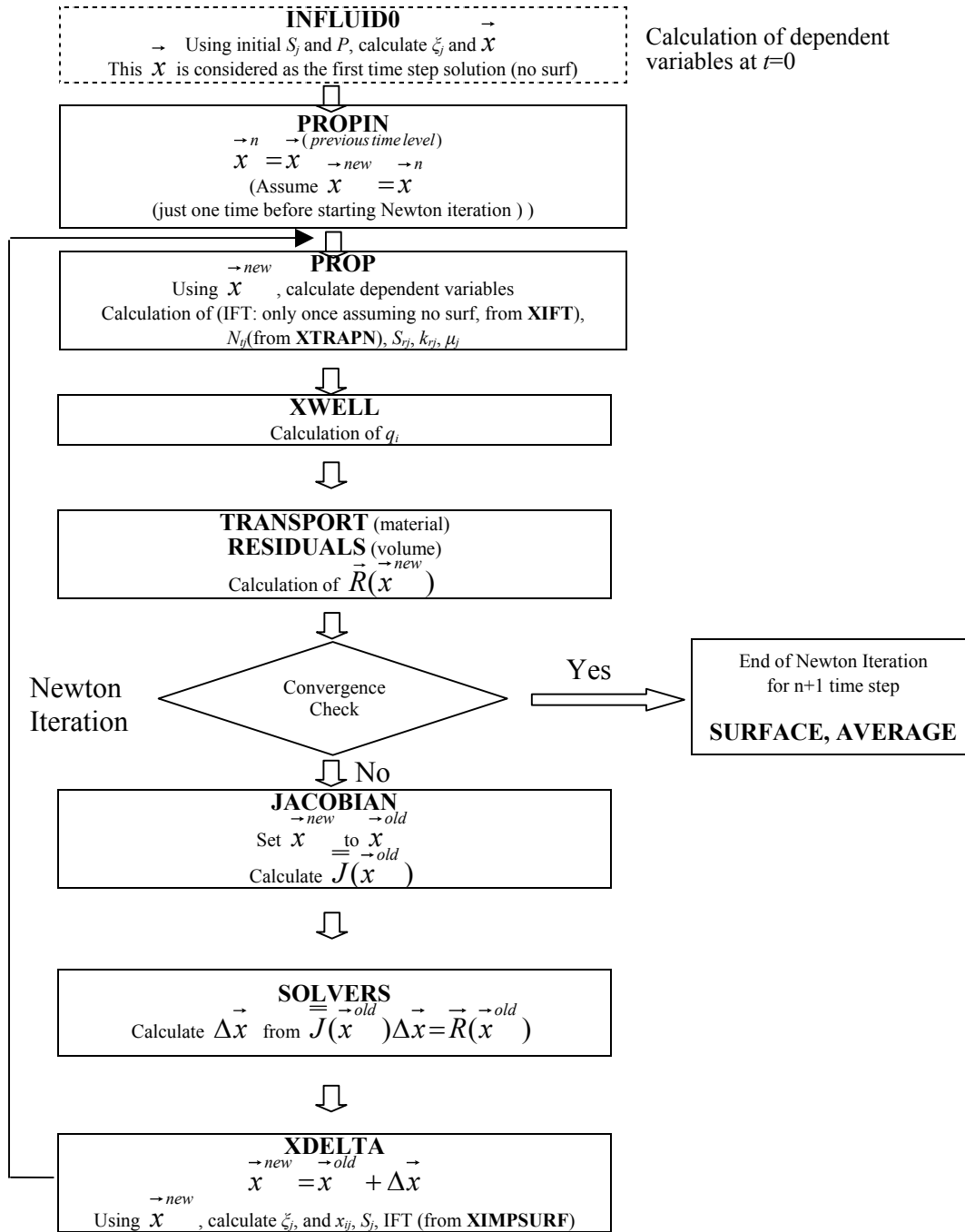


Figure 2.1—Flowchart of Newton method for fully implicit chemical flooding model.

### **Task 3: Validation and Application**

Test cases were run on GPAS and the results were compared with both analytical solutions and output from the miscible-gas flooding compositional simulator, UTCOMP (Chang, 1990), or the chemical flooding simulator, UTCHEM (Delshad *et al.*, 1996; Delshad *et al.*, 2002), to check the correctness of the code. The flash algorithm was tested with a batch flash calculation for a binary mixture. A Buckley-Leverett problem was run on GPAS and compared with the analytical solution and output from the chemical flooding simulator UTCHEM. A comparison of the results from GPAS with UTCOMP of an example simulation of carbon dioxide sequestration is also given. The standard SPE fifth comparative solution project (Killough and Kossack, 1987) was modified and simulations were carried out using GPAS and UTCOMP. The comparison of these results from GPAS with UTCOMP for the modified SPE fifth comparative solution project is discussed.

We performed a series of 2-D and 3-D surfactant/polymer flood simulations using both the explicit and implicit formulations and have made comparison with UTCHEM simulation results.

We have made numerous parallel simulations on up to 128 processors to study the scalability of GPAS. We have made for the first time, a 1000,000 gridblock surfactant/polymer flood simulation on 128 processors.

### **Batch Flash of Ethane-Propylene Binary Mixture**

The goal is to verify the correctness of the flash calculations. The mathematical formulation of the flash algorithm is discussed in Task 1. The test case considered here is a mixture of ethane and propylene. The simulation domain dimensions are 5000 ft in length, 500 ft in width and 20 ft in thickness. Since the simulation is a batch flash, there is no well section in the input. The initial composition is 0.5 mole fraction each of ethane and propylene. The details of the input file including the critical properties of the components are given in Table 3.1. The simulation is run for one day. A program for multi-component vapor-liquid equilibrium calculations using the Peng-Robinson cubic equation-of-state, VLMU (Sandler, 1999) was also used as a basis for comparison.

UTCOMP and GPAS use the same form of PR EOS while VLMU uses a slightly different form of PR EOS. For acentric factor values less than 0.49, all the three codes use the same equations. The only difference is that GPAS and UTCOMP use a different expression for the calculation of  $\kappa$  (Eqs 1.28 and 1.29) for the acentric factor values greater than 0.49 while VLMU uses the same expression for all values of acentric factors. Since in this test case, the acentric factor for both the components is less than 0.49, all the three codes essentially use the same PR EOS equations. The iteration tolerances in UTCOMP and GPAS are set at  $10^{-8}$ . The volume shift parameter functionality is available in UTCOMP but is not implemented in GPAS and VLMU. Hence the volume shift parameter is set to zero in UTCOMP.

**Table 3.1—Summary of Input Data for Ethane-Propylene Mixture Batch-Flash Calculation**

Dimensions (ft)						
Length				5000		
Width				500		
Thickness				20		
Porosity (fraction)				0.35		
Permeability (md)				0.001		
Reservoir Temperature (°F)				100		
Initial Reservoir Pressure (psi)				435.11		
Initial Composition (mole fraction)						
C <sub>2</sub> H <sub>6</sub>				0.5		
C <sub>3</sub> H <sub>6</sub>				0.5		
Simulation Time (day)				1.0		
Time Step Size (day)				0.1		
Component Critical Properties:						
<u>Component</u>	<u>T<sub>c</sub></u> <u>(°R)</u>	<u>P<sub>c</sub></u> <u>(psi)</u>	<u>V<sub>c</sub></u> <u>(ft<sup>3</sup>/lb-mole)</u>	<u><math>\frac{\omega_i}{dyne^{1/4}cm^{11/4}}</math></u> <u><math>\frac{gm-mole}{gm-mole}</math></u>	<u>MW<sub>i</sub></u> <u>(lbm/lb-mole)</u>	<u><math>\Psi_i</math></u>
C <sub>2</sub> H <sub>6</sub>	549.72	708.36	2.3703	0.098	30.070	111
C <sub>3</sub> H <sub>6</sub>	657.00	670.07	2.899	0.148	42.081	40.0
Binary Interaction Coefficients:						
	C <sub>2</sub> H <sub>6</sub>	C <sub>3</sub> H <sub>6</sub>				
C <sub>2</sub> H <sub>6</sub>	0	0				
C <sub>3</sub> H <sub>6</sub>	0	0				

Changing the binary interaction coefficient to 0, 0.01 and 0.05, three simulation runs were made. The oil and gas phase compositions from GPAS, UTCOMP and VLMU are compared in Table 3.2. The differences in the concentrations were based on VLMU solution. The oil and gas phase molar densities obtained from GPAS and UTCOMP are compared in Table 3.3. A reasonable agreement was obtained for the flash calculation results between GPAS and UTCOMP.



**Table 3.2—Comparison of Phase Compositions From GPAS With UTCOMP and VLMU in Ethane Propylene Batch - Flash Test Case**

Binary Interaction coefficient	Component Concentration (mole fraction)	VLMU	GPAS			UTCOMP		
		Final Composition	Solution	Absolute Difference	Difference Percentage	Solution	Absolute Difference	Difference Percentage
0	X C2H6	0.45	0.4496	0.0004	0.0888889	0.4468	0.0032	0.7111111
	Y C2H6	0.62	0.613	0.007	1.1290323	0.61005	0.00995	1.6048387
	X C3H6	0.55	0.5504	-0.0004	-0.072727	0.55332	-0.00332	-0.603636
	Y C3H6	0.38	0.387	-0.007	-1.842105	0.38995	-0.00995	-2.618421
0.01	X C2H6	0.4348	0.4399	-0.0051	-1.172953	0.43064	0.00416	0.9567617
	Y C2H6	0.6007	0.6001	0.0006	0.0998835	0.59667	0.00403	0.670884
	X C3H6	0.5651	0.5601	0.005	0.8847992	0.56936	-0.00426	-0.753849
	Y C3H6	0.3993	0.3999	-0.0006	-0.150263	0.40333	-0.00403	-1.009266
0.05	X C2H6	0.3701	0.3688	0.0013	0.3512564	0.36606	0.00404	1.0915969
	Y C2H6	0.55	0.549	0.001	0.1818182	0.54614	0.00386	0.7018182
	X C3H6	0.6299	0.6312	-0.0013	-0.206382	0.63398	-0.00408	-0.647722
	Y C3H6	0.45	0.451	-0.001	-0.222222	0.45386	-0.00386	-0.857778

X C2H6 concentration of Ethane in the oil phase  
 Y C2H6 concentration of Ethane in the gas phase  
 X C3H6 concentration of Propylene in the oil phase  
 Y C3H6 concentration of Propylene in the gas phase

**Table 3.3—Comparison of Phase Molar Densities From GPAS With UTCOMP and VLMU in Ethane Propylene Batch - Flash Test Case**

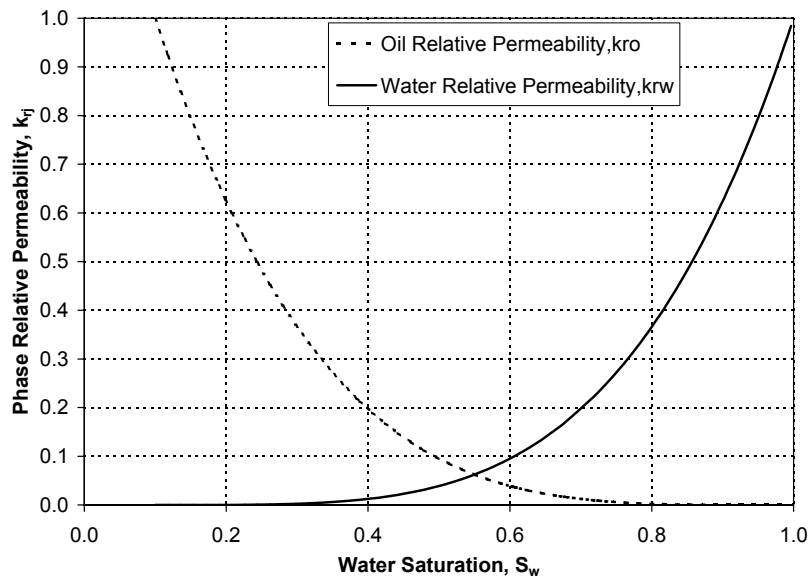
Binary Interaction coefficient	GPAS		UTCOMP		ABSOLUTE DIFFERENCE	
	Oil Phase Molar Density (lbmol/ft3)	Gas Phase Molar Density (lbmol/ft3)	Oil Phase Molar Density (lbmol/ft3)	Gas Phase Molar Density (lbmol/ft3)	Oil Phase Molar Density (lbmol/ft3)	Gas Phase Molar Density (lbmol/ft3)
0	6.9762E-01	1.1366E-01	6.9702E-01	1.1360E-01	5.9721E-04	5.8230E-05
0.01	0.6949	0.1138	0.6943	0.1137	5.7580E-04	5.6940E-05
0.05	0.6864	0.1141	0.6859	0.1140	5.0027E-04	6.3020E-05

## Buckley Leverett 1-D Water Flood

The Buckley-Leverett problem was chosen as a test problem because there is an analytical solution available for comparison. This problem often serves as a test for new numerical methods and as a building block of simulation models involving simultaneous flow of immiscible fluids in porous media.

The problem considered is a one-dimensional waterflood in a gas free, homogenous, isotropic reservoir. The homogenous permeability is 500 md. The simulation domain extends to 400 ft in the x direction, 1 ft in the y direction and 1 ft in the z direction. A 40x1x1 grid was used for this run. The initial water saturation in the reservoir is 0.1. The only injector is located at gridblock (1,1,1) and the only producer is located at gridblock (40,1,1). The initial reservoir pressure is 1500 psia. Water is injected at a constant bottom hole pressure of 2000 psia and the producer is maintained at a constant bottom hole pressure of 1500 psia. The relative permeability curves used for this problem are shown in Fig. 3.1. The two-phase relative permeability tables, used in GPAS, were

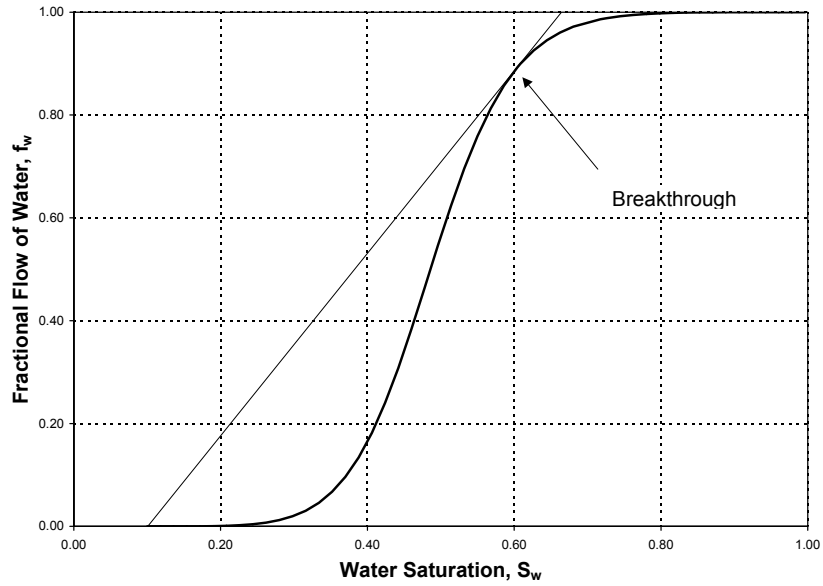
generated based on these functions. The end point mobility ratio is 3.15. The reservoir description, the hydrocarbon component properties and the time step details of the input file are given in Table 3.4. The fractional flow curves of oil and water are presented in Fig. 3.2. The water saturation profile results from GPAS, at different injected pore volumes, was compared with the output from UTCHEM and the analytical solution in Fig. 3.3. The one-point upstream weighting numerical method is employed for this simulation in GPAS, while both one point upstream and third order TVD (total variation diminishing) methods are employed in UTCHEM. A constant time step of 0.1 days is used in UTCHEM while in GPAS the maximum time step is 0.0005 days and the minimum time step is 0.00001 days. When the third order TVD numerical method (total variation diminishing) is used in UTCHEM, its results are closer to the analytical solution than when the one point upstream weighing method is used.



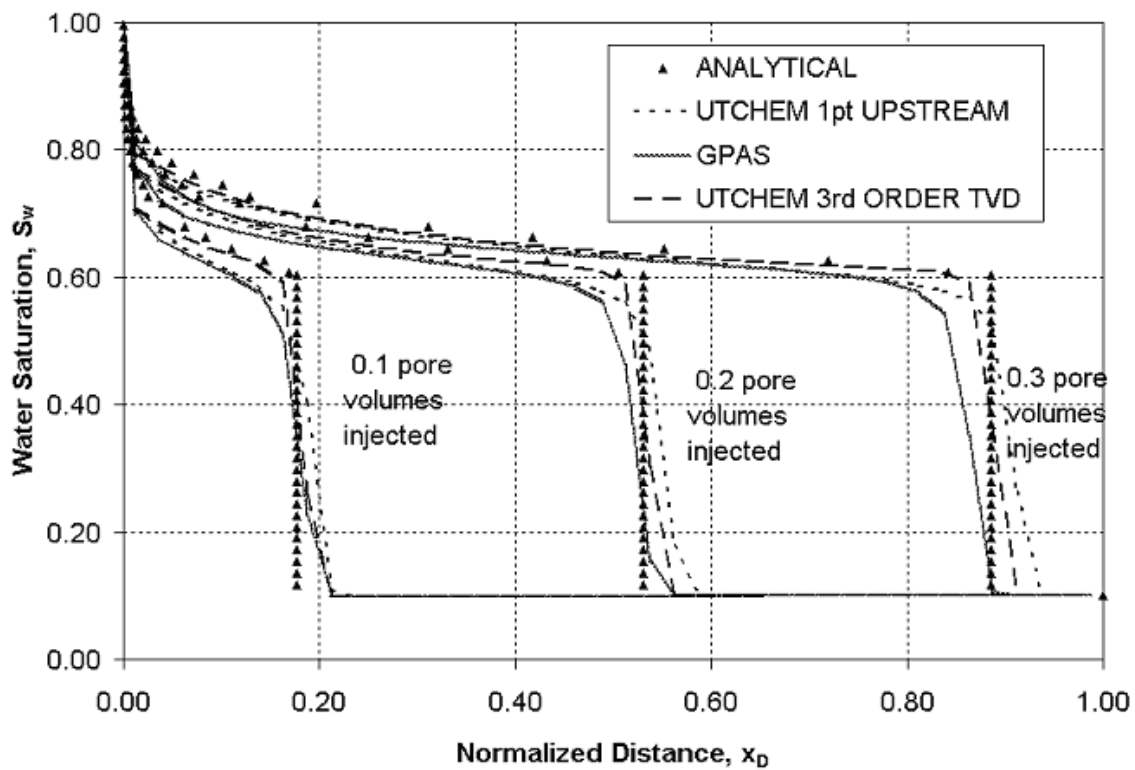
**Figure 3.1—Oil/water relative permeability curves used in the Buckley Leverett problem.**

**Table 3.4—Summary of Input Data for One-Dimensional Buckley Leverett Problem**

Dimensions (ft)						
Length				400		
Width				1		
Thickness				1		
Porosity (fraction)				0.25		
Permeability (md)				500		
Initial Water Saturation (fraction)				0.1		
Reservoir Temperature (°F)				130		
Initial Reservoir Pressure (psi)				1500		
Initial Composition (mole fraction)						
C <sub>10</sub> H <sub>22</sub>				1.0		
Water viscosity (cp)				1.0		
Oil viscosity (cp)				3.1527		
Water Compressibility (1/psi)				3.0x10 <sup>-5</sup>		
Rock Compressibility (1/psi)				0.0		
Simulation Time (day)				75.0		
Minimum Time Step Size in GPAS (day)				0.00001		
Maximum Time Step Size in GPAS (day)				0.0005		
Constant Time Step Size in UTCHEM (day)				0.00001		
Injector Well Location (I,J,K)				(1,1,1)		
Injection Bottomhole Pressure (psi)				2000		
Injection Fluid Composition						
C <sub>10</sub> H <sub>22</sub>				0.0		
H <sub>2</sub> O				1.0		
Producer Well Location (I,J,K)				(40,1,1)		
Production Bottomhole Pressure (psi)				1500		
Producer Well Location (I,J,K)				(40,1,1)		
Production Bottomhole Pressure (psi)				1500		
Component Critical Properties:						
<u>Component</u>	<u>T<sub>c</sub></u> (°R)	<u>P<sub>c</sub></u> (psi)	<u>V<sub>c</sub></u> (ft <sup>3</sup> /lb- mole)	<u>ω<sub>i</sub></u> (dyne <sup>1/4</sup> cm <sup>11/4</sup> / gm-mole)	<u>MW<sub>i</sub></u> (lbm/lb- mole)	<u>Ψ<sub>i</sub></u>
C <sub>10</sub> H <sub>22</sub>	1111.8	304.0	12.087	0.488	142.3	431
Relative Permeability Functions Used for Generating Tables:						
K <sub>r1</sub> <sup>0</sup> = 1	S = $\frac{S_w - S_{wr}}{1 - S_{or} - S_{wr}}$		K <sub>r2</sub> <sup>0</sup> = 1	K <sub>r2</sub> = K <sub>r2</sub> <sup>0</sup> (1 - S) <sup>n</sup>		
m = 4			n = 4			
S <sub>wr</sub> = 0.1	K <sub>r1</sub> = K <sub>r1</sub> <sup>0</sup> (S) <sup>m</sup>		S <sub>or</sub> = 0			



**Figure 3.2—Fractional flow curve of water in the Buckley Leverett problem.**



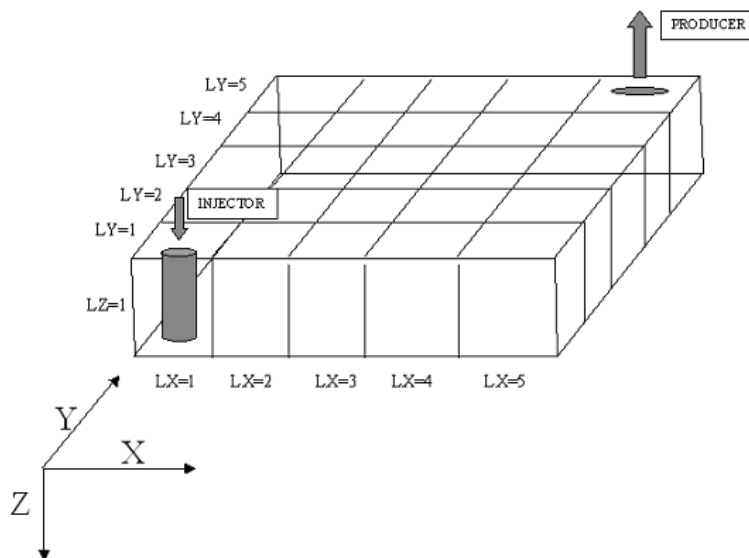
**Figure 3.3—Comparison of the water saturation profile from GPAS with the analytical solution and the result from UTCHEM at 0.1, 0.2 and 0.3 pore volumes injected for one-dimensional Buckley Leverett waterflooding problem.**

## Carbon Dioxide Sequestration Case

The injection of carbon dioxide into a two-dimensional, isotropic, homogenous aquifer was simulated. After making the simulations with GPAS and UTCOMP, the pressure, saturation and composition profiles were compared.

The simulation domain is 500 ft in length, 500 ft in width and 100 ft in thickness. A 5x5x1 grid was used for this simulation. Water is treated here as a component so that the solubility of the carbon dioxide in the liquid water phase can be calculated from the equation-of-state. Phase 2 is the active liquid water phase and phase 3 is the gas phase in these simulations. The usual water phase is 1, but it plays no significant role in these simulations. Its saturation was a constant equal to 0.2. Carbon dioxide is injected into the aquifer through the injector located at gridblock (1,1,1) at a bottom hole pressure of 2000 psi. The producer well is located at gridblock (5,5,1) with a constant bottom hole pressure of 1000 psi. The initial reservoir pressure is 1500 psi. Figure 3.4 shows a diagram of the reservoir. A summary of the input data is given in Table 3.5.

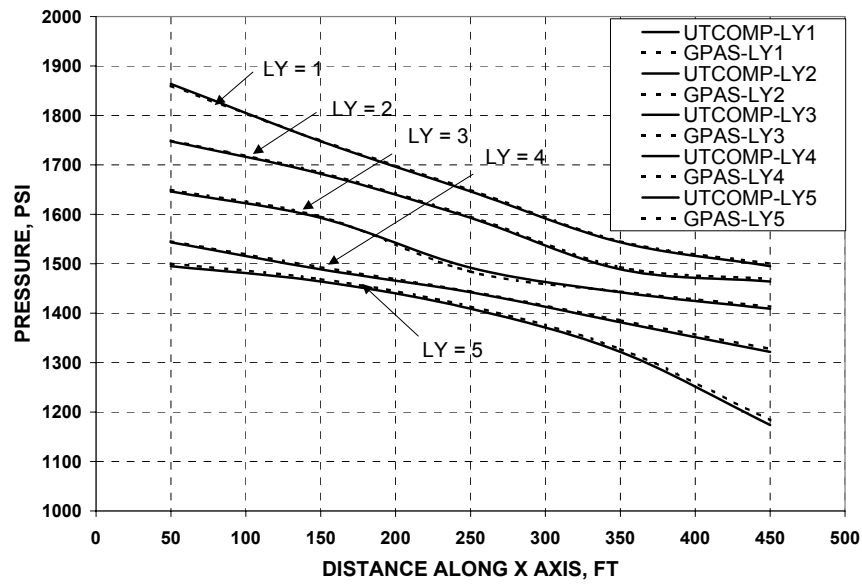
The simulation is carried out for 100 days using GPAS and UTCOMP and pressure, saturation and composition profiles at the end of 100 days compared with good agreement. The pressure profile along the x-axis has been plotted for every Y index in Fig. 3.5. The gas saturation profile along the x-axis was plotted for every Y grid index in Fig. 3.6. The compositions of the water phase and the gaseous phase at the end of 100 days from GPAS and UTCOMP are given in Table 3.6. The absolute differences between the compositions obtained from GPAS and UTCOMP at the end of 100 days are given in Table 3.7.



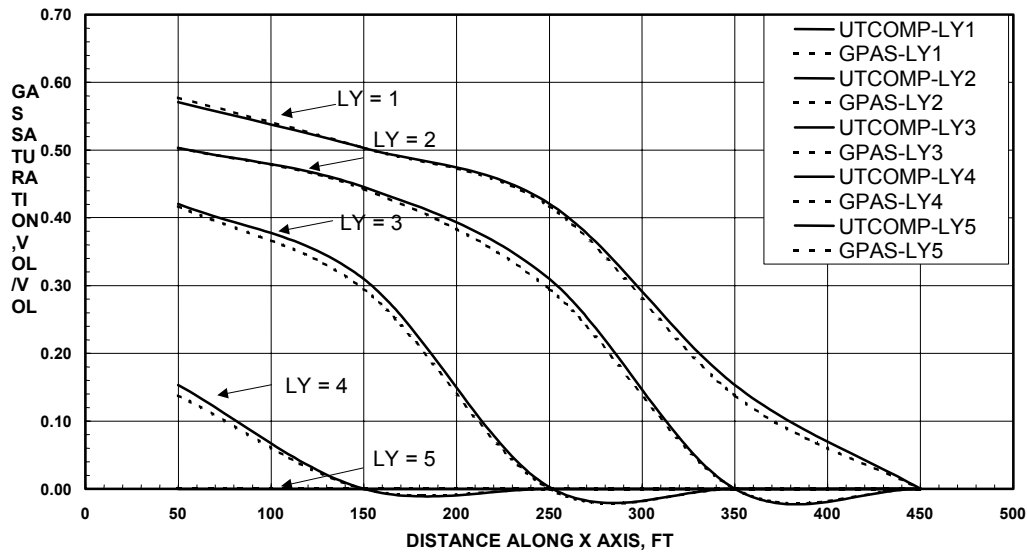
**Figure 3.4—Schematic reservoir geometry (quarter five spot) for the CO<sub>2</sub> sequestration problem.**

**Table 3.5—Summary of Input Data For Two Dimensional Carbon Dioxide Sequestration Problem**

Dimensions (ft)						
Length	500					
Width	500					
Thickness	100					
Porosity (fraction)	0.25					
Permeability (md)	10					
Initial Water Saturation (fraction)	0.2					
Reservoir Temperature (°F)	140					
Initial Reservoir Pressure (psi)	1500					
Initial Composition (mole fraction)						
CO <sub>2</sub>	0.002					
H <sub>2</sub> O	0.998					
Water viscosity (cp)	0.79					
Water Compressibility (1/psi)	6.7x10 <sup>-5</sup>					
Rock Compressibility (1/psi)	0.0					
Simulation Time (day)	100.0					
Constant Time Step Size (day)	0.1					
Injector Well Location (I,J,K)	(1,1,1)					
Injection Bottomhole Pressure (psi)	2000					
Injection Fluid Composition						
CO <sub>2</sub>	0.999					
Producer Well Location (I,J,K)	(5,5,1)					
Production Bottomhole Pressure (psi)	1000					
Component Critical Properties:						
<u>Component</u>	<u>T<sub>c</sub></u> (°R)	<u>P<sub>c</sub></u> (psi)	<u>V<sub>c</sub></u> (ft <sup>3</sup> /lb- mole)	<u>ω<sub>i</sub></u> (dyne <sup>1/4</sup> cm <sup>11/4</sup> / gm-mole)	<u>MW<sub>i</sub></u> (lbm/lb- mole)	<u>Ψ<sub>i</sub></u>
CO <sub>2</sub>	547.5	1071.0	1.505	0.23	44.01	49.0
H <sub>2</sub> O	1165.14	3207.4	0.79890	0.34400	18.02	100.0
Binary Interaction Coefficients:						
	CO <sub>2</sub>	H <sub>2</sub> O				
CO <sub>2</sub>	0	-0.085				
H <sub>2</sub> O	-0.085	0				
Relative Permeability Functions Used for Generating Tables:						
$K_{r1} = 0.4089 \left( \frac{S_1 - 0.2}{0.5} \right)^3$			$K_{r3} = 0.39 \left( \frac{S_3 - 0.05}{0.6} \right)^3$			
$K_{r21} = \left( \frac{0.7 - S_1}{0.5} \right)^2$			$K_{r23} = 0.83886 \left( \frac{0.65 - S_3}{0.6} \right)^{2.1952}$			
Capillary Pressure Functions Used for Generating Tables:						
$P_{c21} = 0.05 + 255.5328 \exp(-8.689S_1)$			$P_{c23} = 6.00524 \times 10^{-4} \exp(13.544618S_3)$			



**Figure 3.5—Comparison of the pressure profile along the x-axis obtained from GPAS with UTCOMP at the end of 100 days for CO<sub>2</sub> sequestration problem.**



**Figure 3.6—Comparison of the gas saturation profile along the x-axis from GPAS with UTCOMP at the end of 100 days for the CO<sub>2</sub> sequestration problem.**

**Table 3.6—Comparison of Mobile Water Phase and Gas Phase Compositions From GPAS With UTCOMP for Carbon Dioxide Sequestration Problem**

GPAS RESULT						UTCOMP RESULT					
CO2 molefraction in the Mobile Water Phase						CO2 molefraction in the Mobile Water Phase					
LX/LY(ft)	50	150	250	350	450	LX/LY(ft)	50	150	250	350	450
50	0.0215	0.0211	0.0207	0.0201	0.0049	50	0.021576	0.02113	0.020671	0.020119	0.004952
150	0.0211	0.0208	0.0204	0.0134	0.0031	150	0.02113	0.020841	0.020395	0.013835	0.003108
250	0.0207	0.0204	0.0193	0.0053	0.0024	250	0.020671	0.020395	0.019807	0.005362	0.002413
350	0.0201	0.0134	0.0053	0.0029	0.0022	350	0.020119	0.013835	0.005362	0.002924	0.002178
450	0.0049	0.0031	0.0024	0.0022	0.0021	450	0.004952	0.003108	0.002413	0.002178	0.002074

H2O molefraction in the Mobile Water Phase						H2O molefraction in the Mobile Water Phase					
LX/LY(ft)	50	150	250	350	450	LX/LY(ft)	50	150	250	350	450
50	0.9785	0.9789	0.9793	0.9799	0.9951	50	0.97842	0.97887	0.97933	0.97988	0.99505
150	0.9789	0.9792	0.9796	0.9866	0.9969	150	0.97887	0.97916	0.9796	0.98616	0.99689
250	0.9793	0.9796	0.9807	0.9947	0.9976	250	0.97933	0.9796	0.98019	0.99464	0.99759
350	0.9799	0.9866	0.9947	0.9971	0.9978	350	0.97988	0.98616	0.99464	0.99708	0.99782
450	0.9951	0.9969	0.9976	0.9978	0.9979	450	0.99505	0.99689	0.99759	0.99782	0.99793

CO2 molefraction in the Gas Phase						CO2 molefraction in the Gas Phase					
LX/LY(ft)	50	150	250	350	450	LX/LY(ft)	50	150	250	350	450
50	0.9808	0.9838	0.9863	0.9885	0	50	0.98077	0.98388	0.98641	0.98853	0
150	0.9838	0.9865	0.9875	0	0	150	0.98388	0.98556	0.98758	0	0
250	0.9863	0.9875	0	0	0	250	0.98641	0.98758	0.98938	0	0
350	0.9885	0	0	0	0	350	0.98853	0	0	0	0
450	0	0	0	0	0	450	0	0	0	0	0

H2O molefraction in the Gas Phase						H2O molefraction in the Gas Phase					
LX/LY(ft)	50	150	250	350	450	LX/LY(ft)	50	150	250	350	450
50	0.0192	0.0162	0.0137	0.0115	0	50	0.019229	0.01612	0.013588	0.011473	0
150	0.0162	0.0145	0.0125	0	0	150	0.01612	0.014439	0.012419	0	0
250	0.0137	0.0125	0	0	0	250	0.013588	0.012419	0.010622	0	0
350	0.0115	0	0	0	0	350	0.011473	0	0	0	0
450	0	0	0	0	0	450	0	0	0	0	0

**Table 3.7—Absolute Differences in Mobile Water Phase and Gas Phase Compositions From GPAS With UTCOMP for Carbon Dioxide Sequestration Problem After 100 Days**

CO2 molefraction in the Mobile Water Phase						CO2 molefraction in the Gas Phase					
LX/LY(ft)	50	150	250	350	450	LX/LY(ft)	50	150	250	350	450
50	-7.6E-05	-3E-05	2.9E-05	-1.9E-05	-5.2E-05	50	3E-05	-8E-05	-0.00011	-3E-05	0
150	-3E-05	-4.1E-05	5E-06	-0.00044	-7.6E-06	150	-8E-05	-6E-05	-8E-05	0	0
250	2.9E-05	5E-06	-0.00051	-6.2E-05	-1.3E-05	250	-0.00011	-8E-05	-0.98938	0	0
350	-1.9E-05	-0.00044	-6.2E-05	-2.4E-05	2.17E-05	350	-3E-05	0	0	0	0
450	-5.2E-05	-7.6E-06	-1.3E-05	2.17E-05	2.56E-05	450	0	0	0	0	0

H2O molefraction in the Mobile Water Phase						H2O molefraction in the Gas Phase					
LX/LY(ft)	50	150	250	350	450	LX/LY(ft)	50	150	250	350	450
50	8E-05	3E-05	-3E-05	2E-05	5E-05	50	-2.9E-05	8E-05	0.000112	2.7E-05	0
150	3E-05	4E-05	0	0.00044	1E-05	150	8E-05	6.1E-05	8.1E-05	0	0
250	-3E-05	0	0.00051	6E-05	1E-05	250	0.000112	8.1E-05	-0.01062	0	0
350	2E-05	0.00044	6E-05	2E-05	-2E-05	350	2.7E-05	0	0	0	0
450	5E-05	1E-05	1E-05	-2E-05	-3E-05	450	0	0	0	0	0

## Six-Component Compositional Simulation Example

This test is a modified version of the SPE fifth comparative solution problem. The hydrocarbon phase consists of six components. The hydrocarbon component critical properties are given in Table 3.8. Two-dimensional and three-dimensional simulations of the same problem were run to verify the correctness of the two- and three-dimensional features of the simulator.



**Table 3.8—Summary of Component Critical Properties and Binary Interaction Coefficients for Two-Dimensional Six Component Simulation Example**

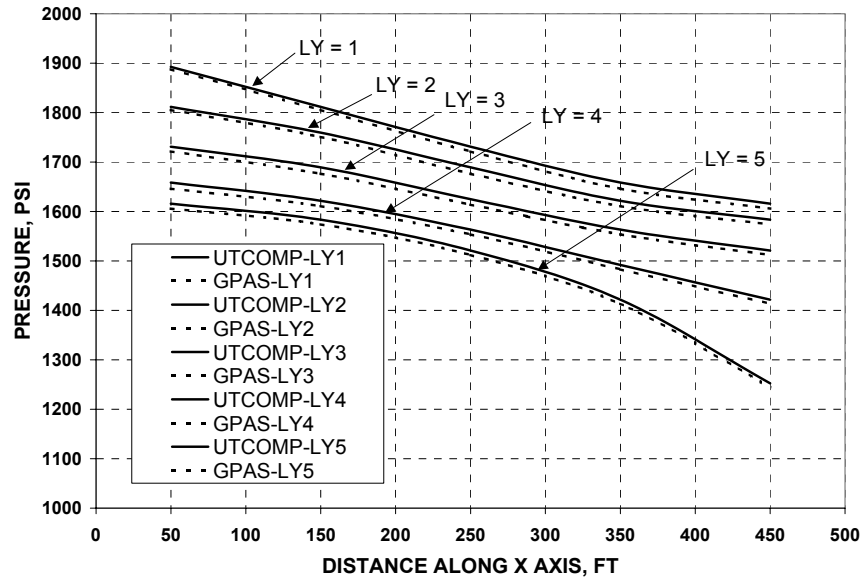
Component Critical Properties:						
<i>Component</i>	$T_c$ (°R)	$P_c$ (psi)	$V_c$ (ft <sup>3</sup> /lb- mole)	$\omega_i$ (dyne <sup>1/4</sup> cm <sup>11/4</sup> / gm-mole)	$MW_i$ (lbm/lb- mole)	$\Psi_i$
CH <sub>4</sub>	343.0	667.8	1.599	0.013	16.0	71
C <sub>3</sub> H <sub>8</sub>	665.7	616.3	3.211	0.152	44.1	151
C <sub>6</sub> H <sub>14</sub>	913.4	436.9	5.923	0.301	86.2	271
C <sub>10</sub> H <sub>22</sub>	1111.8	304.0	10.087	0.488	142.3	431
C <sub>15</sub> H <sub>32</sub>	1270.0	200.0	16.696	0.650	206.0	631
C <sub>20</sub> H <sub>42</sub>	1380.0	162.0	21.484	0.850	282.0	831
Binary Interaction Coefficients:						
	CH <sub>4</sub>	C <sub>3</sub> H <sub>8</sub>	C <sub>6</sub> H <sub>14</sub>	C <sub>10</sub> H <sub>22</sub>	C <sub>15</sub> H <sub>32</sub>	C <sub>20</sub> H <sub>42</sub>
CH <sub>4</sub>	0	0	0	0	0.05	0.05
C <sub>3</sub> H <sub>8</sub>	0	0	0	0	0.005	0.005
C <sub>6</sub> H <sub>14</sub>	0	0	0	0	0	0
C <sub>10</sub> H <sub>22</sub>	0	0	0	0	0	0
C <sub>15</sub> H <sub>32</sub>	0.05	0.005	0	0	0	0
C <sub>20</sub> H <sub>42</sub>	0.05	0.005	0	0	0	0

### Two Dimensional Quarter Five Spot Case

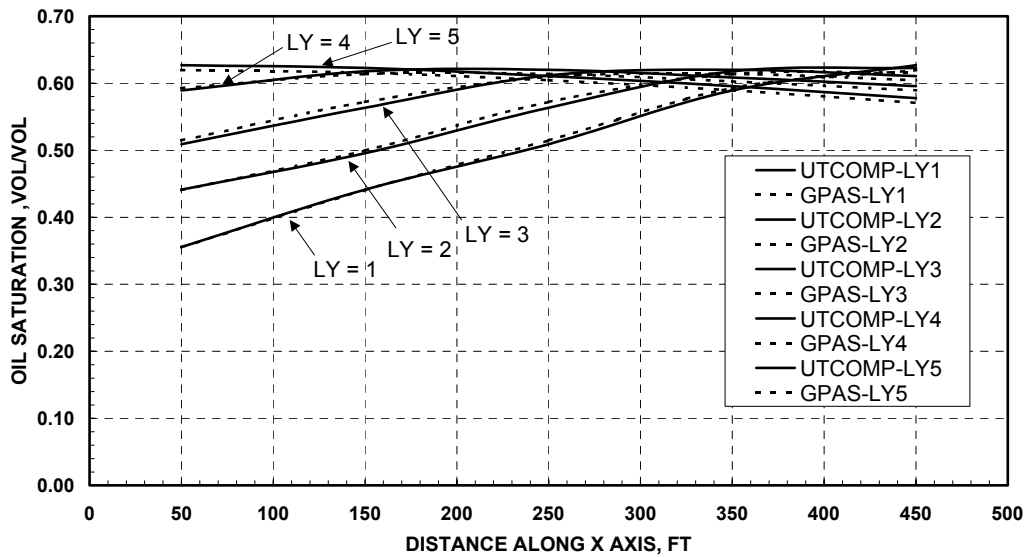
A quarter five spot pattern (5x5x1) is used for this run. The reservoir description and the well constraints are presented in Table 3.9. The simulation domain comprises of 500 ft in the x direction, 500 ft in the y direction and 100 ft in the z direction. The injector well is located at (1,1,1) and the producer well is located at (5,5,1). Gas is injected into the reservoir and it pushes the oil towards the producer. The water phase is immobile. The pressure profile and the saturation profile at the end of 100 days obtained from GPAS and UTCOMP was compared and a good agreement was obtained. The pressure profile along the x-axis has been plotted for every Y index in Fig. 3.7. The oil saturation profile and the gas saturation profile along the x-axis for every y index is plotted in Figs. 3.8 and 3.9 respectively and as shown there is a reasonably good agreement obtained between the two simulator results. This supports the correctness of the two dimensional three phase compositional capabilities of GPAS.

**Table 3.9—Summary of Input Data for Two-Dimensional Six Component Simulation Example**

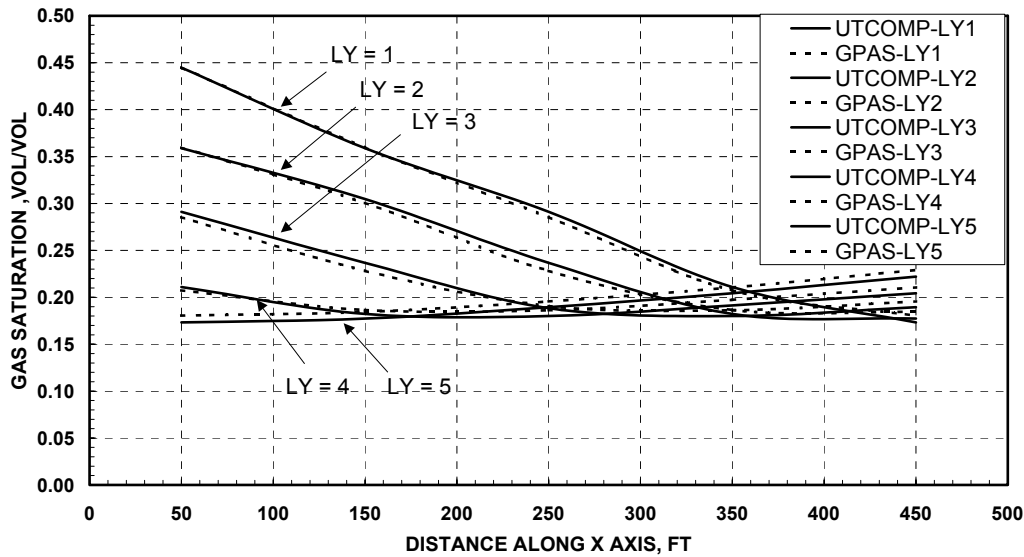
Dimensions (ft)		
Length		500
Width		500
Thickness		100
Porosity (fraction)		0.35
Permeability (md)		10
Initial Water Saturation (fraction)		0.2
Residual Water Saturation (fraction)		0.2
Reservoir Temperature (°F)		160
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)		
CH <sub>4</sub>		0.5
C <sub>3</sub> H <sub>8</sub>		0.03
C <sub>6</sub> H <sub>14</sub>		0.07
C <sub>10</sub> H <sub>22</sub>		0.2
C <sub>15</sub> H <sub>32</sub>		0.15
C <sub>20</sub> H <sub>42</sub>		0.05
Water viscosity (cp)		0.337
Water Compressibility (1/psi)		3.0x10 <sup>-6</sup>
Rock Compressibility (1/psi)		1.0x10 <sup>-6</sup>
Simulation Time (day)		100.0
Constant Time Step Size (day)		0.1
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		2000
Injection Fluid Composition		
CH <sub>4</sub>		0.77
C <sub>3</sub> H <sub>8</sub>		0.20
C <sub>6</sub> H <sub>14</sub>		0.03
Producer Well Location (I,J,K)		(5,5,1)
Production Bottomhole Pressure (psi)		1000
Relative Permeability Functions Used for Generating Tables:		
$K_{r1} = 0.4089 \left( \frac{S_1 - 0.2}{0.5} \right)^3$		
$K_{r3} = 0.39 \left( \frac{S_3 - 0.05}{0.6} \right)^3$		
$K_{r21} = \left( \frac{0.7 - S_1}{0.5} \right)^2$		
$K_{r23} = 0.83886 \left( \frac{0.65 - S_3}{0.6} \right)^{2.1952}$		
Capillary Pressure Functions Used for Generating Tables:		
$P_{c21} = 0.05 + 255.5328 \exp(-8.689S_1) \quad P_{c23} = 6.00524 \times 10^{-4} \exp(13.544618S_3)$		



**Figure 3.7—Comparison of the pressure profile along the x-axis from GPAS with UTCOMP at the end of 100 days for the six component problem.**



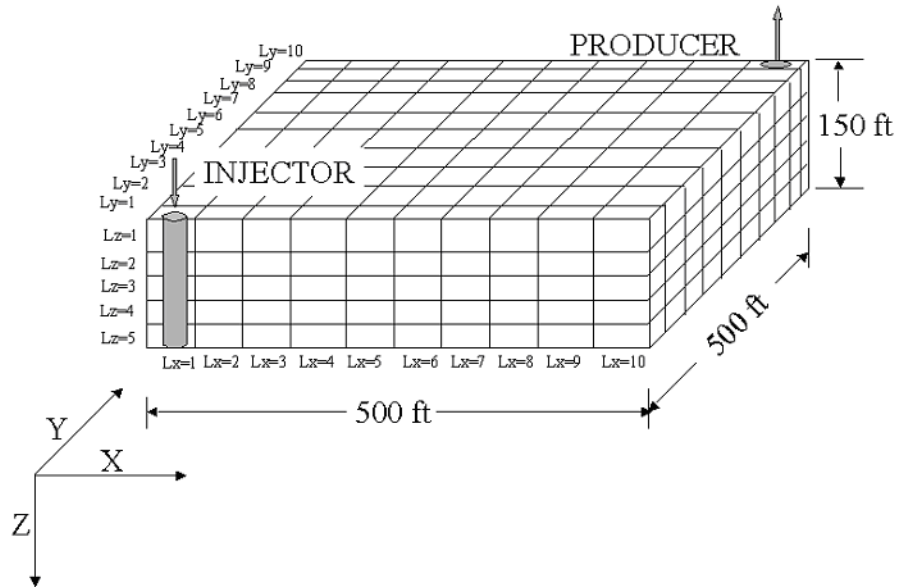
**Figure 3.8—Comparison of the oil saturation profile along the x-axis from GPAS with UTCOMP at the end of 100 days for the six component problem.**



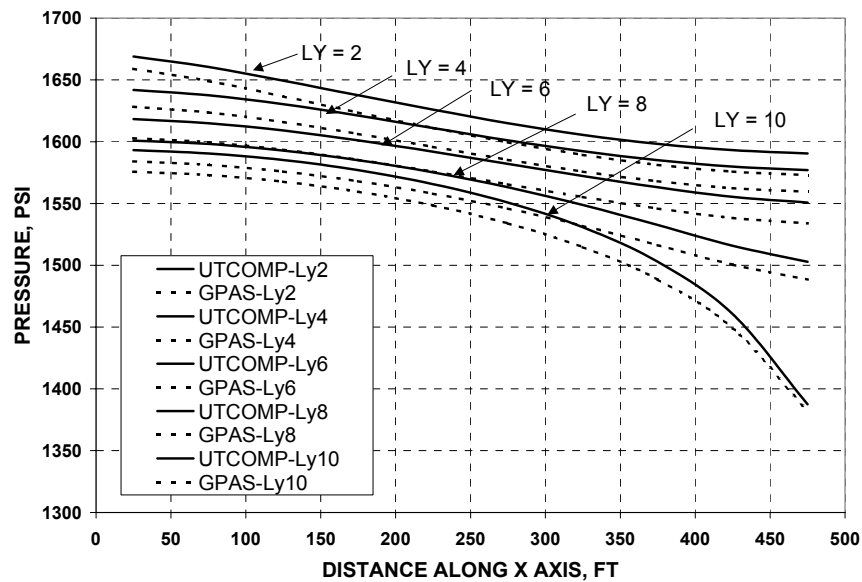
**Figure 3.9—Comparison of the gas saturation profile along the x axis from GPAS with UTCOMP at the end of 100 days for the six component problem.**

### Three Dimensional Case

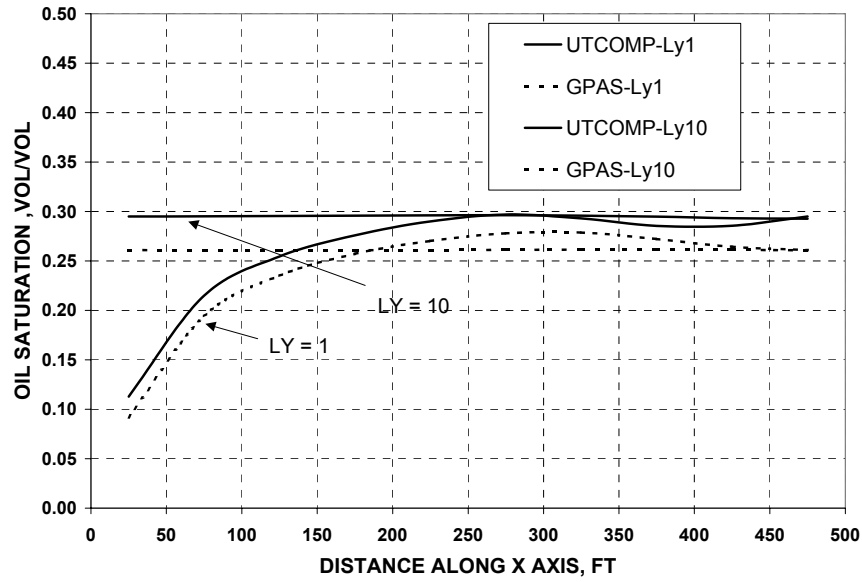
A 10x10x5 finite difference grid was used as shown in Fig. 3.10. The input data summary for this run is given in Table 3.10. The critical properties and the binary interaction coefficients are given in Table 3.8. The injector well and the producer well are completed through all the five layers. The injector is located at (1,1) and the producer well is located at (10,10). Gas is continuously injected at a constant bottomhole pressure of 1700 psi and the producer well is constantly maintained at 1300 psi. The initial reservoir pressure is 1500 psi. The water phase is immobile. The test case is run using GPAS and UTCOMP for a simulation period of two years. The pressure distribution, fluid saturations and concentrations from GPAS compared well with UTCOMP. The plots of the pressure profile along the x axis for the even grid index in the Y direction and only for the top layer at the end of two years is plotted in Fig. 3.11. The oil saturation profile and the gas saturation profile along the x-axis, only for the extreme Y grid index in the top layer and at the end of two years, from GPAS and UTCOMP were compared in Figs. 3.12 and 3.13 respectively. The average reservoir pressure history from GPAS and UTCOMP were compared in Fig. 3.14. The overall production gas oil ratio comparison plots are shown in Fig. 3.15.



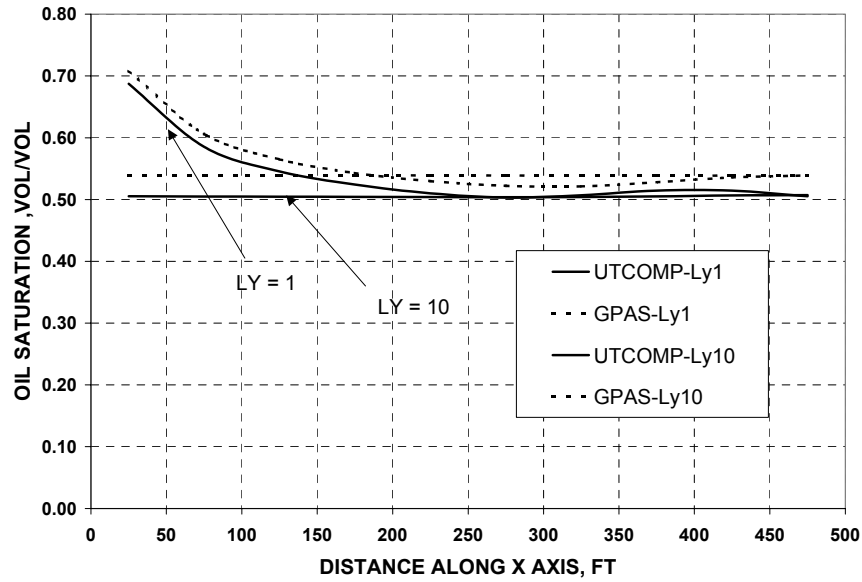
**Figure 3.10—Schematic reservoir geometry for the three dimensional six component simulation case.**



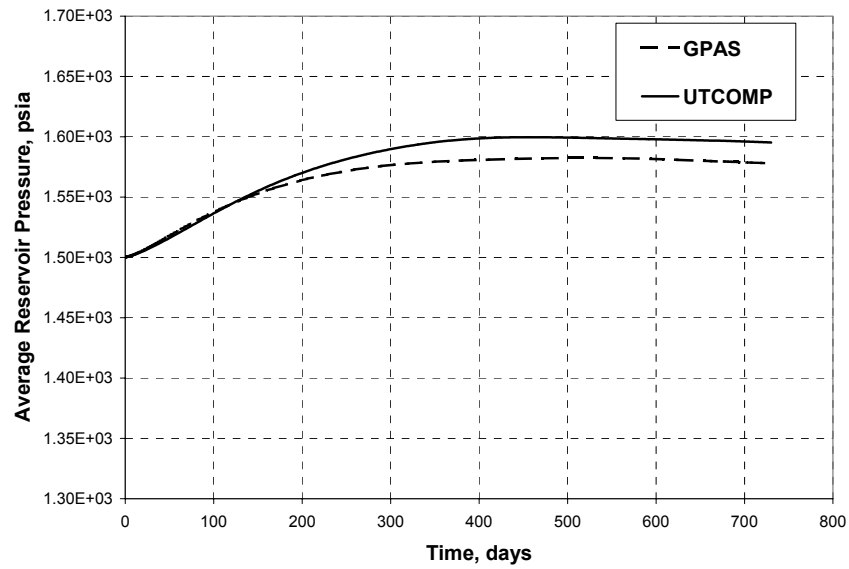
**Figure 3.11—Comparison of the pressure profile along the x-axis from GPAS with UTCOMP for the six component case at the end of two years.**



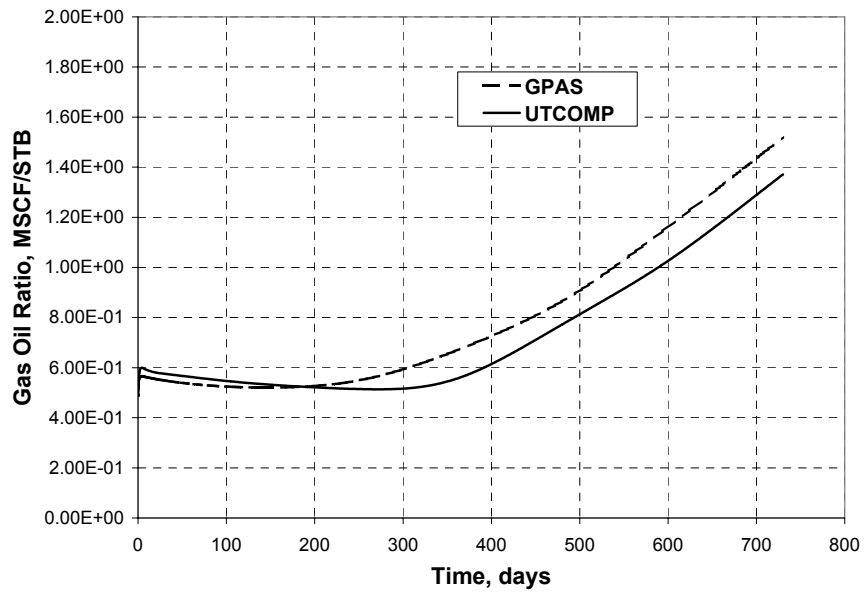
**Figure 3.12—Comparison of the oil saturation profile along the x-axis from GPAS with UTCOMP for the six component case at the end of two years.**



**Figure 3.13—Comparison of the gas saturation profile along the x-axis from GPAS with UTCOMP for the six component case at the end of two years.**



**Figure 3.14—Comparison of the average reservoir pressure history from GPAS with UTCOMP for the six component case.**



**Figure 3.15—Comparison of the producer GOR (gas-oil ratio) history of GPAS with UTCOMP result for the six component case.**

**Table 3.10—Summary of Input Data for Three-Dimensional Six Component Simulation Example**

Dimensions (ft)		
Length		500
Width		500
Thickness		150
Porosity (fraction)		0.35
Permeability (md)		10
Initial Water Saturation (fraction)		0.2
Residual Water Saturation (fraction)		0.2
Reservoir Temperature (°F)		160
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)		
CH <sub>4</sub>		0.5
C <sub>3</sub> H <sub>8</sub>		0.03
C <sub>6</sub> H <sub>14</sub>		0.07
C <sub>10</sub> H <sub>22</sub>		0.2
C <sub>15</sub> H <sub>32</sub>		0.15
C <sub>20</sub> H <sub>42</sub>		0.05
Water viscosity (cp)		0.337
Water Compressibility (1/psi)		3.0x10 <sup>-5</sup>
Rock Compressibility (1/psi)		1.0x10 <sup>-6</sup>
Simulation Time (day)		730.0
Constant Time Step Size (day)		0.1
Injector Well Top Location (I,J,K)		(1,1,1)
Injector Well Bottom Location (I,J,K)		(1,1,5)
Injection Bottomhole Pressure (psi)		1700
Injection Fluid Composition		
CH <sub>4</sub>		0.77
C <sub>3</sub> H <sub>8</sub>		0.20
C <sub>6</sub> H <sub>14</sub>		0.03
Producer Well Top Location (I,J,K)		(10,10,1)
Producer Well Bottom Location (I,J,K)		(10,10,5)
Production Bottomhole Pressure (psi)		1300
Relative Permeability Functions Used for Generating Tables:		
$K_{r1} = 0.4089 \left( \frac{S_1 - 0.2}{0.5} \right)^3$		
$K_{r3} = 0.39 \left( \frac{S_3 - 0.05}{0.6} \right)^3$		
$K_{r21} = \left( \frac{0.7 - S_1}{0.5} \right)^2$		
$K_{r23} = 0.83886 \left( \frac{0.65 - S_3}{0.6} \right)^{2.1952}$		
Capillary Pressure Functions Used for Generating Tables:		
$P_{c21} = 0.05 + 255.5328 \exp(-8.689S_1) \quad P_{c23} = 6.00524 \times 10^{-4} \exp(13.544618S_3)$		



## Chemical Module Tests—Explicit Formulation

The chemical module of the GPAS simulator has been validated by comparing results with analytical solutions for one dimensional aqueous species transport and with the results of The University of Texas Chemical Simulator (UTCHEM). In this section, we present a comparison with UTCHEM that has been validated by comparisons with analytical solutions, laboratory surfactant floods and a variety of laboratory and field results.

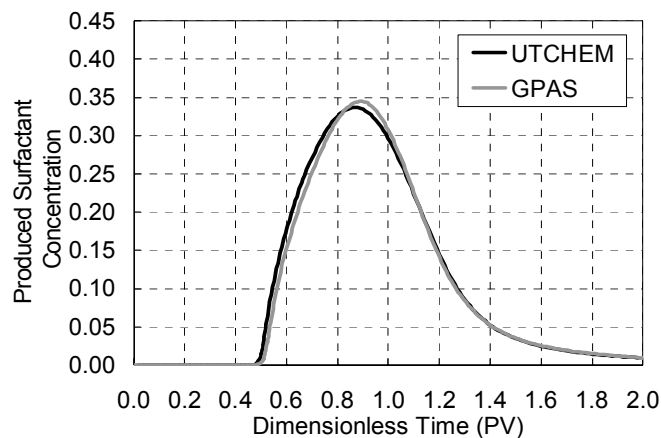
### 2-D Surfactant/Polymer Simulation, Case 1

A two-dimensional test case was setup to model surfactant-polymer slug injection in a quarter of five-spot well pattern. The input data used is summarized in Table 3.11. The reservoir is 480 ft in each direction modeled by a 15x15 grid. The rock properties and relative permeability input are set for typical water wet conditions. The polymer viscosity parameters resulted in viscosity of about 12 cp at the maximum polymer concentration used in the simulation (0.05 wt%). The oil phase was modeled using single component n-decane and the resulting phase properties (as computed by the EOS model) were used in UTCHEM. It must be noted that UTCHEM uses constant fluid compressibility and density whereas these are updated for each timestep of the EOS model based on the composition and pressure. For the sake of comparison, model parameters are set to allow not much variation in fluid properties over the range of changes encountered during the run.

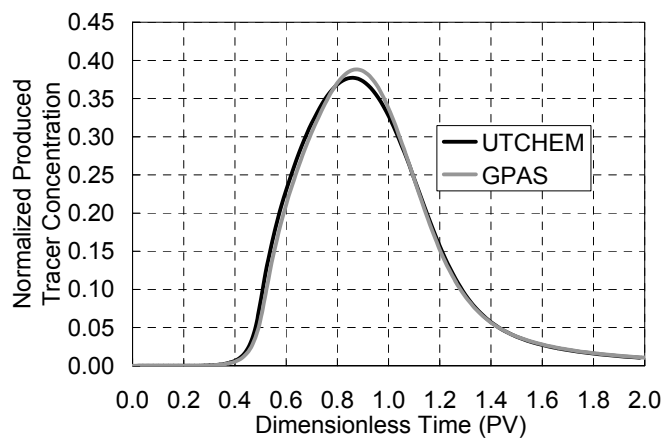
The results are compared with UTCHEM output in Figs. 3.16 through 3.25. The surfactant, tracer and polymer concentration histories agree well (Figs. 3.16, 3.17 and 3.18). Figures 3.19 and 3.20 show the polymer concentration profiles at 0.5 PV injected, which agree well. The surfactant concentration profiles at the same time are shown in Figs. 3.21 and 3.22. The UTCHEM profile shows a slightly larger spread than GPAS. The oil saturation profiles (Fig. 3.23 and 3.24) show the effect of the chemical slug. The mobilized oil is pushed ahead of the surfactant forming an oil bank and leaving very little oil in the swept region of the reservoir. UTCHEM oil saturations are slightly lower in the swept region because it accounts for additional oil removal by solubilization. Figure 3.25 shows the comparison of oil production rates and cumulative oil recovery. The breakthrough times for the oil bank and the oil production rates are in satisfactory agreement. The cumulative oil recovery using UTCHEM is 7% higher than for GPAS due to the additional solubilized oil not accounted for in GPAS.

**Table 3.11—Input parameters used for Case 1**

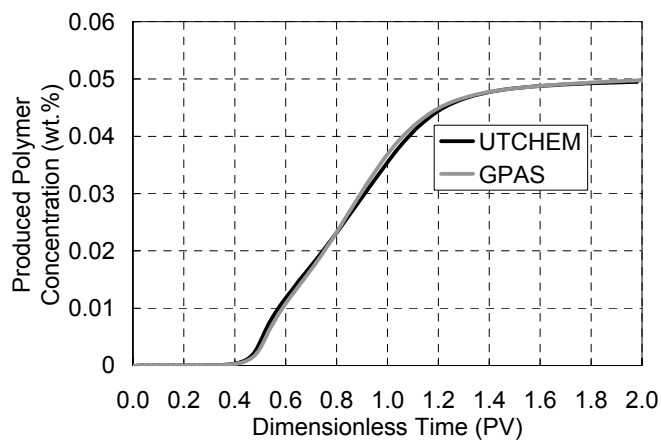
<b>Property</b>	<b>Value</b>
Reservoir Size	480ft x 480ft x 50ft
Reservoir Gridblock Size	32ft x 32ft x 50ft
Number of Gridblocks	15 x 15 x 1
Matrix Permeability	540 md
Matrix Porosity	0.136
Initial Water Saturation	0.75
Initial Pressure	200 psi
Water Viscosity	1 cp
Oil Viscosity	2.6 cp
Oil Specific Gravity	0.3858 psi/ft
Oil Compressibility	$3 \times 10^{-6}$ 1/psi
Polymer Viscosity Parameters	81, 2700, 2500
<i>Surfactant Properties</i>	
Heights of Binodal Curve	0.07, 0.04 vol. fraction
Salinity Limits	0.177, 0.25 meq/ml
CMC	0.0001 vol. fraction
Adsorption Parameters	1.5, 0.5, 1000
IFT Correlation Parameters (a,c)	9, 0.2
<i>Relative Permeability Model</i>	
Residual Saturations at Low Trapping Number	0.14, 0.25, 0.0
Residual Saturations at High Trapping Number	0.0, 0.0, 0.0
Endpoints at Low Trapping Number	0.106, 0.8, 0.0
Endpoints at High Trapping Number	1.0, 1.0, 1.0
Exponents at Low Trapping Number	2.1, 1.7, 0.0
Exponents at High Trapping Number	0.48, 1.5, 0.0
Trapping Number Parameters	364, 59074, 364
Injection Rate	1.0, 1.0, 1.0
Production Constraint (BHP)	3000 ft <sup>3</sup> /day
Chemical Slug Size	200.0 psi
Chemical Slug Composition	0.2 PV
	0.17 meq/ml salt, 0.05 vol. fraction, surfactant, 0.05 wt % polymer, water, tracer
Chase Fluid Composition	0.17 meq/ml salt, 0.05 wt % polymer, water
Simulation Time	2 PV



**Figure 3.16—Comparison of produced surfactant concentrations for Case 1.**



**Figure 3.17—Comparison of produced tracer concentration for Case 1.**



**Figure 3.18—Comparison of produced polymer concentration for Case 1.**

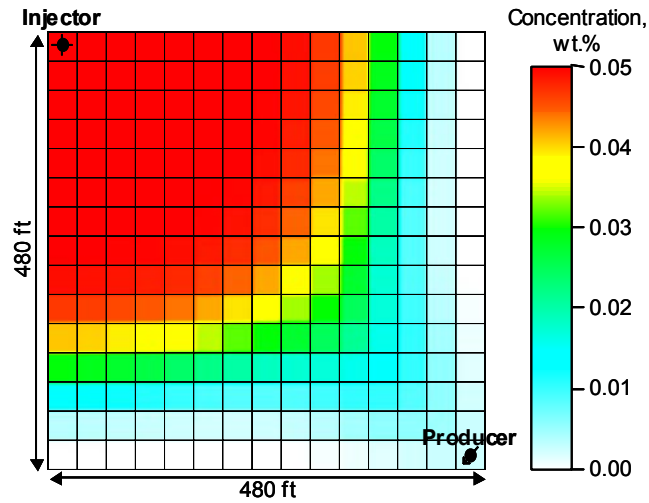


Figure 3.19—Polymer concentration (GPAS) at  $T_d=0.5$  PV.

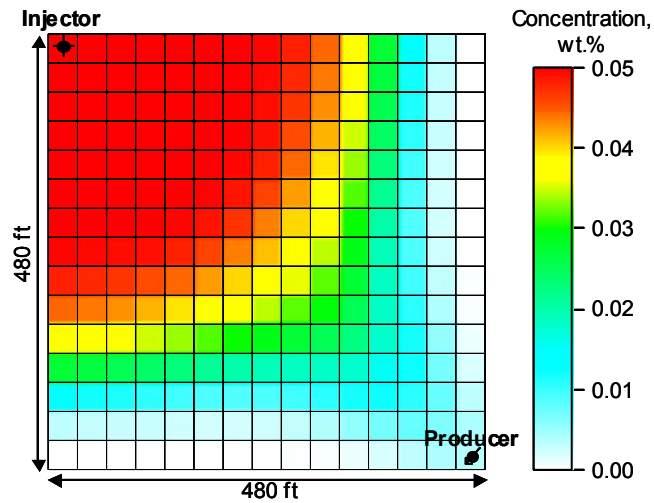


Figure 3.20—Polymer concentration (UTCHEM) at  $T_d=0.5$  PV.

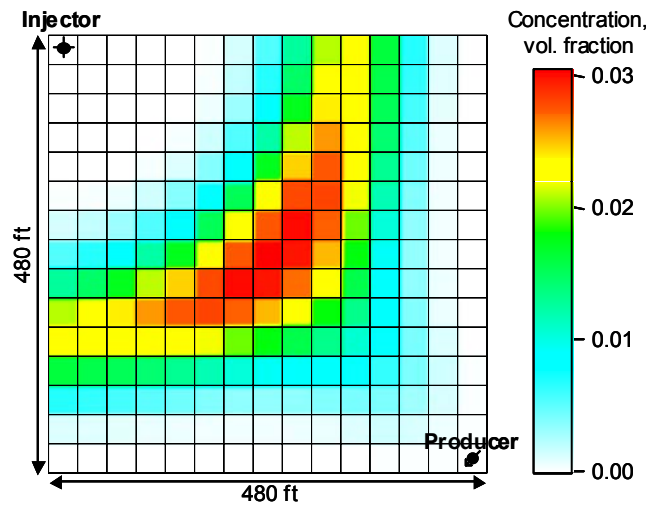


Figure 3.21—Surfactant concentration in fluid (GPAS) at  $T_d=0.5$  PV.

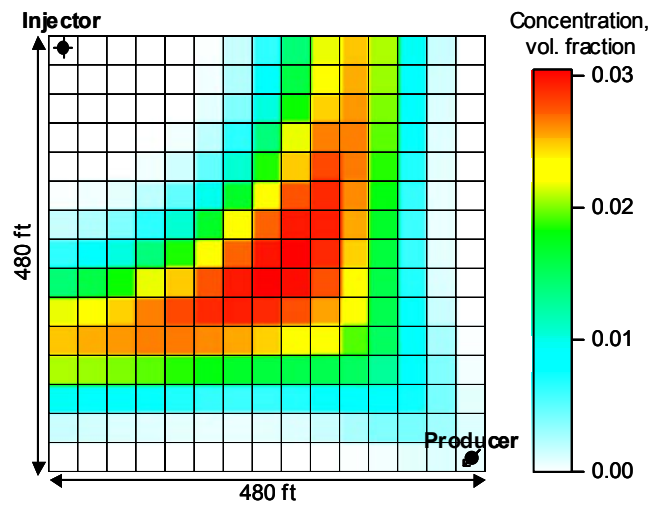


Figure 3.22—Surfactant concentration in fluid (UTCHEM) at  $T_d=0.5$  PV.

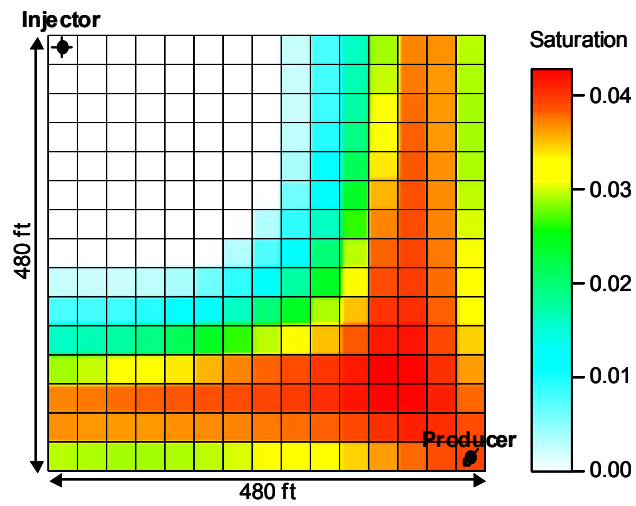


Figure 3.23—Oil saturation profile (UTCHEM) at  $T_d=0.5$  PV.

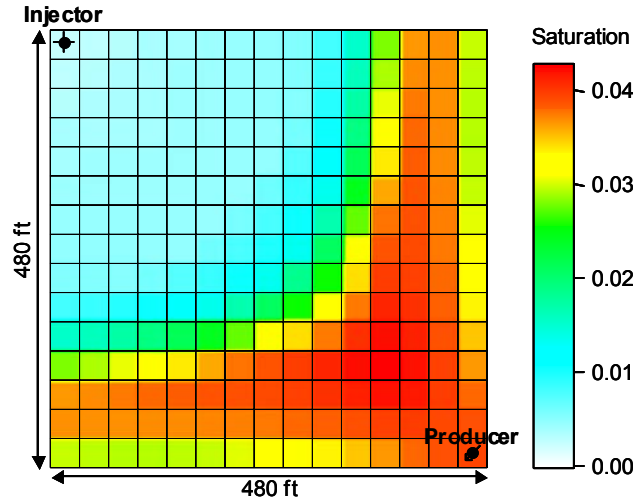


Figure 3.24—Oil saturation profile (GPAS) at  $T_d=0.5$  PV.

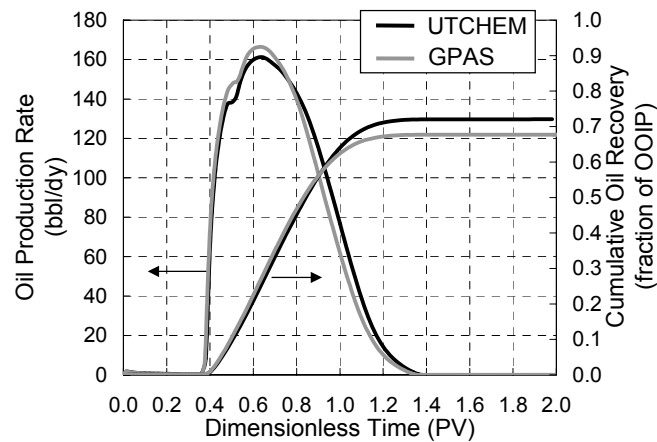
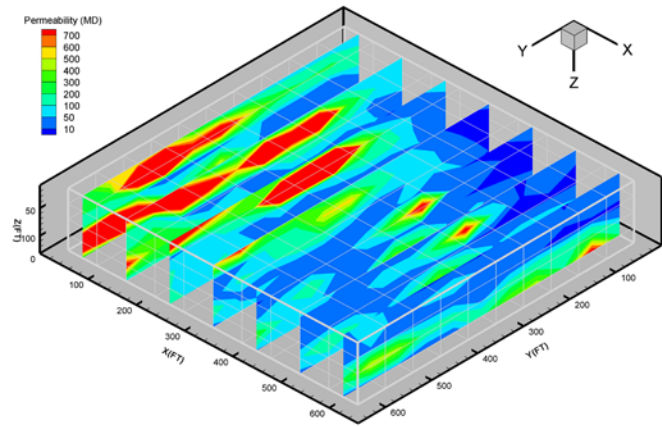


Figure 3.25—Comparison of oil production rate and cumulative oil production for Case 1.

### 3-D Surfactant/polymer simulation, Case 2

The 3-D reservoir simulated in this example is an idealization of a light oil mid-continent US sandstone reservoir. This reservoir is assumed to be waterflooded to an initial oil saturation of about 0.30. Since the assumed waterflood residual saturation is 0.25, most of the oil production after the start of the chemical flood can be considered tertiary oil recovery. The reservoir is 3500 ft deep and 140 ft thick. The oil viscosity is 7.78 cp. The permeability distribution was generated by a stochastic method with a geometric mean of about 50 md and a Dykstra-Parsons coefficient of 0.80. A spherical variogram and a log-normal permeability distribution were used. Correlation lengths of 660 ft in the x and y directions and 28 ft in the z direction were used. Figure 3.26 shows the permeability field. The vertical permeability was 10 md. A uniform porosity of 0.136 was used. The well arrangement is a quarter symmetry element of a forty-acre five-spot well pattern. The injection and production wells were located in the opposite corners and

perforated across the entire reservoir thickness. Both wells are operating at constant bottom hole pressure. The reservoir and fluid properties are given in Table 3.12.



**Figure 3.26—Permeability profile.**

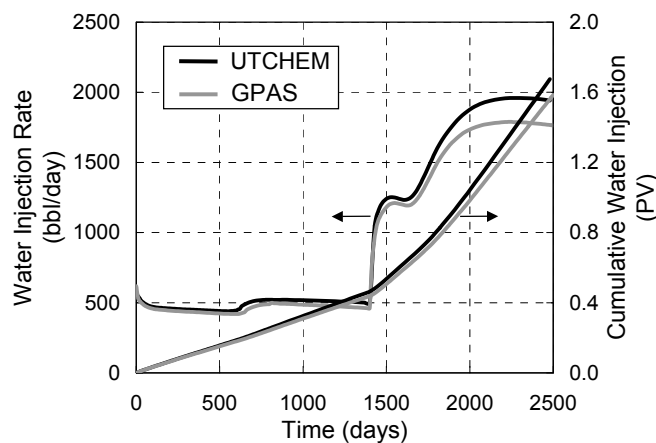
**Table 3.12—Input Parameters for Case 2**

Property	Value
Reservoir Size	660ft x 660ft x 140ft
Reservoir Gridblock Size	60ft x 60ft x 28ft
No. of Gridblocks	11 x 11 x 5
Matrix Permeability	Stochastic, MDM method using $V_{DP}=0.8$ , $K_{avg}=50\text{mD}$ (geom), 180mD (arithmetic)
Matrix Porosity	0.136
Initial Water Saturation	0.7
Water Viscosity	0.7 cp
Oil Viscosity	7.78 cp
Oil Specific Gravity	0.2678 psi/ft
Oil Compressibility	$7.57 \times 10^{-6}$ 1/psi
Polymer Viscosity Parameters	81, 2500, 2700
Surfactant Properties	
Heights of Binodal Curve	0.016, 0.01 vol. fraction
Salinity Limits	0.55, 0.91 meq/ml
CMC	0.00006 vol. fraction
Adsorption Parameters	1.0, 0.5, 1000
Initial Reservoir Pressure	900 psi
Injection Pressure	1500 psi
Production Constraint (BHP)	100 psi
Chemical Slug Injection Time	800 days
Chemical Slug Composition	0.54 meq/ml salt, 0.01 vol. fraction, surfactant, 0.05 wt % polymer, water
Polymer Drive Injection Time	600 days
Polymer Drive Composition	0.54 meq/ml salt, 0.05 wt % polymer, water
Post Flush	0.54 meq/ml brine
Simulation Time	2500 days

A chemical slug containing 0.01 volume fraction surfactant and 0.05 wt% (500 ppm) polymer was injected followed by a polymer drive of the same concentration and lastly a water drive. The salinity was constant and equal to 0.5 4 meq/ml (19,500 ppm), a value below the optimum salinity i.e. in the Type II(-) region. The relative permeability and interfacial tension correlation parameters are the same as in Case 1. No alcohol was needed with this surfactant. The surfactant/polymer slug was injected for 800 days and polymer drive was injected for 600 days. The model was setup and simulated both in GPAS and UTCHEM and the results are compared as discussed below.

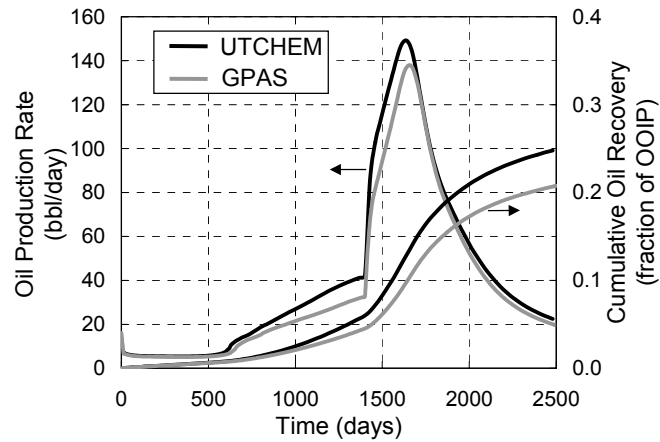
The injection rate was about 500 STB/d during the chemical slug and then increased to about 1800 STB/d during the water injection since it is less viscous (Fig. 3.27). Figure 3.28 shows the rate of oil production. Oil rate starts fairly low at less than 20 STB/d during the chemical slug injection and it peaks at about 140 STB/d during the water drive. The injection and production histories show good agreement between the simulators. As discussed in Case 1, the cumulative oil recovery predicted by GPAS is lower than UTCHEM by about 5% due to the neglect of solubilized oil. The average reservoir pressure is compared in Fig. 3.29 and show excellent agreement for this heterogeneous case. Several cross sections of the surfactant concentration at the end of chemical slug of 0.10 PV and oil saturation during the water drive at about 0.5 PV are shown in Figs. 3.30 and 3.31. Oil saturation at some locations was reduced to 0.05 from the initial value of 0.30 at the end of the simulation period.

The intention here was not to optimize the chemical flood for this reservoir, but simply to show a realistic simulation of a chemical flood with GPAS. A better chemical flood design for this reservoir included horizontal wells.

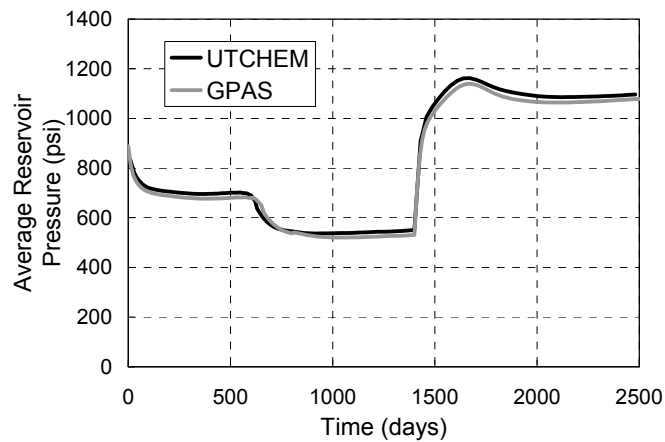


**Figure 3.27—Injection rate and cumulative fluid injected for Case 2.**

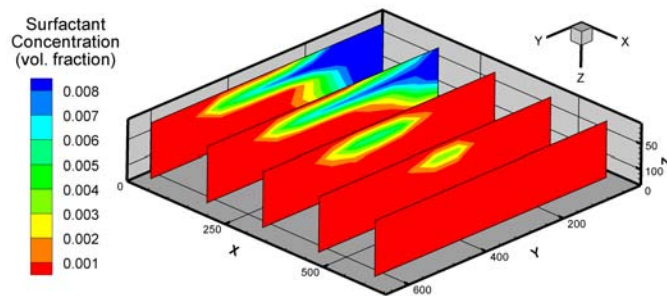




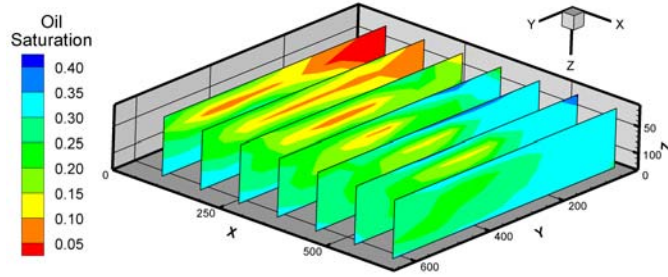
**Figure 3.28—Oil production rate and cumulative oil recovery for Case 2.**



**Figure 3.29—Average reservoir pressure for Case 2.**



**Figure 3.30—Surfactant concentration distribution at the end of chemical slug of 0.1 PV.**

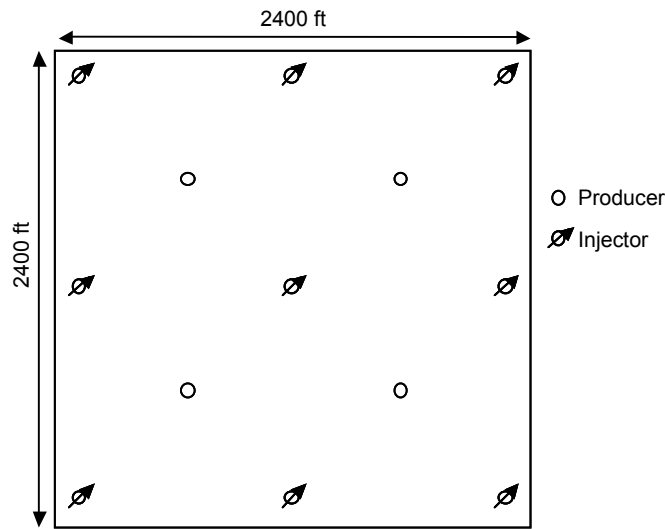


**Figure 3.31—Oil saturation distribution during post water flush at 0.5 PV.**

### 3-D Surfactant/polymer simulation, Case 3

Many previous test runs have been performed using the fully implicit EOS compositional model to simulate gas injection. The case described next was setup to test the parallel performance of the simulator with the chemical model. An oil reservoir of dimensions 2400 ft x 2400 ft x 12 ft having 13 wells in a five-spot pattern was simulated (Fig. 3.32). The numerical grid was 96x96x12 (110,592 gridblocks) in the x, y and z directions, respectively. The fluid properties, relative permeability data and chemical parameters are the same as for Case 1. The reservoir was initialized at residual oil saturation and 900 psi pressure. The chemical slug injected in all the injection wells at a constant pressure of 1500 psi. The input data is summarized in **Table 3.13**.

A Dell Power Edge 1750 cluster was used as the test platform. Each node is a dual processor Intel Xeon processor running at 3.06 GHz having 2 GB RAM and 533 MHz bus speed. The nodes are connected by a 100 Mbps ethernet switch for communication. The cluster runs Red Hat Linux v9.0 with GNU C/C++ and Portland Group Fortran 77/90 compilers version 5.0, MPICH version 1.2.4, which is a portable implementation of the Message Passing Interface (MPI) from Argonne National Laboratory for parallel communication, and PETSc version 2.1.6 with incomplete LU preconditioner and Generalized Minimum Residual Algorithm (GMRES) solver.



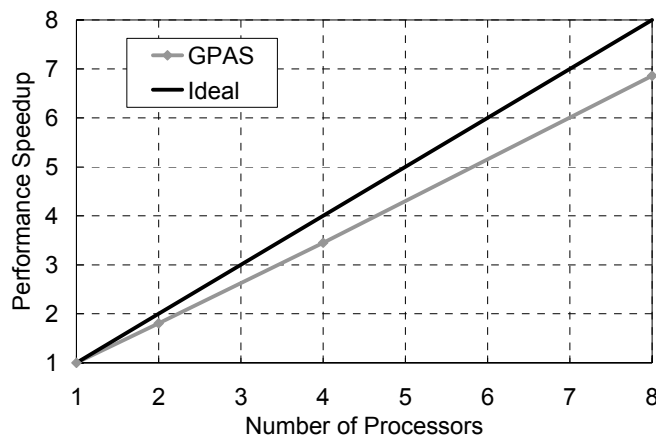
**Figure 3.32—Well locations for Case 3.**

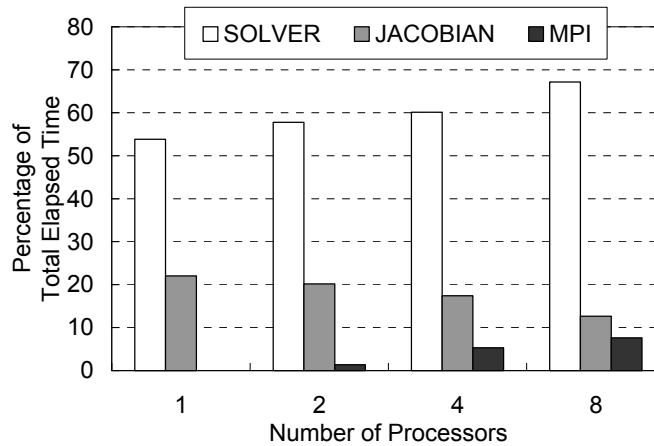
**Table 3.13—Input parameters for Case 3**

Property	Value
Reservoir size	2400ft x 2400ft x 96ft
Reservoir gridblock size	25ft x 25ft x 8ft
No. of gridblocks	96 x 96 x 12
Initial pressure	900 psi
Injection rates	1000 STB/day
Production pressure	500 psi
Number of injection wells	9
Number of production wells	4
Well pattern	Five spot pattern

Figure 3.33 shows the speedup of the run for 100 days of simulation time on two and four processors. The curve is normalized against the timing from the single processor run. The performance speedup is reduced with more processors because of the insufficient amount of computational task for each processor as the domains becomes smaller. Figure 3.34 shows the percentage of elapsed time used in the major computational tasks on different number of processors. Though the overall elapsed time decreases, the fraction of time spent in solver and MPI communication increases. The overall simulation time for the surfactant/polymer simulation was observed to be only 15% longer than a comparable waterflood simulation for the same case on a single processor.

A different indicator of parallel performance would be to keep the problem size per processor constant and increase the number of processors. Also, the use of layered or stochastic permeability fields affects the performance as reported in an earlier paper for the EOS compositional model. We plan to perform larger surfactant/polymer floods with stochastic permeability fields and with a larger number of gridblocks.

**Figure 3.33—Parallel performance speedup for Case 3.**



**Figure 3.34—Comparison of time spent in solver, Jacobian and MPI communication for Case 3.**

## **Chemical Module Tests—Fully Implicit Formulation**

### 1-D Surfactant/Polymer Flood

The first validation test was based on a 1-D surfactant/polymer flood simulation with 100 gridblocks in the x direction. The purpose of this simulation was to compare the formulation and the implementation of the fully implicit chemical module against both UTCHEM and the explicit method of GPAS. The simulation domain consisted of 400 ft in the x direction, 1 ft in the y direction and 1 ft in the z direction. An injector was located at the grid index (1,1,1) operating at a constant rate of 0.5 STB/d and a producer located at the (100,1,1) operating at a constant bottomhole pressure of 20 psi. Surfactant slug contained 1% surfactant and 500 ppm polymer with a salt concentration of about 0.17 meq/ml. The slug was injected for about 11.4 days followed by a polymer drive of about 500 ppm concentration. A conservative tracer was also injected during the chemical slug. The results are favorably compared with those of UTCHEM and GPAS explicit formulation in Figs. 3.35 through 3.40.

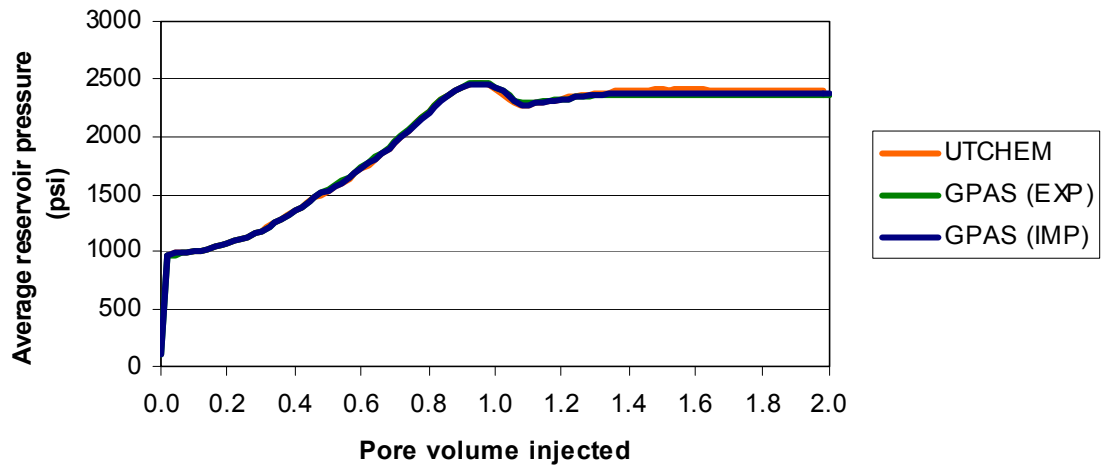


Figure 3.35—Comparison of average reservoir pressure for 1-D surfactant/polymer case.

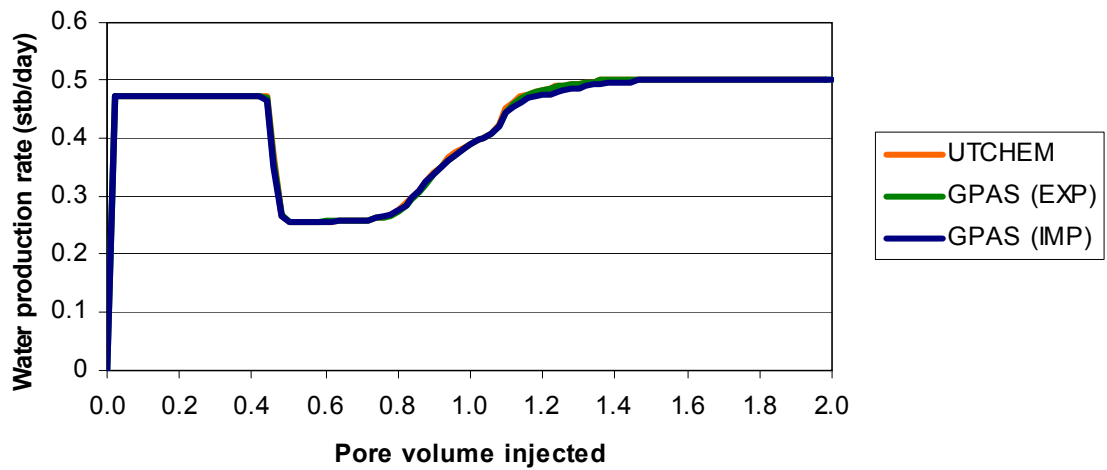


Figure 3.36—Comparison of water production rate for 1-D surfactant/polymer case.

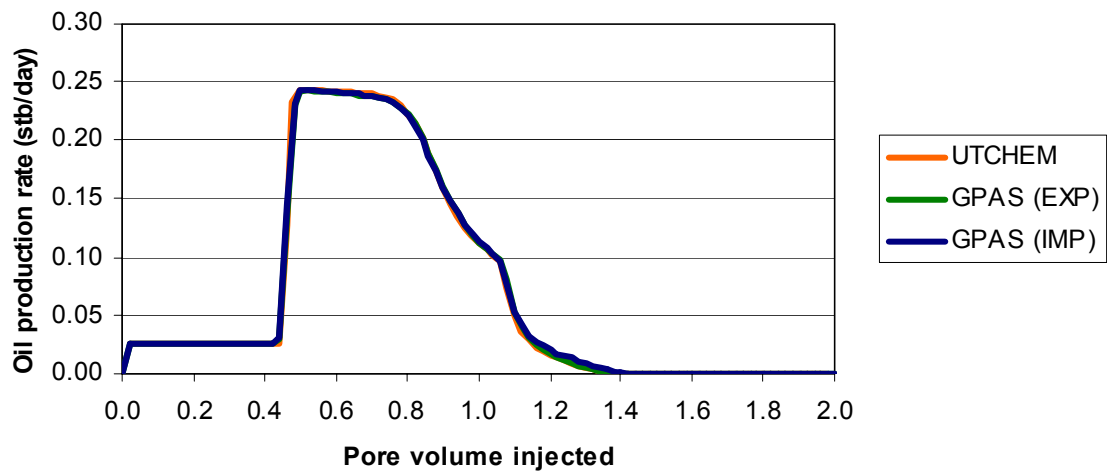
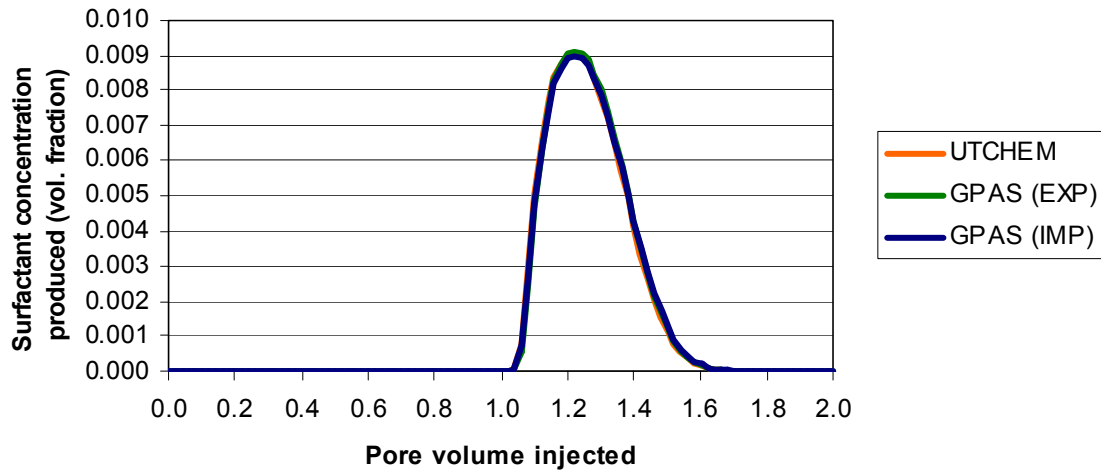
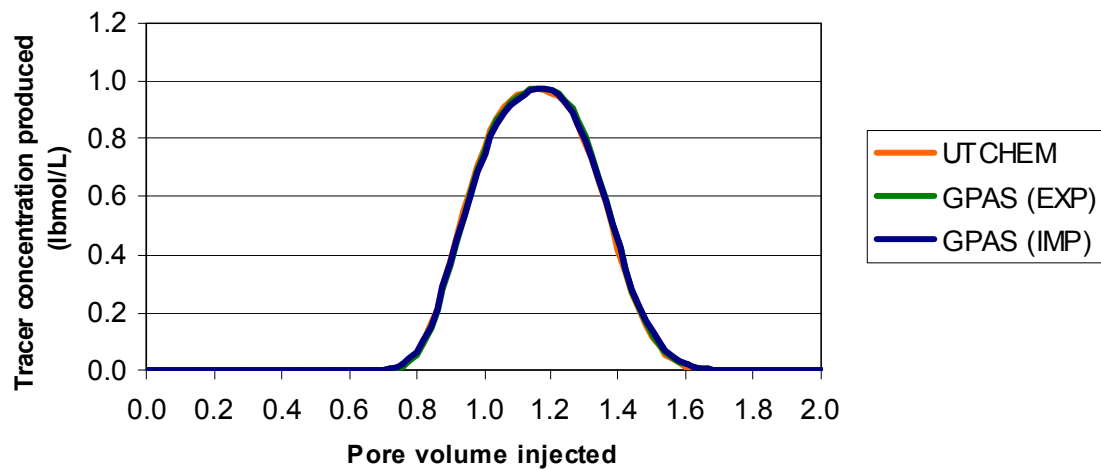


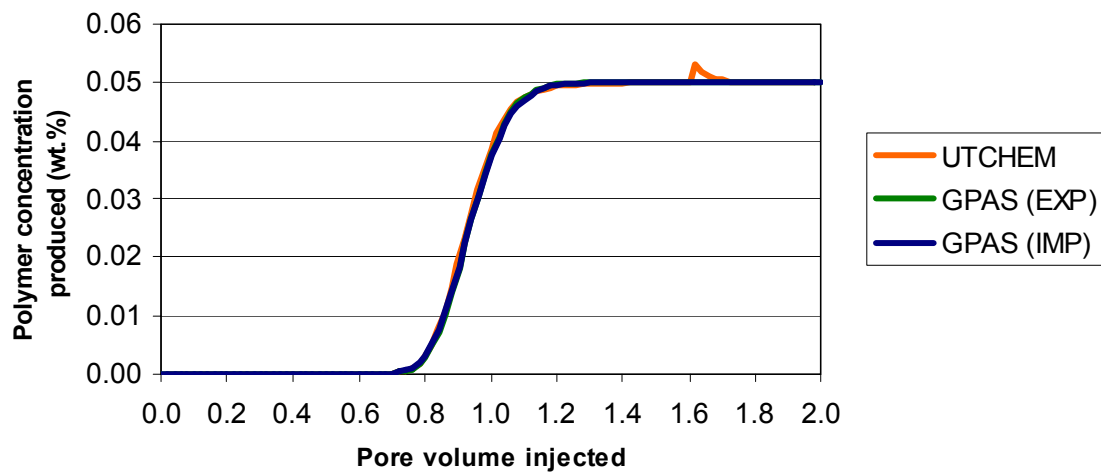
Figure 3.37—Comparison of oil production rate for 1-D surfactant/polymer case.



**Figure 3.38—Comparison of surfactant concentration produced for 1-D surfactant/polymer case.**



**Figure 3.39—Comparison of tracer concentration for 1-D surfactant/polymer case.**

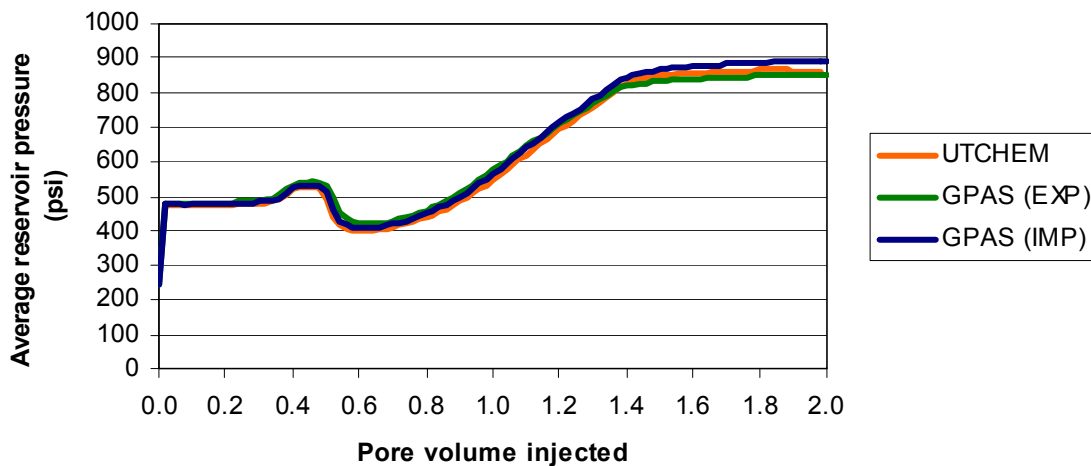


**Figure 3.40—Comparison of polymer concentration for 1-D surfactant/polymer case.**

## 2-D Surfactant/Polymer Flood

A two-dimensional homogeneous test case was setup to model surfactant-polymer slug injection in a quarter of five-spot well pattern using the fully implicit formulation. The input data used is summarized in Table 3.11. The reservoir is 480 ft in each direction modeled by a 15x15 areal grid. The permeability is isotropic and 540 md. A porosity of 0.136 with an initial oil saturation of 0.25 was used. The rock properties and relative permeability input are set for typical water wet conditions. The chemical slug consisted of 5% surfactant, 500 ppm polymer in a water at the salinity of 0.17 meq/ml. The polymer viscosity parameters resulted in viscosity of about 12 cp at the maximum polymer concentration used in the simulation (0.05 wt%). The chemical slug was injected for 104.5 days followed by a polymer drive with a concentration of 500 ppm. The oil phase was modeled using single component n-decane and the resulting phase properties. The injection rate was 534 STB/d and the producer was operating under a constant pressure of 200 psi.

The results are favorably compared with UTCHEM and GPAS explicit module in Figs. 3.41 through 3.46. The surfactant, tracer, and polymer concentration histories agree very well (Figs. 3.44 through 3.46).



**Figure 3.41—Comparison of average reservoir pressure for 2-D surfactant/polymer case.**

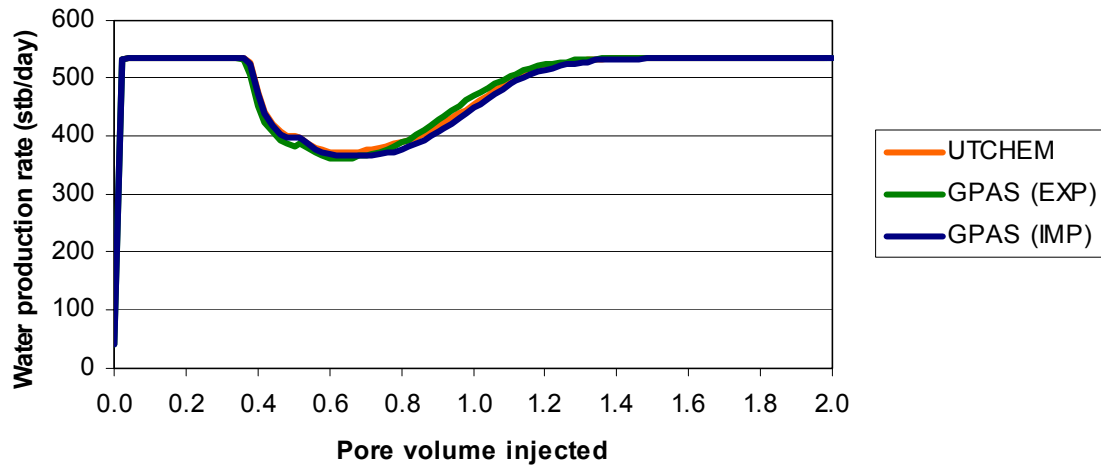


Figure 3.42—Comparison of water production rate for 2-D surfactant/polymer case.

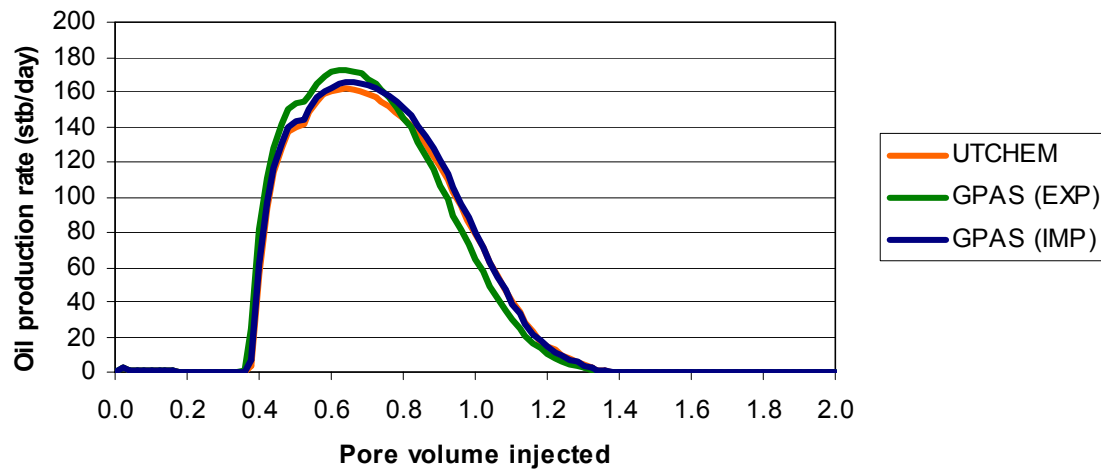


Figure 3.43—Comparison of oil production rate for 2-D surfactant/polymer case.

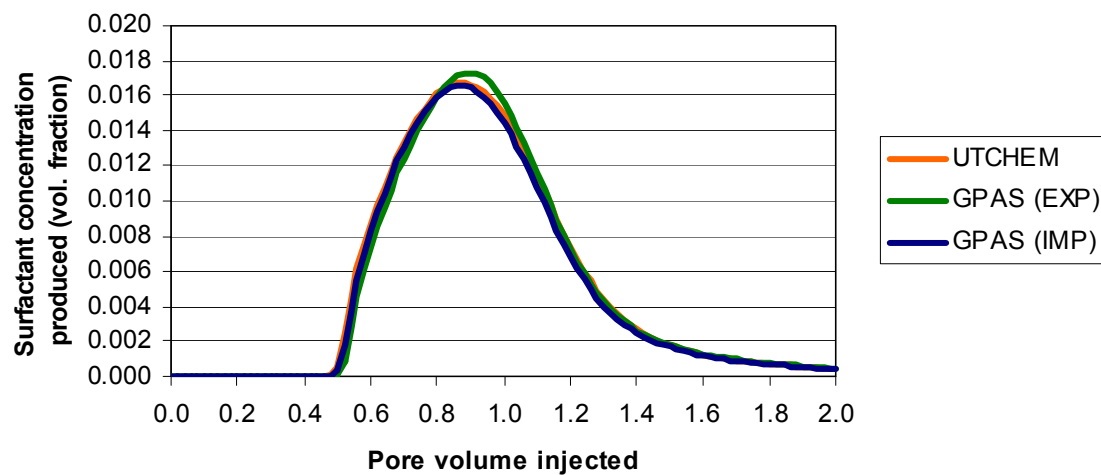
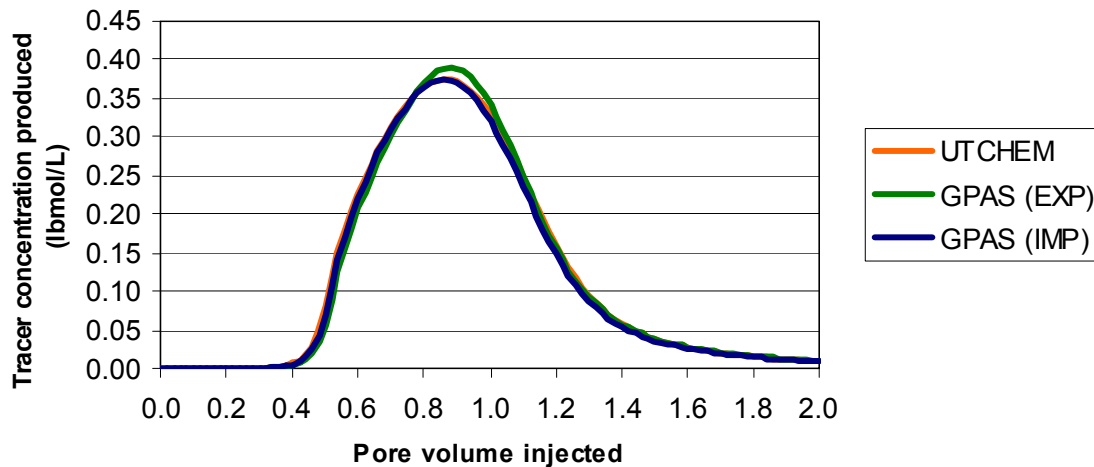
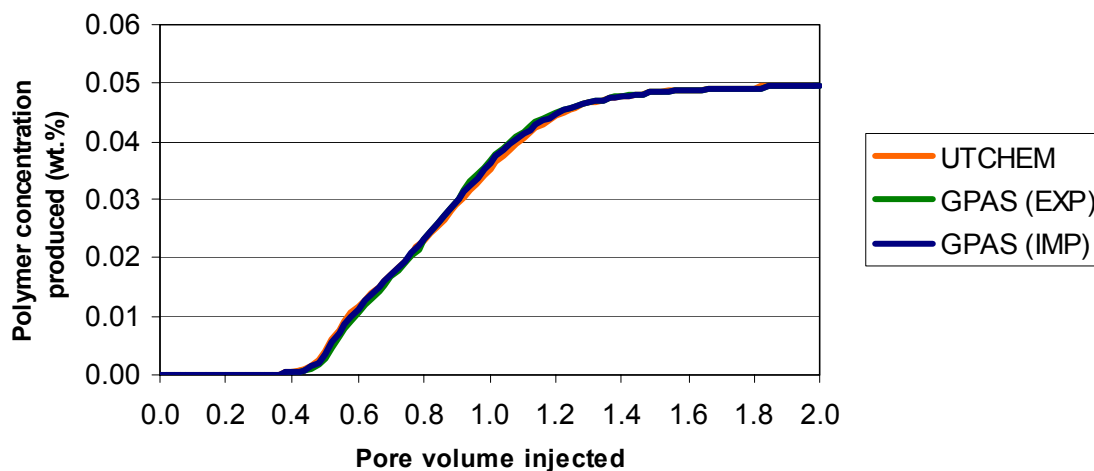


Figure 3.44—Comparison of surfactant concentration for 2-D surfactant/polymer case.





**Figure 3.45—Comparison of tracer concentration for 2-D surfactant/polymer case.**



**Figure 3.46—Comparison of polymer concentration for 2-D surfactant/polymer case.**

### 3-D Surfactant/Polymer Flood

The next validation of the fully implicit chemical module was based on a three-dimensional homogeneous surfactant/polymer simulation. The reservoir is 400ft in x and y direction with thickness of 4 ft modeled with 10x10x4 gridblocks. The permeability is isotropic and 500 md. A porosity of 0.20 with an initial oil saturation of 0.3 was used. A constant salinity surfactant/polymer flood was performed. The chemical slug consisted of 1% surfactant and 500 ppm polymer at a salinity of 0.17 meq/ml. The chemicals were injected for a period of 64 days followed by polymer drive with a concentration of 500 ppm. The injection rate was 142.475 STB/day and the production well was under a constant pressure of 20 psi.

Historical results of average reservoir pressure, water production rate, oil production rate, produced surfactant concentration, produced tracer concentration, produced polymer

concentration are given in Figs. 3.47 through 3.52. The results of UTCHEM, GPAS explicit, and GPAS implicit are not differentiable for this case.

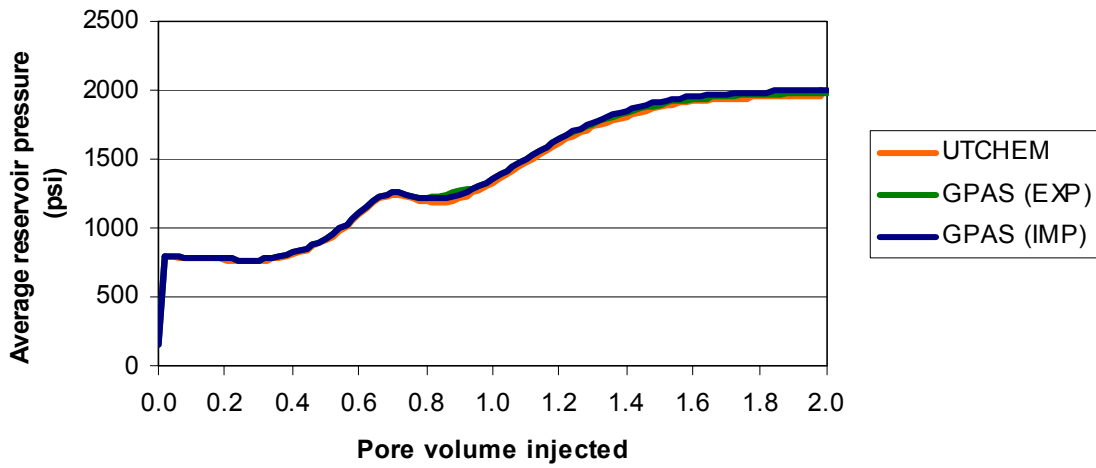


Figure 3.47—Comparison of average reservoir pressure for 3-D surfactant/polymer case.

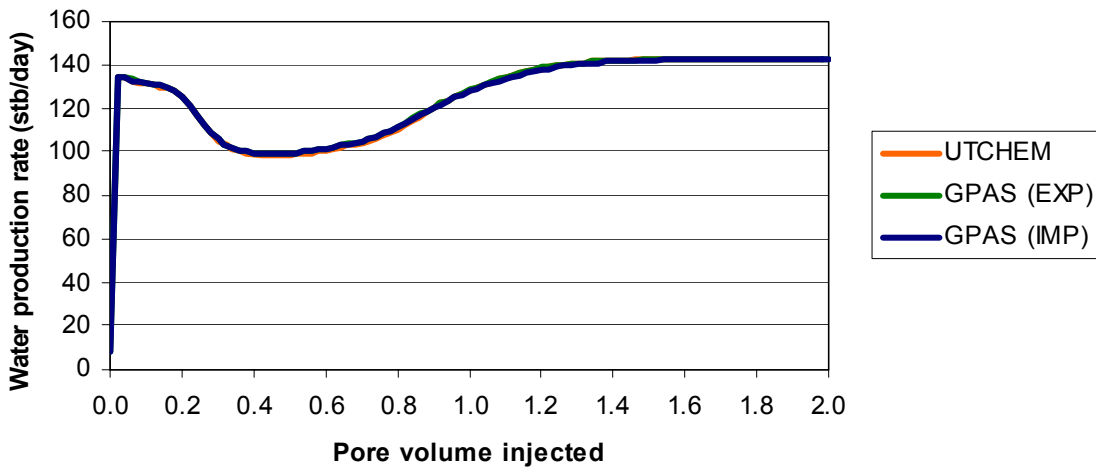
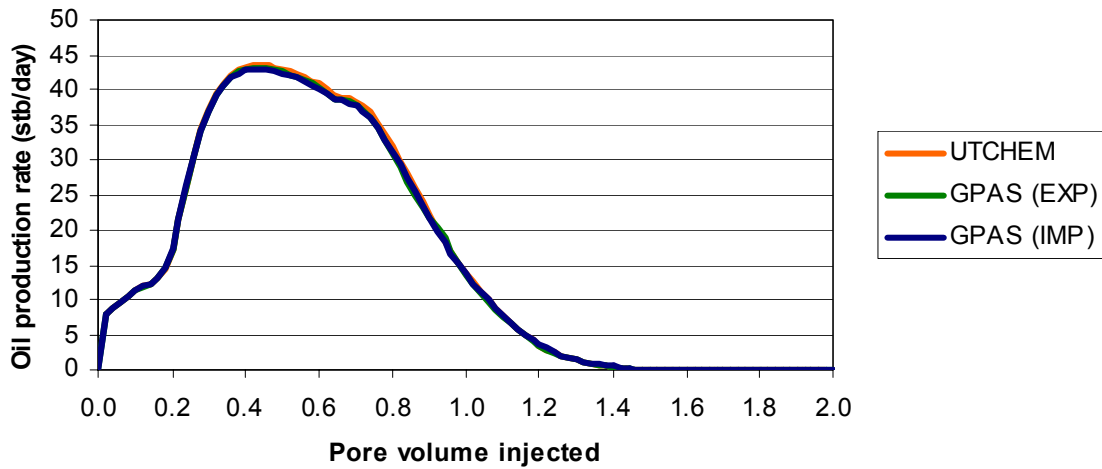
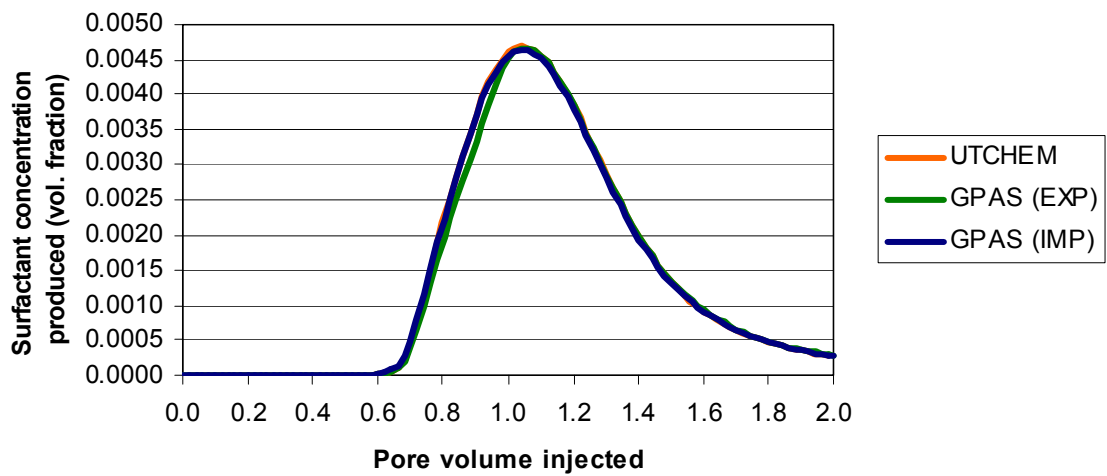


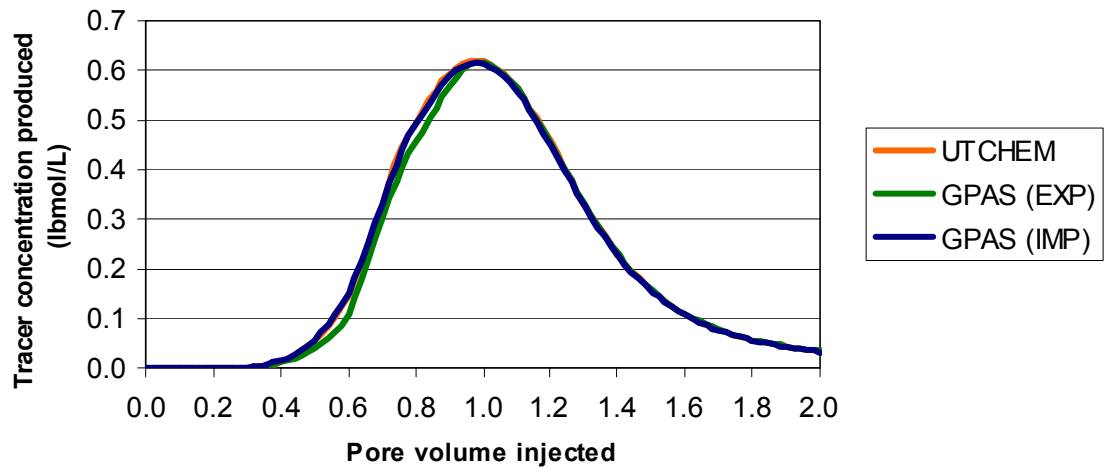
Figure 3.48—Comparison of water production rate for 3-D surfactant/polymer case.



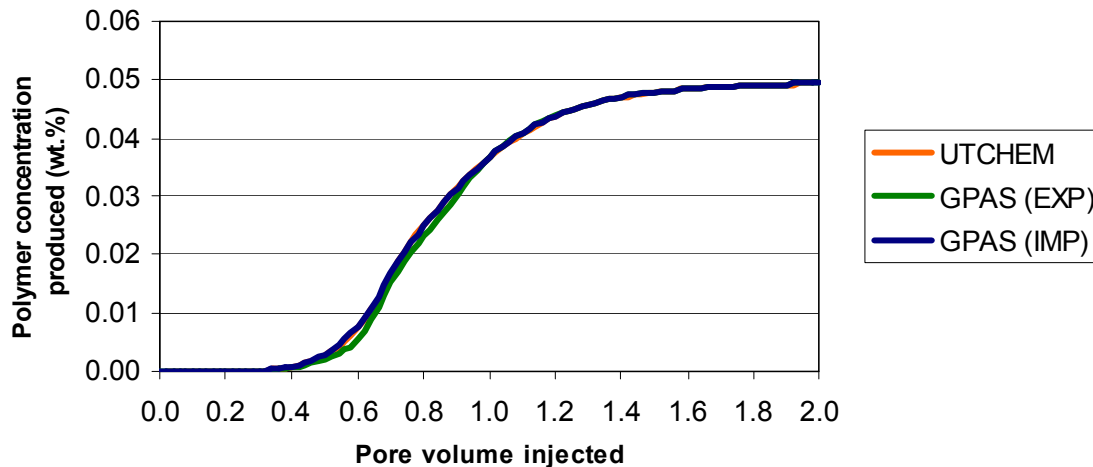
**Figure 3.49—Comparison of oil production rate for 3-D surfactant/polymer case.**



**Figure 3.50—Comparison of surfactant concentration for 3-D surfactant/polymer case.**



**Figure 3.51—Comparison of tracer concentration for 3-D surfactant/polymer case.**



**Figure 3.52—Comparison of polymer concentration for 3-D surfactant/polymer case.**

## Parallel Simulations

We have performed several simulations on Intel Xeon processors to study the scalability of GPAS.

### Benchmark Simulations

We conducted a series of benchmarks by running GPAS on an Intel Xeon-based Linux cluster and studied the scalability while using different interconnects for the cluster.

Our testing environment is based on a cluster consisting of 64 Dell PowerEdge 2650 servers interconnected with Fast Ethernet, Gigabit Ethernet, and Myrinet. Each PowerEdge 2650 has two Intel Xeon processors running at 2.4 GHz with 512KB L2 cache, 2GB of DDR-RAM (double data rate RAM) memory operating on a 400 MHz Front Side Bus. The chipset of PowerEdge 2650 is the ServerWorks GC-LE, which accommodates up to six registered DDR 200 (PC1600) DIMMs with a 2-way interleaved memory architecture. Each of the two PCI-X controllers on the 2650 has its own dedicated 1.6 GB/s full duplex connection to the North Bridge to accommodate the peak traffic generated by the PCI-X busses it controls.

The operating system installed for the cluster is RedHat Linux 7.3 with kernel version 2.4.18-4smp. The GPAS is compiled with Portland Group C/C++ and FORTRAN 77/90 compilers, and the PETSc library.

### *Simulation Model*

In this example, the reservoir description is layered with permeability different in each direction, and permeability in z-direction is 10 times smaller than the permeability in x-direction. In addition, the size of the reservoir simulated is 7.3 miles in x direction, 24.25 miles in y direction and 500 feet in z direction. The reservoir domain is divided into

197,120 blocks with 77x256x10 scheme. This leads to 1.5 million unknowns solved at each time-step simultaneously. There are 34 production wells and 54 injection wells totaling 88 wells scattered across the reservoir in a staggered line drive pattern. 100 days of gas injection is simulated. To run the case, the GPAS requires a total of 1.7GB memory.

### ***Performance Metric***

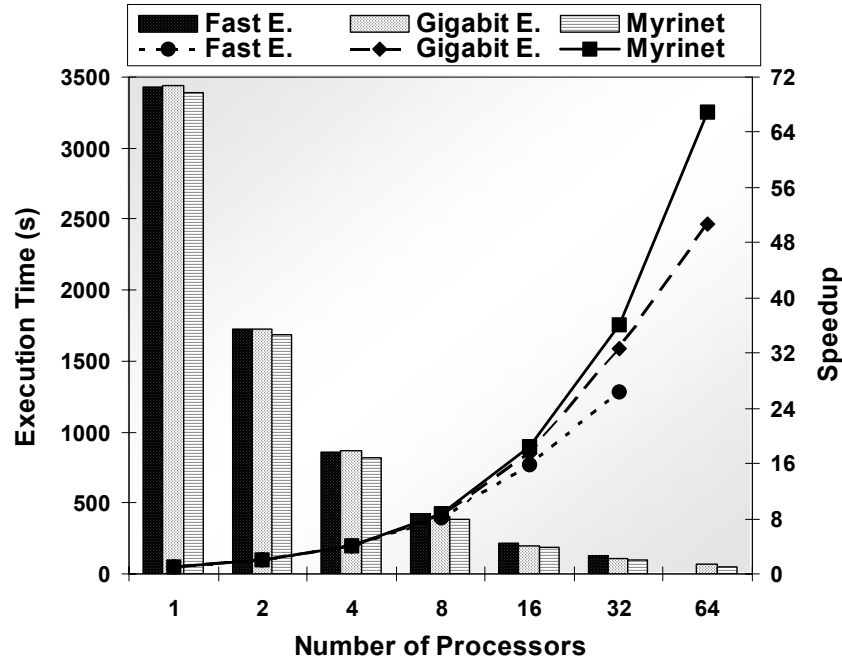
The efficiency in parallel applications is usually measured by speedup. In this study, the term “*speedup*” of a cluster with  $N$  processors is defined as

$$\text{Speedup} = \frac{t_1}{t_N}$$

where  $t_1$  is the amount of execution time by running on one processor and  $t_N$  is the amount of execution time spent with  $N$  processors. The ideal speedup of parallel simulation with  $N$  processors is  $N$ , that is, the program runs  $N$  times faster. However, in reality, as the number of processors becomes larger, we usually observe a speedup less than  $N$ . The performance reduction is due to increasing inter-processor communication or the memory contention arising from a cluster whose nodes are Symmetric Multi-Processors (SMP) machines. Sometimes, it can be due to an unfavorable programming style, in which the inefficient program does not decompose the application evenly. These are known as the overheads that are not encountered if there is only one processor.

### ***Results and Analysis***

We have configured three different interconnects, Fast Ethernet, Gigabit Ethernet, and Myrinet for our study. For this part of the study, only one processor per compute node is used. Figure 3.53 shows the execution times (bars) and speedups (lines) of the simulation case from one processor to 64 processors. The left y axis in the figure is the execution time in terms of seconds and the right y axis is the performance speedup of the GPAS. The GPAS has the best performance as well as the scalability (speedup) on the cluster with Myrinet. In fact, the speedup at 16 processors is almost 20, a super linear speedup. The performance difference becomes significant only after 16 processors.

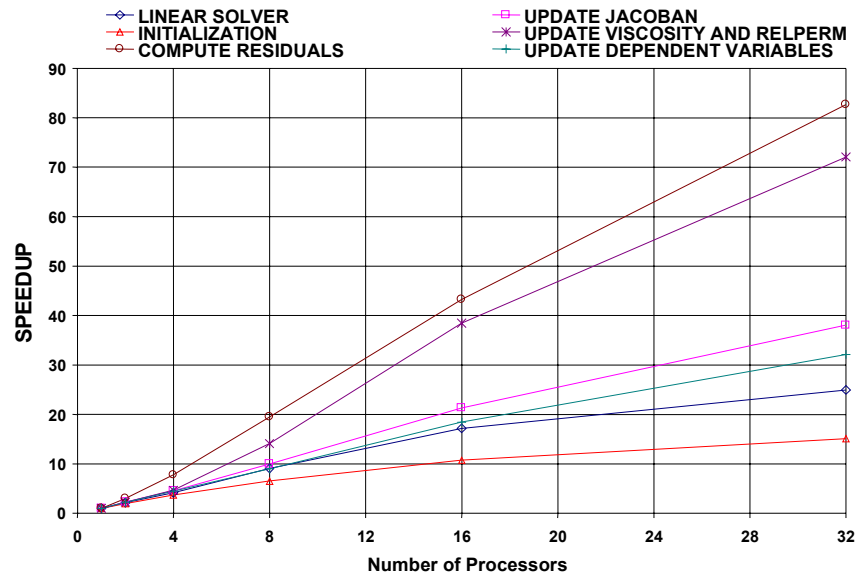


**Figure 3.53—GPAS execution time and speedup plots for the case of 77x256x10 using single processor per node.**

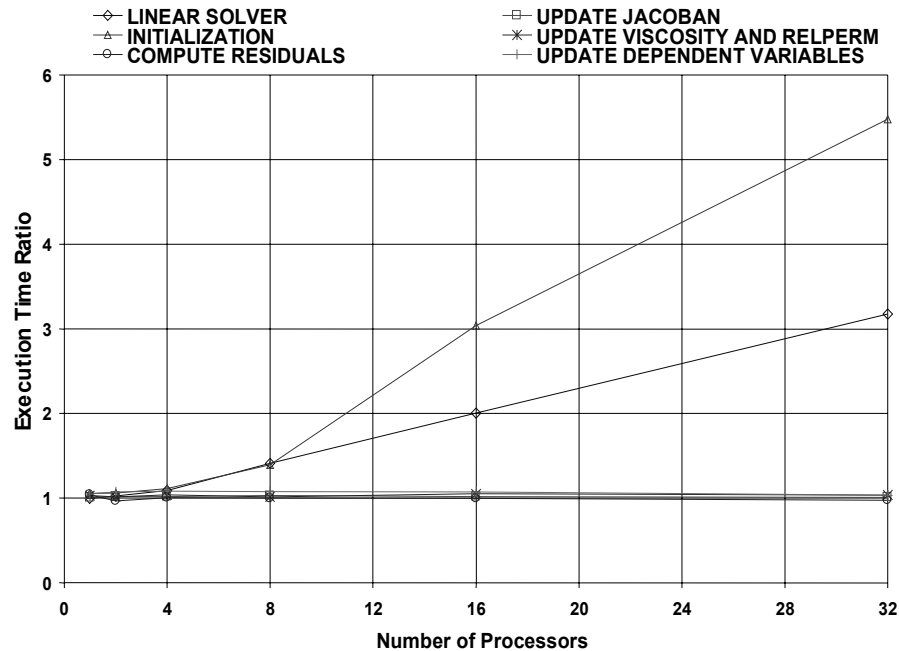
The simulation was constituted of several sections such as I/O, Initialization, Linear Solver, Viscosity and Relative Permeability, Residuals, Ghost Region, Jacobian, and Dependent Variables. These sections are timed individually during the simulation and reported at the end. To identify which section of the GPAS is affected by which part of the cluster configuration, we use the time spent on each section at a single processor as the base to generate a speedup curve for each section of the code. Figure 3.54 shows the speedup of each GPAS's section for the large case (77x256x10) on the Myrinet cluster. The plot shows that each section of the code has its own speedup rate from one processor to 32 processors. The speedups of "Compute Residuals" and the "Update Viscosity and Relperm" have a slope 2.5 that indicates these sections benefit considerably from the aggregation of the memory bandwidth and cache sizes. The "Update Dependent Variables" and the "Update Jacobian" sections have the speedup rate around 1.0. That indicates the aggregated memory and cache effects on these sections are offset by the communication overheads of having more processors. Finally, the speedup rates for the two sections, "Total Linear Solver" and the "Initialization" are less than 1.0. That is because these sections are communication sensitive and the performances are affected more by the interconnect capacity than others.

Furthermore, we calculate the ratio of the time spent in each section for Fast Ethernet to that of Myrinet, and plot the results in Fig. 3.55. The ratios clearly show that "Linear Solver" and the "Initialization" are the most communication dependent sections and they are the only two sections show relatively large difference from Fast Ethernet configuration to Myrinet configuration. On the other hand, all other sections have ratio of 1, which indicates the communication that required by those sections is minimal.

Therefore, the differences between the two interconnect performance is not shown in those sections.



**Figure 3.54—Speedup curves of each section for the simulation case (77x256x10) using Myrinet as the interconnect.**

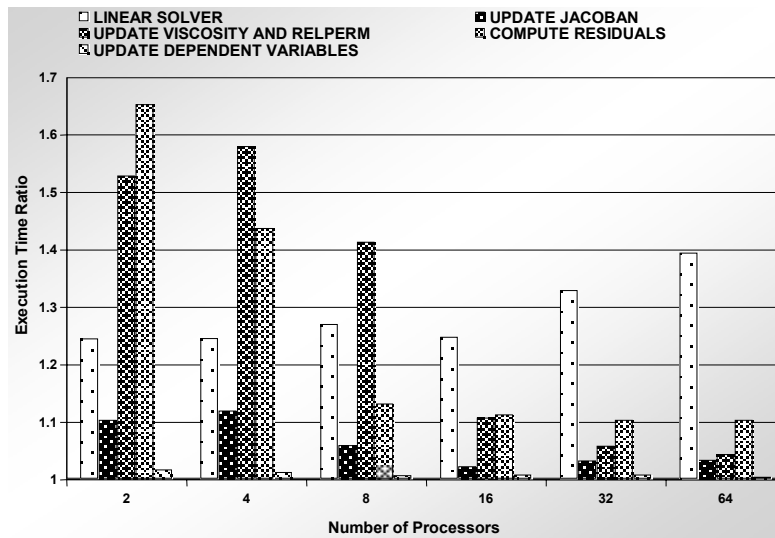


**Figure 3.55—Execution time ratio of Fast Ethernet to Myrinet for each sections of the larger simulation case (77x256x10).**

### Single-Processor vs. Dual-Processor

We continued our study in comparing the performances of GPAS for single-processor and dual-processor per node configurations. Two processes running in a system will

compete for the memory resource. Particularly, the shared memory bus will be the performance bottleneck, when the memory is accessed at the same time by the two processes. In addition, the communication traffics generated by the two processes could create another potential bottleneck on the I/O resource, such as the PCI bus or the network interface card. Based on the general knowledge, we calculate the ratio of the time spent in each section for single-processor to that of dual-processor runs performed on the Myrinet cluster, and plot them in Fig. 3.56. Almost all the ratios are decreasing as the number of processor increases. That shows the memory contention problem becomes less as more processors are used. The reason is the data set per processor getting smaller, which reduces the memory contention relatively. On the other hand, with the fact that as the processor count increases, the communication among processes also increases, for the communication sensitive sections such as the “Total Linear Solver Time”, the ratio is not just considerably large, but also increases with the processor count.

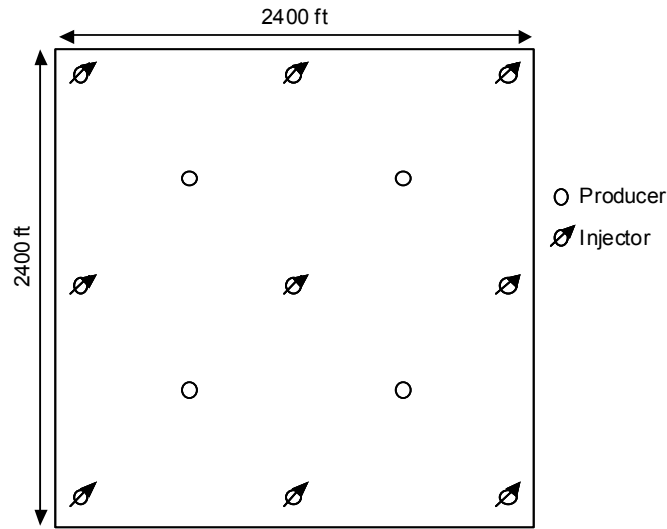


**Figure 3.56—Execution time ratio of single-processor to dual-processor for each sections of the simulation case (77x256x10) on the cluster with Myrinet.**

### 3-D Surfactant/Polymer Flood

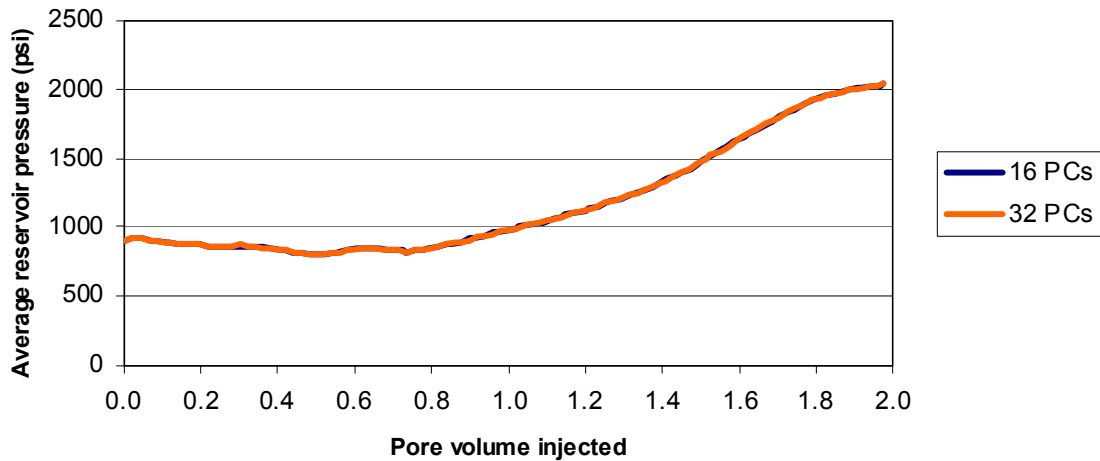
A 3-D surfactant/polymer flood simulation was set up with 96x96x11 gridblocks (101,376 gridblocks) and ran on both 16 and 32 processors. The gridblocks are 25 ft by 25 ft by 6 ft. The permeability was homogeneous and isotropic at about 540 md with a uniform porosity of 0.336. The initial oil saturation was at a residual value of 0.3. Several five spot patterns were chosen with injectors injecting at a constant rate of 1000 STB/d and the production wells at a constant pressure of 500 psi. The well pattern consisted of 9 injectors and 4 producers as given in Fig. 3.57. The 500-day chemical slug consisted of 5% surfactant, 500 ppm polymer at a salinity of 0.17 meq/ml. The slug followed by a 500 ppm polymer drive.



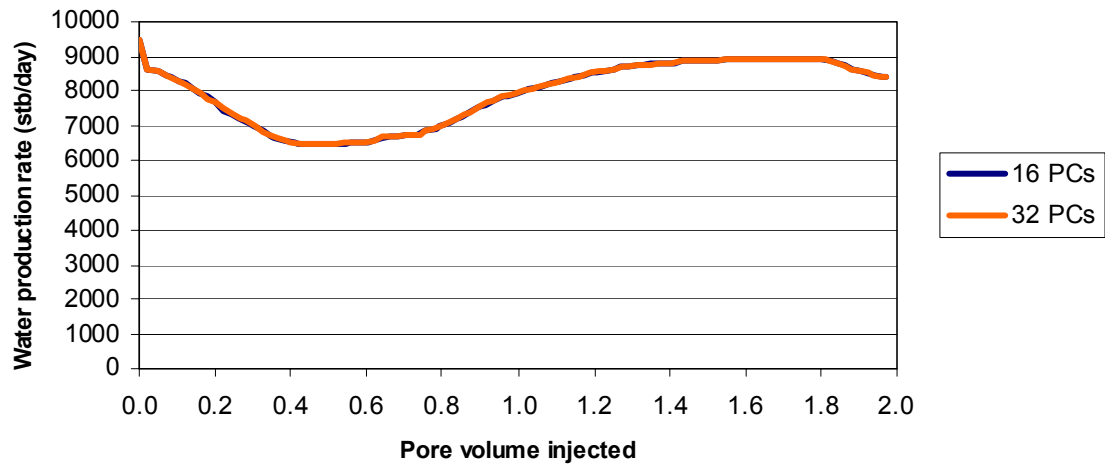


**Figure 3.57—Well locations for Case 3.**

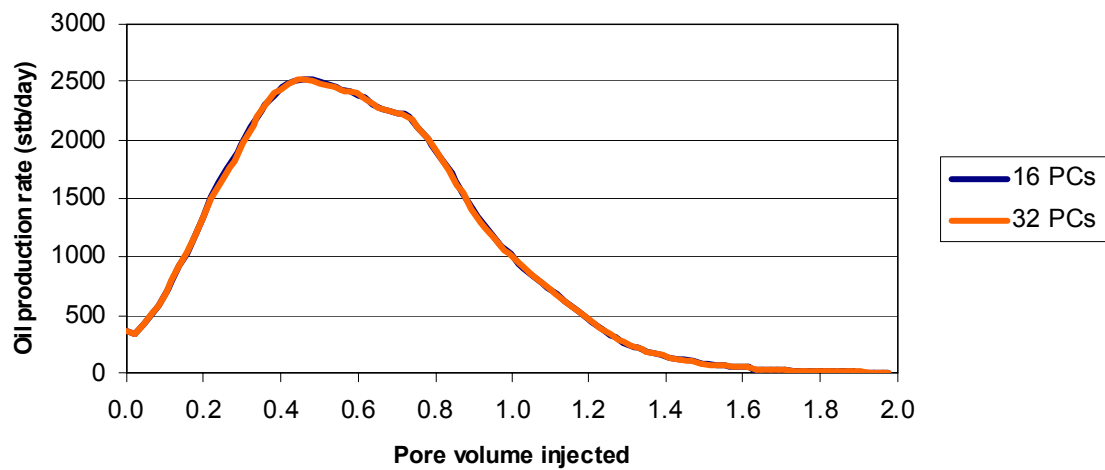
The simulation results were the same despite the number of processors used (Figs. 3.58 through 3.61). With the automatic time step option, the total CPU time for 2 pore volumes of injection was 17.8 hrs and 10.2 hrs for 16 and 32 processors, respectively. Figure 3.62 shows the time steps to demonstrate the advantage of large time steps using the fully implicit chemical module. Time steps are initially small during the chemical injection but it increases as high as 30 days after about 500 days of simulation. The number of Newton iterations is given in Fig. 3.63. The number of iterations is on the order of 10 with a peak value of about 50 at about 1500 days of injection. A copy of the input file is given in Appendix D.



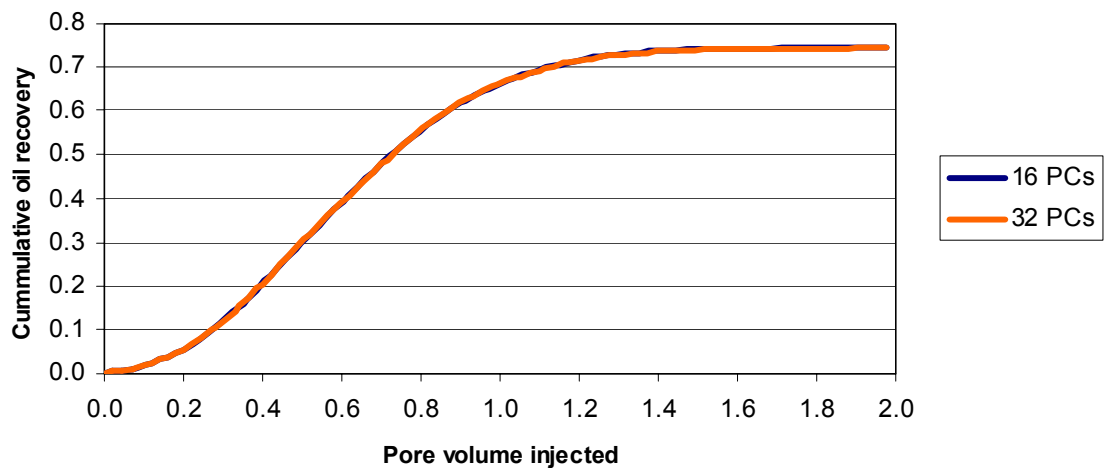
**Figure 3.58—Average reservoir pressure for the parallel surfactant/polymer flood simulation (96x96x11).**



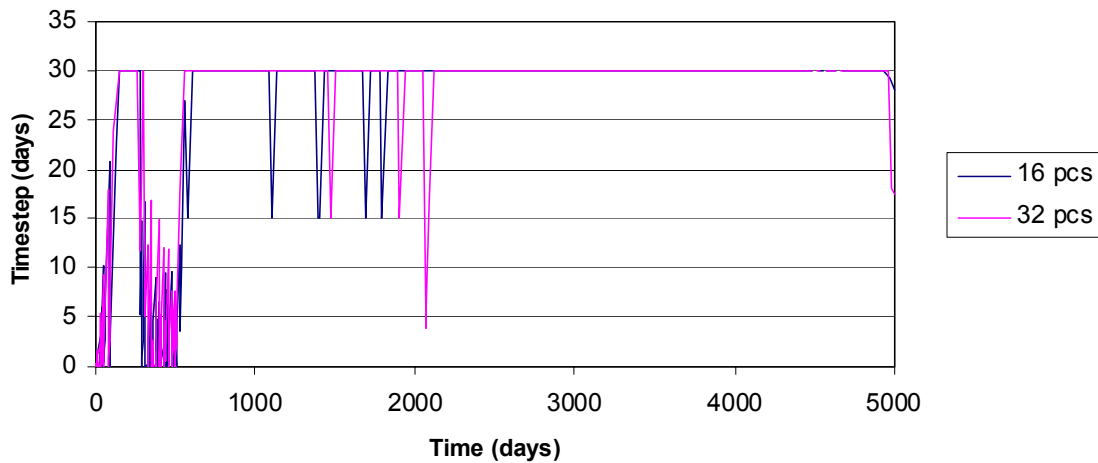
**Figure 3.59—Water production rate for the parallel surfactant/polymer flood simulation (96x96x11).**



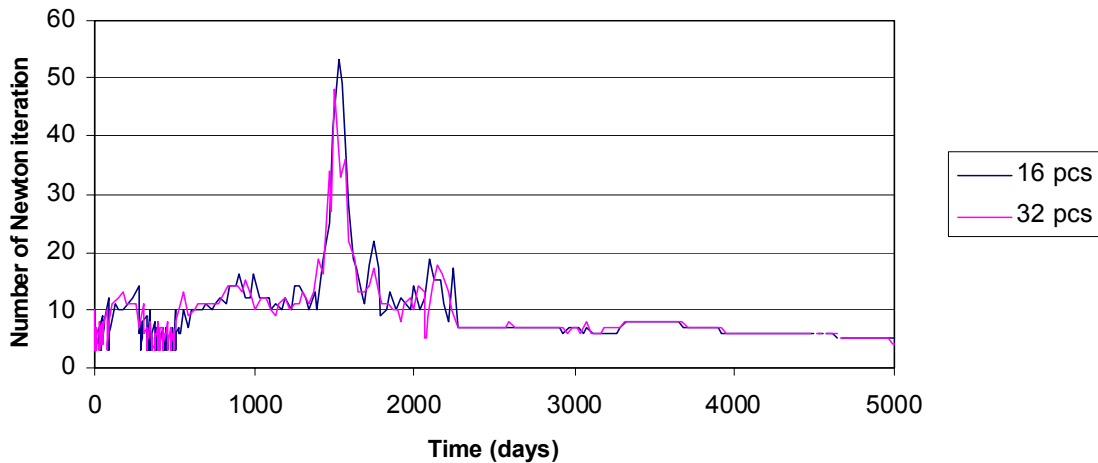
**Figure 3.60—Oil production rate for the parallel surfactant/polymer flood simulation (96x96x11).**



**Figure 3.61—Cumulative oil recovery for the parallel surfactant/polymer flood simulation (96x96x11).**



**Figure 3.62—Time step for the parallel surfactant/polymer flood simulation (96x96x11).**



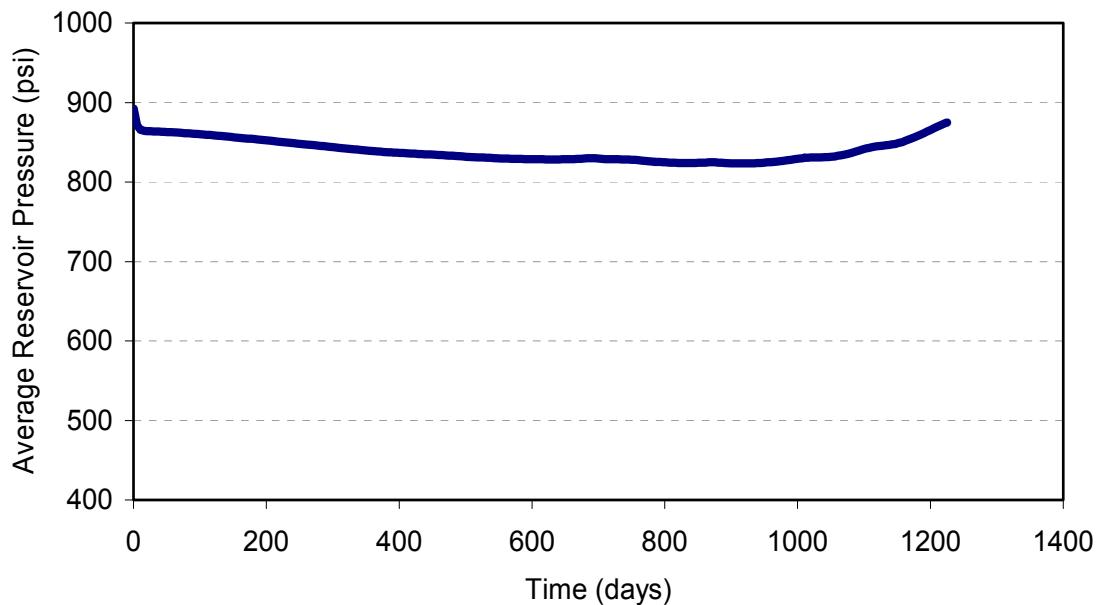
**Figure 3.63—Number of Newton iteration for the parallel surfactant/polymer flood simulation (96x96x11).**

### A Million-Gridblock 3-D Surfactant/Polymer Flood

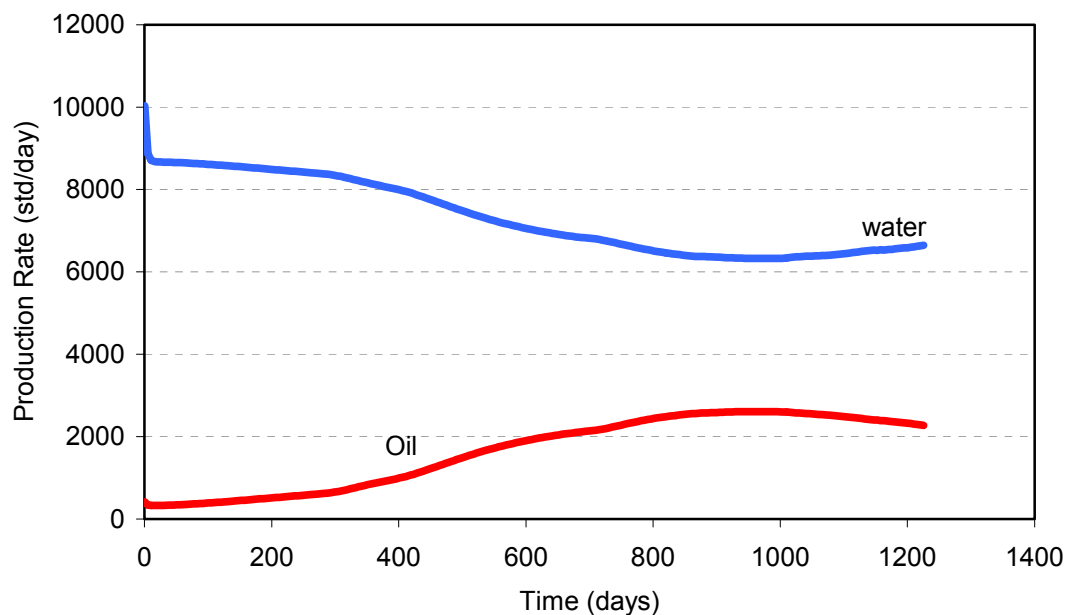
A 3-D surfactant/polymer flood simulation was set up with over a million gridblock (384x384x7 gridblocks) and ran on 128 processors. The gridblocks are 6 ft by 6 ft by 9 ft. The simulation area is about 122 acres with multiple five spot well patterns. The permeability was homogeneous and isotropic at about 540 md with uniform porosity of 0.336. Water viscosity was 0.75 cp whereas that of the oil phase was 2.6 cp. The initial oil saturation was 0.3 with the waterflood residual saturation of 0.25. Several five spot patterns were chosen with injectors injecting at a constant rate of 1000 STB/d and the production wells at a constant pressure of 500 psi. The well pattern consisted of 9 injectors and 4 producers as given in Fig. 3.57. The 125-day chemical slug consisted of 5% surfactant, 500 ppm polymer at a salinity of 0.17 meq/ml. The slug followed by a 500 ppm polymer drive with the same salinity of 0.17 meq/ml. As the previous simulation, there were 13 wells with 9 injection wells and 4 production wells. The

injectors were operated at a constant rate of 1000 STB/d and the production wells were producing at a constant pressure of 500 psi. The simulation was run for 1200 days.

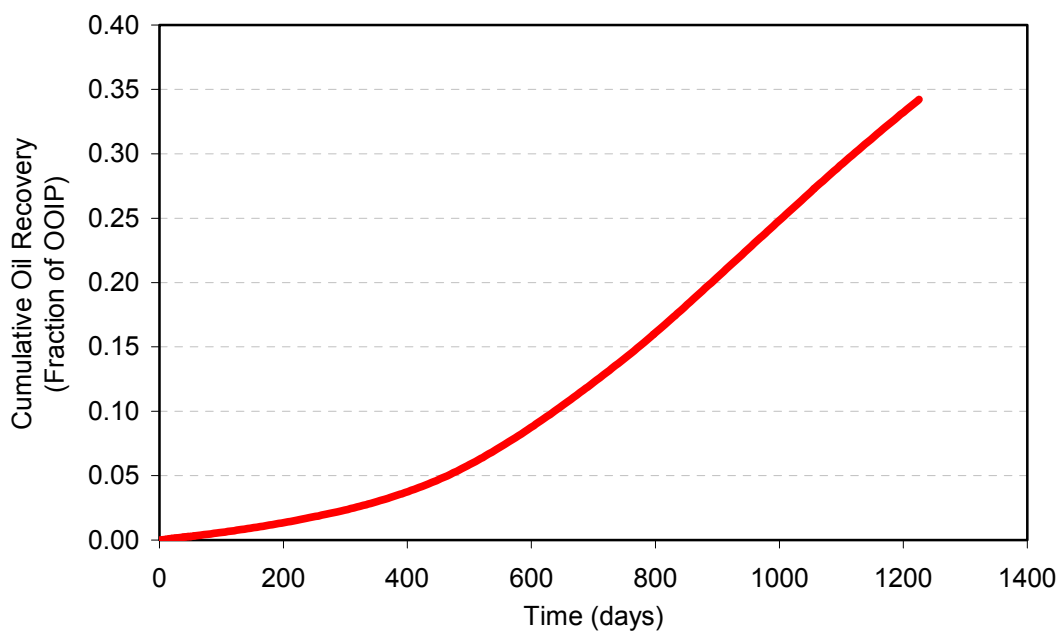
The simulation results are given in Figs. 3.64 through 3.66. Figure 3.67 shows the time steps to demonstrate the advantage of large time steps using the fully implicit chemical module. The automatic time step option was utilized in this simulation where the minimum time step was 0.5 days and the maximum time step was 5 days. Time steps are initially small during the chemical injection but it increases as high as 5 days. The number of Newton iterations is given in Fig. 3.68. The number of iterations is on the order of 10 with a peak value of about 8 at about 1150 days of injection.



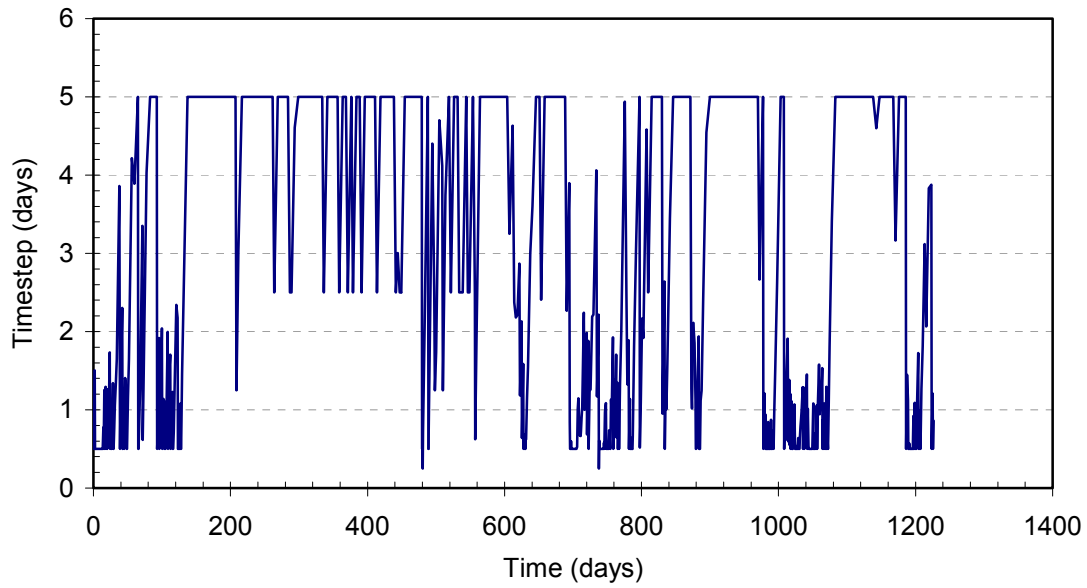
**Figure 3.64—Average reservoir pressure for the parallel surfactant/polymer flood simulation (384x384x7).**



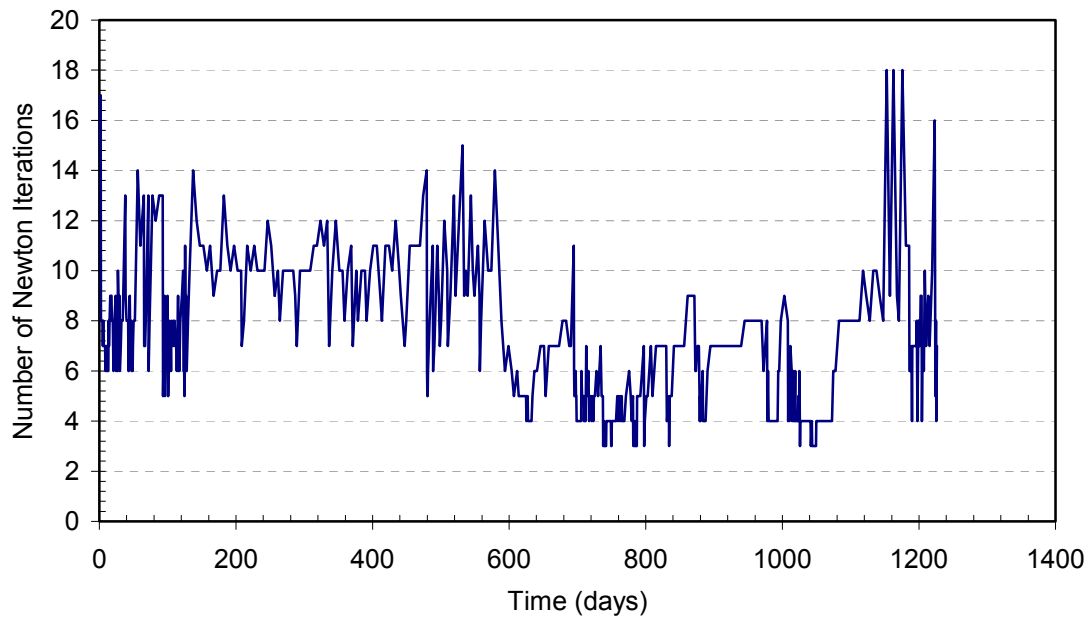
**Figure 3.65—Total water and oil production rates for the parallel surfactant/polymer flood simulation (384x384x7).**



**Figure 3.66—Cumulative oil recovery for the parallel surfactant/polymer flood simulation (384x384x7).**



**Figure 3.67—Time step values for the parallel surfactant/polymer flood simulation (384x384x7).**



**Figure 3.68—Number of Newton iterations for the parallel surfactant/polymer flood simulation (384x384x7).**

## CONCLUSIONS

This is our final report of the project and we gave our progress on Tasks 1 through 3. We have formulated the mass conservation equations and physical properties for aqueous species such as tracers, polymer, surfactant, and electrolytes. We formulated and implemented the chemical module first as a hybrid method and then fully implicit.

Several new well operating conditions were added to facilitate the chemical flooding simulations. A new automatic time step control was added for better numerical solution stability and efficiency. The new implementation is based on the maximum change in pressure, phase saturation, and molar concentration of each species. Several solvers were also implemented and tested for robustness and efficiency. The default solver package is PETSc developed at Argonne National Laboratory.

The fully implicit, equation of state, compositional model of GPAS was extended to model chemical oil recovery processes. First, with the hybrid approach, the aqueous species mass balance and associated surfactant/oil/water phase behavior are calculated explicitly, but their influence on multiphase flow is modeled using a fully implicit trapping number formulation. The simulation results were validated by comparison with results from the explicit chemical flooding simulator, UTCHEM. Test runs were performed with more than 100,000 gridblocks in a parallel environment with results indicating a good scalability of the simulator. The approach described in this research is modular, easy to implement and provides reasonably accurate results. This same approach could be implemented in other implicit, compositional EOS simulators to add aqueous chemistry effects to them, which would greatly extend their range of applications.

We, then, formulated and implemented the fully implicit chemical module. This required the derivative of the equations and development of a new Jacobian. The physical property calculations were the same as those in the explicit formulation. The advantages over the hybrid approach are the larger time step and more general surfactant formulation. Several simulations were performed and the results were compared with the results of both UTCHEM and explicit GPAS.

We have formulated, implemented, and performed a series of validation test for more flexible corner point grid geometry in GPAS.

We have also conducted a series of benchmarks by running the General Purpose Adaptive (GPAS) simulator on a Linux cluster and studied the scalability while using different interconnects. The results were very encouraging and indicated that GPAS performance scales linearly from one to 64 single processor nodes using a low latency, high-bandwidth such as Myrinet.

The biggest accomplishment of this project was the capability of successfully making field-scale simulations of surfactant/polymer flood processes with large number of

gridblocks on the order of one million with a fully implicit formulation and parallel capability on a cluster of PCs



## REFERENCES

- Abate, J.M., P. Wang and K. Sepehrnoori, "Parallel Compositional Reservoir Simulation on a Cluster of PCs," International Journal of High Performance Computing Applications, Spring 2001.
- Aziz, K., Ramesh, B., and Woo, P.T.:" Fourth SPE Comparative Solution Project: A Comparison of Steam Injection Simulators", SPE 13510 presented at the Eighth SPE symposium on Reservoir Simulation, Dallas, TX, 1985.
- Baker, L.E., Pierce, A.C. and Luks, K.D.:" Gibbs Energy Analysis of Phase Equilibria," Society of Petroleum Engineers Journal, 22, No.5, 1982.
- Balay, S., Gropp, W., McInnes, L.C. and Smith, B.:" Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," Modern Software Tools in Scientific Computing, Argonne National Laboratory, October 1997.
- Camilleri, D., A. Fil, G. A. Pope, B. A. Rouse and K. Sepehrnoori. "Comparison of an Improved Compositional Micellar/Polymer Simulator With Laboratory Core Floods," *SPE*, Nov. 1987.
- Camilleri, D., A. Fil, G.A. Pope, B.A. Rouse and K. Sepehrnoori: "Comparison of an Improved Compositional Micellar/Polymer Simulator With Laboratory Core Floods," *SPE Reservoir Engineering*, Nov. 1987.
- Camilleri, D., Engelsen, S., Lake, L.W., Lin, E.C., Ohno, T., Pope, G.A. and Sepehrnoori, K.:" Description of an Improved Compositional Micellar/Polymer Simulator," *SPE Reservoir Engineering*, November 1987.
- Chang, Y.B.:" Development and Application of an Equation of State Compositional Simulator," Ph.D. Dissertation, 1990.
- Chatzis, I. and N. R. Morrow: "Correlation of Capillary Number Relationships for Sandstones," paper SPE 10114 presented at the 56th Annual Conference of the SPE, San Antonio, TX, Oct. 5-7, 1981.
- Delshad, M., Pope, G.A., and Sepehrnoori, K.:" A Compositional Simulator for Modeling Surfactant Enhanced Aquifer Remediation," *Journal of Contaminant Hydrology*, 23, 1996.
- Delshad, M., D. Bhuyan, G.A. Pope, and L.W. Lake: "Effect of Capillary Number of the Residual Saturation of a Three-Phase Micellar Solution," paper SPE 14911 presented at the SPE fifth symposium of Enhanced Oil Recovery, Tulsa, OK, April 1986.
- Delshad, M.: "Trapping of Micellar Fluids in Berea Sandstone," Ph.D. dissertation, The University of Texas at Austin, 1990.

- Delshad, M.: Trapping of Micellar Fluids in Berea Sandstone, Ph.D. Dissertation, The University of Texas at Austin, 1990.
- Dogru, A.H., H.A. Sunaidi, L.S. Fung, W.A. Habiballah, N. Al-Zamel, and K.G. Li: "A Parallel Reservoir Simulator for Large-Scale Reservoir Simulation," *SPEREE*, February 2002.
- Flory, P.J.: *Principles of Polymer Chemistry*, Ithaca, New York, Cornell University Press, 1953.
- Gai X., R.H. Dean, M.F. Wheeler and R. Liu: "Coupled Geomechanical and Reservoir Modeling on Parallel Computers," paper SPE 79700 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, U.S.A., 3-5 February 2003.
- Gosset, H. and Kalitventzeff.: "An Efficient Algorithm to Solve Cubic Equations of State," *Fluid Phase Equilibria*, 25, 1986.
- Gropp, W., Morgan, T., Smith, B., Arbogast, T., Dawson, C.N., Lake, L.L., McKinney, D.C., Pope, G.A., Sepehrnoori, K., and Wheeler, M.F.: *New Generation Framework For Petroleum Reservoir Simulation*, First Annual Report, May 31, 1996, Advanced Computational Technology Initiative, Argonne National Laboratory and The University of Texas at Austin.
- Habiballah, W.A., M.E. Hayder, M.S. Khan, K.M. Issa, S.H. Zahrani, R.A. Shaikh, A.H. Uwaiyedh, T.P. Tyraskis and M.A. Baddourah: "Parallel Reservoir Simulation Utilizing PC-Clusters in Massive Reservoir Simulation Models," paper SPE 84065 presented at the SPE Annual Technical Conference and Exhibition, Denver, Colorado, U.S.A., 5-8 October 2003.
- Hand, D.B.: "Dimeric Distribution: I. The Distribution of a Consolute Liquid between Two Immiscible Liquids," *Journal of Physics and Chemistry*, 34, 1939.
- Hirasaki, G. J. and G. A. Pope. "Analysis of Factors Influencing Mobility and Adsorption in the Flow of Polymer Solution Through Porous Media," *SPEJ*, 337-346, Aug. 1974.
- Hirasaki, G.J.: "Application of The Theory of Multicomponent, multiphase displacement to three-component, two-phase surfactant Flooding," *SPEJ*, 191-204, 1981.
- Huh, C.: "Interfacial Tension and Solubilization Ability of a Microemulsion Phase that Coexists with Oil and Brine," *Journal of Colloid Interface Science*, 71, 1979.
- Jin, M.: "A Study of Non-Aqueous Phase Liquid Characterization and Surfactant Remediation," PhD dissertation, University of Texas at Austin. 1995.
- John, A.K: " Development of a hybrid approach to couple chemical and compositional models in an implicit, parallel simulator" M.S. thesis, The University of Texas at Austin. 2003.

- Killough, J.E. and Kossack, C.A.: "Fifth Comparative Solution Project: Evaluation of Miscible Flood Simulators," SPE 16000, presented at the ninth SPE symposium on Reservoir Simulation held in San Antonio, Texas, February, 1987.
- Leibovici, C.F. and Neoschil, J.: "A New Look at the Rachford-Rice Equation for Flash Calculations," Fluid Phase Equilibria, 1992.
- Lohrenz, J., Bray, B.G. and Clark, C.R.: "Calculating Viscosities of Reservoir Fluids from their Compositions," Trans., AIME, 1964.
- Maliska, C.R., Computational Heat Transfer and Fluid Mechanics, 2<sup>nd</sup> Ed., LTC, Rio de Janeiro, 2004, Brazil. (In Portuguese).
- Michelsen, J.L.: "The Isothermal Flash Problem. Part I. Stability," Fluid Phase Equilibrium, 1982.
- Morrow, N.R., I. Chatzis and H. Lim: "Relative Permeabilities at Reduced Residual Saturation," *J. Can. Pet. Techol.*, 62-69, Jul-Aug., 1985.
- Nalla, G: "Implementation of the chemical flooding module in GPAS," M.S. thesis, The University of Texas at Austin. 2002.
- Nelson, R. C. and G. A. Pope. "Phase Relationships in Chemical Flooding," SPEJ, 18,325-338, Trans. AIME, 265, Oct. 1978.
- Parashar M., J.A. Wheeler, G. Pope, K. Wang and P. Wang: "A New Generation EOS Compositional Reservoir Simulator: Part II - Framework and Multiprocessing," paper SPE 37977 presented at the SPE Reservoir Simulation Symposium, Dallas, TX., Jun. 8-11, 1997.
- Peng, D.Y. and Robinson, D.B." A New Two-Constant Equation of State," Ind. Eng. Chem. Fundam., 1976.
- Perschke, D.R.: "Equation of State Phase Behavior Modeling for Compositional Simulator," Ph.D. Dissertation, University of Texas at Austin, 1988.
- Pope, G.A., W. Wu, G. Narayanaswamy, M. Delshad, M.M. Sharma and P. Wang: "Modeling Relative Permeability Effects in Gas-Condensate Reservoirs," SPE Reservoir Evaluation & Engineering, April, 2000.
- Prouvost, L., G. A. Pope and B. A. Rouse. "Microemulsion Phase Behavior: A Thermodynamic Modeling of the Phase Partitioning of Amphiphilic Species," SPEJ, 693-703, Oct. 1985.
- Reed, R. L. and R. N. Healy.. "Some Physico-Chemical Aspects of Microemulsion Flooding: A Review," *Improved Oil recovery by Surfactant and Polymer Flooding*, D. O. Shah and R. S. Schechter(eds.), Academic Press, New York. 1977.

- Reid, R.C., Prausnitz, J.M. and Poling, B.E.: *The Properties of Gases and Liquids*, Fourth Edition, McGraw-Hill, Inc., NY, 1987.
- Sandler, Stanley I.: *Chemical and Engineering Thermodynamics*, Third Edition, John Wiley & Sons, Inc., 1999.
- Thompson, J. F., Warsi, Z.U.A, and Mastin C. W., *Numerical Grid Generation – Foundations and Applications*, Elsevier Science Publishing, New York, 1985.
- Trangenstein, J.A.," Customized Minimized Techniques for Phase Equilibrium Computations in Reservoir Simulation," Chem. Eng. Sci., 42, No.12, 1987.
- Uetani T., B. Guler, and K. Sepehrnoori, "Reservoir Simulation on High Performance Clusters," presented at the Sixth World Multiconference on Systematics, Cybernetics, and Informatics, Orlando, FL, 14-18 July, 2002.
- Wang, P., I. Yotov, M Wheeler, T. Arbogast, C. Dawson, M. Parashar and K. Sepehrnoori: "A New Generation EOS Compositional Reservoir Simulator: Part I - Formulation and Discretization," paper SPE 37979 presented at the SPE Reservoir Simulation Symposium, Dallas, TX, June 1997.
- Wang, P., Balay, S., Sepehrnoori, K., Wheeler, J., Abate, J., Smith, B. and Pope, G.A.:" A Fully Implicit Parallel EOS.Compositional Simulator for Large Scale Reservoir Simulation," SPE 51885 presented at the 1999 SPE 15<sup>th</sup> Reservoir Simulation Symposium, Houston, TX, Feb. 14-17, 1999.
- Winsor, P. A. *Solvent Properties of Amphiphilic Compounds* , Butterworths, London. 1954. Zhang, K., Y.S. Wu, C. Ding, K. Pruess and E. Elmroth: "Parallel Computing Techniques for Large-Scale Reservoir Simulation of Multi-Component and Multiphase Fluid Flow," paper SPE 66343 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, 11-14 February 2001.

## NOMENCLATURE

$A_{p1}, A_{p2}, A_{p3}, S_p$	viscosity model parameters for polymer
$a$	Chun-Huh IFT correlation parameter
$a_{31}, a_{31}, b_3$	adsorption model parameters for surfactant
$a_{41}, a_{41}, b_4$	adsorption model parameters for polymer
$C_i$	volume of $i^{\text{th}}$ component in fluid per fluid volume, fraction
$\tilde{C}_i$	overall volume of $i^{\text{th}}$ component per pore volume, fraction
$\hat{C}_i$	adsorbed volume of $i^{\text{th}}$ component per pore volume, fraction
$C_i^f$	volume of $i^{\text{th}}$ component in fluid per pore volume, fraction
$C_{ij}$	volume of $i^{\text{th}}$ component in $j^{\text{th}}$ phase per volume of $j^{\text{th}}$ phase, fraction
$C_{SE}$	effective salinity, meq/ml
$C_{SEOP}$	Optimum salinity, meq/ml
$C_{3\max,s}$	height of binodal curve at salinity $s$
$C_{43}$	concentration of polymer in aqueous or microemulsion phase, wt. %
CMC	critical micelle concentration
$c$	Chun-Huh IFT correlation parameter
$c_i$	compressibility of $i^{\text{th}}$ component, $\text{psi}^{-1}$
$c_i^o$	compressibility of $i^{\text{th}}$ component at reference pressure, $\text{psi}^{-1}$
$c_{i,\text{inj}}$	injected concentration of $i^{\text{th}}$ component in aqueous phase; vol. fraction (surfactant), wt. % (polymer), meq/ml (salt), lbmol/L (tracer)
$c_f$	compressibility of formation, $\text{psi}^{-1}$
$D$	depth, ft
$g$	gravity acceleration, $\text{ft/day}^2$
$\bar{J}$	Jacobian matrix
$\bar{J}_{i,j}$	Jacobian matrix for $i^{\text{th}}$ gridblock with respect to unknowns of $j^{\text{th}}$ gridblock

$K_i$	equilibrium ratio for $i^{\text{th}}$ component, $K_i = y_i / x_i$
$k$	absolute permeability, md
$\bar{k}$	absolute permeability tensor
$\tilde{k}$	transmissibility, md•ft
$k_{rj}$	relative permeability of $j^{\text{th}}$ phase, fraction
$k_x$	absolute permeability in x direction, md
$k_y$	absolute permeability in y direction, md
$k_z$	absolute permeability in z direction, md
$L_j$	ratio of moles in $j^{\text{th}}$ phase to the total number of moles in the mixture
$M_i$	molecular weight of $i^{\text{th}}$ component, lb/lbmol
$\bar{M}_j$	average molecular weight of $j^{\text{th}}$ phase, lb/lbmol
$N_i$	mole of $i^{\text{th}}$ component per pore volume, lbmol/ft <sup>3</sup>
$\vec{N}_i$	flux vector of $i^{\text{th}}$ species in units of mass of i per surface area-time
$N_i^f$	mole of $i^{\text{th}}$ component in fluid per pore volume, lbmol/ft <sup>3</sup>
$N_{\text{oil},3}$	mole of hydrocarbons in microemulsion phase per pore volume, lbmol/ft <sup>3</sup>
$N_{T,3}$	mole of all the components in microemulsion phase per pore volume, lbmol/ft <sup>3</sup>
$N_{tj}$	trapping number of $j^{\text{th}}$ phase, fraction
$n_a$	total number of aqueous components
$n_b$	total number of gridblocks
$n_c$	total number of hydrocarbon components
$n_j$	relative permeability exponent of $j^{\text{th}}$ phase
$P$	fluid pressure, psi
$P^o$	reference pressure, psi
$P_j$	fluid pressure for $j^{\text{th}}$ gridblock or $j^{\text{th}}$ phase, psi
$P_{\text{wf}}$	well bottomhole pressure, psi

$q_i$	sink(-) or source(+) of $i^{\text{th}}$ component, lbmol/day
$\vec{R}$	residual vector
$\vec{R}_i$	residual vector for $i^{\text{th}}$ gridblock
$R_i$	mass rate of production in units of mass of $i$ per bulk volume-time
$R_{m_i}$	residual of material balance equation of $i^{\text{th}}$ component, lbmol/day
$R_{m_{i,j}}$	residual of material balance equation of $i^{\text{th}}$ component for $j^{\text{th}}$ gridblock, lbmol/day
$R_v$	residual of volume constraint
$R_{v_i}$	residual of volume constraint for $i^{\text{th}}$ gridblock
$r_o$	equivalent well gridblock radius, ft
$r_w$	well radius, ft
$S_j$	saturation of $j^{\text{th}}$ phase, fraction
$S_{jr}$	residual saturation of $j^{\text{th}}$ phase, fraction
$S_p$	polymer viscosity correlation parameter, dimensionless
$T$	temperature, °F
$t$	time, day
$\bar{u}_j$	flux of $j^{\text{th}}$ phase, ft/day
$V$	primary variable
$V_{(I)}$	primary variable at (I,J,K) gridblock at new time level
$V_{(I+1)}$	primary variable at (I+1,J,K) gridblock at new time level
$V_{(I-1)}$	primary variable at (I-1,J,K) gridblock at new time level
$V_b$	bulk volume for a cell, ft <sup>3</sup>
$V_p$	pore volume for a cell, ft <sup>3</sup>
$\bar{v}_j$	molar volume of phase $j$ , ft <sup>3</sup>
$W_i$	overall concentration of $i$ in units of mass of $i$ per unit bulk volume
$x_i$	mole fraction of $i^{\text{th}}$ component in liquid phase
$\vec{x}$	solution vector

$\vec{x}_i$	solution vector for $i^{\text{th}}$ gridblock
$x_{ij}$	mole fraction of $i^{\text{th}}$ component in $j^{\text{th}}$ phase, fraction
$\Delta x$	length in x direction of a gridblock, ft
$y_i$	mole fraction of $i^{\text{th}}$ component in gaseous phase
$\Delta y$	length in y direction of a gridblock, ft
$\Delta z$	length in z direction of a gridblock, ft
$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$	microemulsion viscosity correlation parameters
$\Delta$	difference operator
$\gamma_j$	specific gravity of $j^{\text{th}}$ phase, psi/ft
$\Phi$	flow potential
$\phi$	porosity, fraction
$\phi_r$	porosity at reference pressure, fraction
$\lambda_{rj}$	relative mobility of $j^{\text{th}}$ phase
$\mu_j$	viscosity of $j^{\text{th}}$ phase, cp
$\rho_j$	mass density of $j^{\text{th}}$ phase, lb/ft <sup>3</sup>
$\rho_s$	mass density of stationary phase s, lb/ft <sup>3</sup>
$\sigma_{ow}$	interfacial tension between oleic and aqueous phases, dyne/cm
$\sigma_{23}$	interfacial tension between oleic and microemulsion phases, dyne/cm
$\tau_j$	trapping number model parameter of $j^{\text{th}}$ phase
$\omega_{ij}$	mass fraction of $i^{\text{th}}$ component in $j^{\text{th}}$ phase
$\xi_i$	molar density of $i^{\text{th}}$ component, lbmol/ft <sup>3</sup>
$\xi_i^0$	molar density of $i^{\text{th}}$ component at reference pressure, lbmol/ft <sup>3</sup>
$\xi_j$	molar density of $j^{\text{th}}$ phase, lbmol/ft <sup>3</sup>
$\xi_o$	overall molar density of hydrocarbon components, lbmol/ft <sup>3</sup>
$\nabla$	gradient operator
$\bar{\nabla} \bullet$	divergence operator



**Superscript**

High	high trapping number
Low	low trapping number
O	endpoint
n	old time level
n+1	new time level
new	after update by Newton iteration
old	before update by Newton iteration

**Subscript**

i	<p>index for component</p> <p>for <math>C_i</math>, <math>\tilde{C}_i</math>, <math>\hat{C}_i</math>, <math>C_i^f</math>, and <math>C_{ij}</math></p> <p>1: water, 2: oil, 3: surfactant, 4: polymer, 5: salt</p> <p>for other symbols</p> <p>1 to <math>n_c</math>: hydrocarbon component</p> <p><math>n_c + 1</math> to <math>n_c + n_a</math>: aqueous component except water</p> <p><math>H_2O</math>: water</p> <p>o: oil</p> <p>poly: polymer</p> <p>salt: salt</p> <p>surf: surfactant</p> <p>trac: tracer</p>
j	<p>index for phase</p> <p>1: aqueous phase (hybrid model)</p> <p>2: oleic phase</p> <p>3: gas (EOS model), microemulsion phase (hybrid model), aqueous or microemulsion phase (fully implicit model), displaced phase (trapping number calculations)</p>
j'	displacing phase for trapping number calculations
s	salinity level (0: low, 1: optimal)

## APPENDIX A—Trapping Number Model Calculations Used in the Jacobian

The expression for the relative permeability is

$$k_{rj} = k_{rj}^o \bar{S}_j^{n_j} \quad (\text{A.1})$$

where

$$\bar{S}_j = \frac{S_j - S_{jr}}{1 - S_{jr} - S_{j'r}} \quad (\text{A.2})$$

Let  $V = \{\ln K, p, N\}$  represent the independent variables in the solution. The partial derivative of Eq. A.1 with respect to  $V$  is

$$\frac{\partial k_{rj}}{\partial V} = \frac{\partial k_{rj}^o}{\partial V} \bar{S}_j^{n_j} + k_{rj}^o \left( n_j \bar{S}_j^{n_j-1} \frac{\partial \bar{S}_j}{\partial V} + \bar{S}_j^{n_j} \ln \bar{S}_j \frac{\partial n_j}{\partial V} \right) \quad (\text{A.3})$$

For evaluating Eq. A.3, we need the derivatives of the relative permeability endpoints, exponents and normalized saturation with respect to variables in  $V$ .

For the relative permeability endpoint (Eq. 2.32) we have

$$\frac{\partial k_{rj}^o}{\partial V} = - \frac{k_{rj}^{o\text{High}} - k_{rj}^{o\text{Low}}}{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}} \frac{\partial S_{j'r}}{\partial V} \quad (\text{A.4})$$

and for the relative permeability exponent (Eq. 2.33)

$$\frac{\partial n_j}{\partial V} = - \frac{n_j^{\text{High}} - n_j^{\text{Low}}}{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}} \frac{\partial S_{j'r}}{\partial V} \quad (\text{A.5})$$

Note that both Eqs. A.4 and A.5 require the partial derivative of the normalized saturation of the conjugate phase. The partial derivative of the normalized phase saturation Eq. A.2 is

$$\frac{\partial \bar{S}_j}{\partial V} = \frac{1}{(1 - S_{jr} - S_{j'r})} \left( \frac{\partial S}{\partial V} - \frac{\partial S_{jr}}{\partial V} \right) + \frac{S_j - S_{jr}}{(1 - S_{jr} - S_{j'r})^2} \left( \frac{\partial S_{j'r}}{\partial V} + \frac{\partial S_{jr}}{\partial V} \right) \quad (\text{A.6})$$

This involves the derivatives of residual saturations of both phases. The partial derivative of the residual saturations is given by

$$\frac{\partial S_{jr}}{\partial V} = -\frac{\tau_j T_j N_{T_j}^{\tau_j-1} (S_{jr}^{\text{Low}} - S_{jr}^{\text{High}})}{(1 + T_j N_{T_j}^{\tau_j})^2} \frac{\partial N_{T_j}}{\partial V} \quad (\text{A.7})$$

In its general form the trapping number can be expressed as

$$N_{T_j} = \frac{1}{\sigma_{jj'}} \left[ \left\{ k_x \left( -\frac{\partial p_{j'}}{\partial x} \right) \right\}^2 + \left\{ k_y \left( -\frac{\partial p_{j'}}{\partial y} \right) \right\}^2 + \left\{ k_z \left( -\frac{\partial p_{j'}}{\partial z} + \rho_j g \right) \right\}^2 \right]^{1/2} \quad (\text{A.8})$$

The trapping number derivatives with respect to the independent variables are

$$\frac{\partial N_{T_j}}{\partial V} = \frac{g k_z^2}{\sigma_{jj'}^2 N_{T_j}} \left( \rho_j g - \frac{\partial p_{j'}}{\partial z} \right) \frac{\partial \rho_j}{\partial V} - \frac{N_{T_j}}{\sigma_{jj'}} \frac{\partial \sigma_{jj'}}{\partial V} \quad (\text{A.9})$$

The calculations that ultimately result in the evaluation of Eq. A.3 start from the calculation of the trapping number derivatives (Eq. A.9). Note that Eq. A.9 uses the derivatives of phase densities that are obtained from the EOS model calculations. The IFT is assumed to be independent of hydrocarbon phase composition and pressure making its derivatives vanish. The residual saturation derivatives are calculated next (Eq. A.7) followed by the computation of the expressions in Eqs. A.6, A.5 and A.4. All the above calculations are performed along with the EOS model calculations for Jacobian terms for each Newton iteration. The trapping number model has been formulated for two phase flow (absence of gas) in this work hence the derivatives with respect to  $\ln K$  are not defined.

The use of Corey type relative permeability model simplifies the expressions for the relative permeability derivatives. A generalized approach would be to use lookup tables and an expression as shown below:

$$\log k_{rj} = \log k_{rj}^0 + \log \bar{S}_j^{n_j} + \frac{\log \left( \frac{k_{rj}}{k_{rj}^0} \right)^{\text{Low}} - \log \bar{S}_j}{1 + T_{j'} (N_{T_{j'}})^{\tau_{j'}}} \quad (\text{A.10})$$

## APPENDIX B—Residuals and Derivation of Jacobian for Newton Iteration

### Volume constraint

Residual of volume constraint for each gridblock is given by the following equation.

$$R_v = \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}}{\xi_{surf}} + \frac{\sum_{i=1}^{n_c} N_i}{\xi_o} - 1.0 \quad (B.1)$$

In Eq. B.1, the molar density of oil component is assumed to be the same as one of oleic phase ( $\xi_2$ ). The residual of volume constraint is only dependent on primary variables (or unknowns) at its own gridblock. The primary variables are  $V=(N_1, \dots, N_{n_c}, N_{n_c+1}, \dots, N_{n_c+n_a}, P, N_{H_2O})$  at new time level. Therefore, derivatives of the residual can be easily obtained as follows.

$$\frac{\partial R_v}{\partial V} = \begin{cases} \frac{1}{\xi_2} - \frac{\sum_{i=1}^{n_c} N_i}{\xi_2^2} \frac{\partial \xi_2}{\partial V} & (\text{if } V = N_1, \dots, N_{n_c}) \\ \frac{1}{\xi_{H_2O}} & (\text{if } V = N_{H_2O}) \\ \frac{1}{\xi_{surf}} & (\text{if } V = N_{surf}) \\ -\frac{N_{H_2O}}{\xi_{H_2O}^2} \xi_{H_2O}^o c_{H_2O}^o - \frac{N_{surf}}{\xi_{surf}^2} \xi_{surf}^o c_{surf}^o - \frac{\sum_{i=1}^{n_c} N_i}{\xi_2^2} \frac{\partial \xi_2}{\partial P} & (\text{if } V = P) \\ 0 & (\text{if } V = N_{poly}, N_{salt}, \text{ and } N_{trac}) \end{cases} \quad (B.2)$$

### Material Balance Equations

After finite differencing the material balance equations (Eq. B.3a and B.3b),

$$V_b \frac{\partial}{\partial t}(\phi N_i) - V_b \bar{\nabla} \bullet \left( \xi_2 \frac{\bar{k}k_{r2}}{\mu_2} x_{i2} (\nabla P - \gamma_2 \nabla D) + \xi_3 \frac{\bar{k}k_{r3}}{\mu_3} x_{i3} (\nabla P - \gamma_3 \nabla D) \right) \quad (B-3a)$$

$= q_i$  (for HC components)

$$V_b \frac{\partial}{\partial t}(\phi N_i) - V_b \bar{\nabla} \bullet \left( \xi_3 \frac{\bar{k}k_{r3}}{\mu_3} x_{i3} (\nabla P - \gamma_3 \nabla D) \right) = q_i \text{ (for aqueous components)} \quad (B-3b)$$

We can define the residual of material balance equation for component  $i$  at a gridblock of  $(I, J, K)$  as

$$\begin{aligned} R_{m_i(I, J, K)} = & \left[ \frac{V_b}{\Delta t} \left( (\phi N_i)^{n+1} - (\phi N_i)^n \right) \right]_{(I, J, K)} \\ & + \left[ \left( F_{i(I+1/2, J, K)} - F_{i(I-1/2, J, K)} \right)^{n+1} \right. \\ & \quad + \left( F_{i(I, J+1/2, K)} - F_{i(I, J-1/2, K)} \right)^{n+1} \\ & \quad \left. + \left( F_{i(I, J, K+1/2)} - F_{i(I, J, K-1/2)} \right)^{n+1} \right] \end{aligned} \quad (B.3c)$$

where

$$\phi = \phi_r (1 + c_f (P - P^0))$$

$$F_{i(I+1/2, J, K)} = \begin{cases} T_{i2(I+1/2, J, K)} V_{2(I+1/2, J, K)} + T_{i3(I+1/2, J, K)} V_{3(I+1/2, J, K)} \\ \quad \text{(for hydrocarbon component)} \\ T_{i3(I+1/2, J, K)} V_{3(I+1/2, J, K)} \\ \quad \text{(for aqueous component)} \end{cases}$$

$$F_{i(I-1/2, J, K)} = \begin{cases} T_{i2(I-1/2, J, K)} V_{2(I-1/2, J, K)} + T_{i3(I-1/2, J, K)} V_{3(I-1/2, J, K)} \\ \quad \text{(for hydrocarbon component)} \\ T_{i3(I-1/2, J, K)} V_{3(I-1/2, J, K)} \\ \quad \text{(for aqueous component)} \end{cases}$$

$$F_{i(I,J+1/2,K)} = \begin{cases} T_{i2(I,J+1/2,K)} V_{2(I,J+1/2,K)} + T_{i3(I,J+1/2,K)} V_{3(I,J+1/2,K)} \\ \quad \text{(for hydrocarbon component)} \\ T_{i3(I,J+1/2,K)} V_{3(I,J+1/2,K)} \\ \quad \text{(for aqueous component)} \end{cases}$$

$$F_{i(I,J-1/2,K)} = \begin{cases} T_{i2(I,J-1/2,K)} V_{2(I,J-1/2,K)} + T_{i3(I,J-1/2,K)} V_{3(I,J-1/2,K)} \\ \quad \text{(for hydrocarbon component)} \\ T_{i3(I,J-1/2,K)} V_{3(I,J-1/2,K)} \\ \quad \text{(for aqueous component)} \end{cases}$$

$$F_{i(I,J,K+1/2)} = \begin{cases} T_{i2(I,J,K+1/2)} V_{2(I,J,K+1/2)} + T_{i3(I,J,K+1/2)} V_{3(I,J,K+1/2)} \\ \quad \text{(for hydrocarbon component)} \\ T_{i3(I,J,K+1/2)} V_{3(I,J,K+1/2)} \\ \quad \text{(for aqueous component)} \end{cases}$$

$$F_{i(I,J,K-1/2)} = \begin{cases} T_{i2(I,J,K-1/2)} V_{2(I,J,K-1/2)} + T_{i3(I,J,K-1/2)} V_{3(I,J,K-1/2)} \\ \quad \text{(for hydrocarbon component)} \\ T_{i3(I,J,K-1/2)} V_{3(I,J,K-1/2)} \\ \quad \text{(for aqueous component)} \end{cases}$$

where

$$T_{ij(I+1/2,J,K)} = \begin{cases} T_{ij(I,J,K)} & (v_{j(I+1/2,J,K)} \geq 0) \\ T_{ij(I+1,J,K)} & (v_{j(I+1/2,J,K)} < 0) \end{cases}$$

$$T_{ij(I-1/2,J,K)} = \begin{cases} T_{ij(I-1,J,K)} & (v_{j(I-1/2,J,K)} \geq 0) \\ T_{ij(I,J,K)} & (v_{j(I-1/2,J,K)} < 0) \end{cases}$$

$$T_{ij(I,J+1/2,K)} = \begin{cases} T_{ij(I,J,K)} & (v_{j(I,J+1/2,K)} \geq 0) \\ T_{ij(I,J+1,K)} & (v_{j(I,J+1/2,K)} < 0) \end{cases}$$

$$T_{ij(I,J-1/2,K)} = \begin{cases} T_{ij(I,J-1,K)} & (v_{j(I,J-1/2,K)} \geq 0) \\ T_{ij(I,J,K)} & (v_{j(I,J-1/2,K)} < 0) \end{cases}$$

$$T_{ij(I,J,K+1/2)} = \begin{cases} T_{ij(I,J,K)} & (v_{j(I,J,K+1/2)} \geq 0) \\ T_{ij(I,J,K+1)} & (v_{j(I,J,K+1/2)} < 0) \end{cases}$$

$$T_{ij(I,J,K-1/2)} = \begin{cases} T_{ij(I,J,K-1)} & (v_{j(I,J,K-1/2)} \geq 0) \\ T_{ij(I,J,K)} & (v_{j(I,J,K-1/2)} < 0) \end{cases}$$

where

$$T_{ij(I,J,K)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I,J,K)}$$

$$T_{ij(I+1,J,K)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I+1,J,K)}$$

$$T_{ij(I-1,J,K)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I-1,J,K)}$$

$$T_{ij(I,J+1,K)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I,J+1,K)}$$

$$T_{ij(I,J-1,K)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I,J-1,K)}$$

$$T_{ij(I,J,K+1)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I,J,K+1)}$$

$$T_{ij(I,J,K-1)} = \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right)_{(I,J,K-1)}$$

$$v_{j(I+1/2,J,K)} = -\tilde{k}_{(I+1/2,J,K)} \left( (P_j - \gamma_j D)_{(I+1,J,K)} - (P_j - \gamma_j D)_{(I,J,K)} \right)$$

$$v_{j(I-1/2,J,K)} = -\tilde{k}_{(I-1/2,J,K)} \left( (P_j - \gamma_j D)_{(I,J,K)} - (P_j - \gamma_j D)_{(I-1,J,K)} \right)$$

$$v_{j(I,J+1/2,K)} = -\tilde{k}_{(I,J+1/2,K)} \left( (P_j - \gamma_j D)_{(I,J+1,K)} - (P_j - \gamma_j D)_{(I,J,K)} \right)$$

$$v_{j(I,J-1/2,K)} = -\tilde{k}_{(I,J-1/2,K)} \left( (P_j - \gamma_j D)_{(I,J,K)} - (P_j - \gamma_j D)_{(I,J-1,K)} \right)$$

$$v_{j(I,J,K+1/2)} = -\tilde{k}_{(I,J,K+1/2)} \left( (P_j - \gamma_j D)_{(I,J,K+1)} - (P_j - \gamma_j D)_{(I,J,K)} \right)$$

$$v_{j(I,J,K-1/2)} = -\tilde{k}_{(I,J,K-1/2)} \left( (P_j - \gamma_j D)_{(I,J,K)} - (P_j - \gamma_j D)_{(I,J,K-1)} \right)$$

where

$$\tilde{k}_{(I+1/2,J,K)} = \frac{2\Delta y_{(I,J,K)}\Delta z_{(I,J,K)}k_{x(I,J,K)}k_{x(I+1,J,K)}}{\Delta x_{(I+1,J,K)}k_{x(I,J,K)} + \Delta x_{(I,J,K)}k_{x(I+1,J,K)}}$$

$$\tilde{k}_{(I-1/2,J,K)} = \frac{2\Delta y_{(I,J,K)}\Delta z_{(I,J,K)}k_{x(I,J,K)}k_{x(I-1,J,K)}}{\Delta x_{(I-1,J,K)}k_{x(I,J,K)} + \Delta x_{(I,J,K)}k_{x(I-1,J,K)}}$$

$$\tilde{k}_{(I,J+1/2,K)} = \frac{2\Delta x_{(I,J,K)}\Delta z_{(I,J,K)}k_{y(I,J,K)}k_{y(I,J+1,K)}}{\Delta y_{(I,J+1,K)}k_{y(I,J,K)} + \Delta y_{(I,J,K)}k_{y(I,J+1,K)}}$$

$$\tilde{k}_{(I,J-1/2,K)} = \frac{2\Delta x_{(I,J,K)}\Delta z_{(I,J,K)}k_{y(I,J,K)}k_{y(I,J-1,K)}}{\Delta y_{(I,J-1,K)}k_{y(I,J,K)} + \Delta y_{(I,J,K)}k_{y(I,J-1,K)}}$$

$$\tilde{k}_{(I,J,K+1/2)} = \frac{2\Delta x_{(I,J,K)}\Delta y_{(I,J,K)}k_{z(I,J,K)}k_{z(I,J,K+1)}}{\Delta z_{(I,J,K+1)}k_{z(I,J,K)} + \Delta z_{(I,J,K)}k_{z(I,J,K+1)}}$$

$$\tilde{k}_{(I,J,K-1/2)} = \frac{2\Delta x_{(I,J,K)}\Delta y_{(I,J,K)}k_{z(I,J,K)}k_{z(I,J,K-1)}}{\Delta z_{(I,J,K-1)}k_{z(I,J,K)} + \Delta z_{(I,J,K)}k_{z(I,J,K-1)}}$$

$$q_{i(I,J,K)} =$$

for constant rate injection (that is, source),



$$\frac{q_{H_2O}|_{\text{surface}} (PI_t)_z}{\sum_{m=z_t}^{z_b} (PI_t)_m} \quad (\text{if } i = H_2O)$$

where

$$(PI_t)_z = \sum_{j=2}^3 (PI_j)_z$$

where

$$(PI_j)_z = c_z \left( \frac{k_{rj}}{\mu_j} \right)_z$$

where

$$c_z = \frac{\sqrt{k_x k_y} \Delta z}{25.14872 \ln(r_o / r_w)}$$

$$c_{i,inj} \bullet (q_{H_2O})_{(I,J,K)} \frac{\xi_{surf}}{\xi_{H_2O}} \bigg|_{\text{surface}} \quad (\text{if } i = \text{surfactant})$$

$$\frac{c_{i,inj}}{100} \bullet (q_{H_2O})_{(I,J,K)} \frac{M_{H_2O}}{M_{poly}} \quad (\text{if } i = \text{polymer})$$

$$\frac{c_{i,inj}}{16.037} \bullet \frac{(q_{H_2O})_{(I,J,K)}}{\xi_{H_2O}|_{\text{surface}}} \quad (\text{if } i = \text{salt or tracer})$$

for a constant pressure injection well (that is, the source term),

$$\frac{c_z}{1 + c_{surf,inj}} \bullet \xi_{H_2O} \bigg|_{inj} \left( \frac{k_{r3}}{\mu_3} \right)_z (P_{wf} - P_3)_z \quad (\text{if } i = H_2O)$$

$$c_{i,inj} \bullet (q_{H_2O})_{(I,J,K)} \frac{\xi_{surf}}{\xi_{H_2O}} \bigg|_{\text{surface}} \quad (\text{if } i = \text{surfactant})$$

$$\frac{c_{i,\text{inj}}}{100} \bullet (q_{\text{H}_2\text{O}})_{(I,J,K)} \frac{M_{\text{H}_2\text{O}}}{M_{\text{poly}}} \quad (\text{if } i = \text{polymer})$$

$$\frac{c_{i,\text{inj}}}{16.037} \bullet \frac{(q_{\text{H}_2\text{O}})_{(I,J,K)}}{\xi_{\text{H}_2\text{O}}|_{\text{surface}}} \quad (\text{if } i = \text{salt or tracer})$$

for a constant pressure production well (that is, the sink term),

$$\sum_{j=2}^3 \left( (\xi_j x_{ij} P I_j)_z \bullet (P_{\text{wf}} - P_j)_z \right) \quad (\text{for all components})$$

### Derivatives for Accumulation Term

The accumulation term in residual of material balance equation (Eq. B.3c) is only dependent on primary variables at its own gridblock. Therefore, the derivative of the accumulation term is

$$\frac{\partial}{\partial V} \left( \frac{V_b}{\Delta t} \left( (\phi N_i)^{n+1} - (\phi N_i)^n \right) \right)_{(I,J,K)} = \begin{cases} \left( \frac{V_b}{\Delta t} \phi^{n+1} \right)_{(I,J,K)} & (\text{if } V = N_i) \\ \left( \frac{V_b}{\Delta t} \phi_f c_f N_i^{n+1} \right)_{(I,J,K)} & (\text{if } V = P) \end{cases} \quad (\text{B.4})$$

### Derivatives for Flux Term

The derivatives of flux term in residual of material balance equation (Eq. B.3c) are more complicated because the flux term is dependent on primary variables at six neighbor gridblocks as well as those at its own gridblock. Here we derive the derivative of  $F_{i(I+1/2,J,K)}$  for an aqueous component only that means the flux between (I,J,K) and (I+1,J,K) gridblocks. The derivatives of the remaining terms and the derivatives of flux for hydrocarbon components can be derived following the similar procedure. Several indices are omitted for simplicity.

$$\begin{aligned} & \frac{\partial}{\partial V} (F_{i(I+1/2)}) \\ &= \frac{\partial}{\partial V} (T_{i3(I+1/2)} v_{3(I+1/2)}) \\ &= \frac{\partial}{\partial V} (T_{i3(I+1/2)}) v_{3(I+1/2)} + T_{i3(I+1/2)} \frac{\partial v_{3(I+1/2)}}{\partial V} \end{aligned} \quad (\text{B.5})$$

If upstream is I+1 gridblock, Eq. B.5 is rewritten as,

$$\begin{aligned}
& \frac{\partial}{\partial V} \left( T_{i3(I+1/2)} \right) v_{3(I+1/2)} + T_{i3(I+1/2)} \frac{\partial v_{3(I+1/2)}}{\partial V} \\
&= \frac{\partial}{\partial V} \left( T_{i3(I+1)} \right) v_{3(I+1/2)} + T_{i3(I+1)} \frac{\partial v_{3(I+1/2)}}{\partial V} \\
&= \frac{\partial}{\partial V} \left( \left( \xi_3 \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I+1)} \right) v_{3(I+1/2)} + \left( \xi_3 \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I+1)} \frac{\partial v_{3(I+1/2)}}{\partial V}
\end{aligned} \tag{B.6}$$

where

$$\begin{aligned}
& \frac{\partial}{\partial V} \left( \left( \xi_3 \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I+1)} \right) \\
&= \begin{cases} \frac{\partial \xi_{3(I+1)}}{\partial V} \left( \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I+1)} + \left( \xi_3 \frac{x_{i3}}{\mu_3} \right)_{(I+1)} \frac{\partial k_{r3(I+1)}}{\partial V} \\ \quad - \left( \frac{\xi_3 k_{r3}}{\mu_3^2} x_{i3} \right)_{(I+1)} \frac{\partial \mu_{3(I+1)}}{\partial V} + \left( \xi_3 \frac{k_{r3}}{\mu_3} \right)_{(I+1)} \frac{\partial x_{i3(I+1)}}{\partial V} & \text{(if } V = V_{(I+1)}) \\ 0 & \text{(if } V = V_{(I-1)} \text{ and } V_{(I)}) \end{cases}
\end{aligned}$$

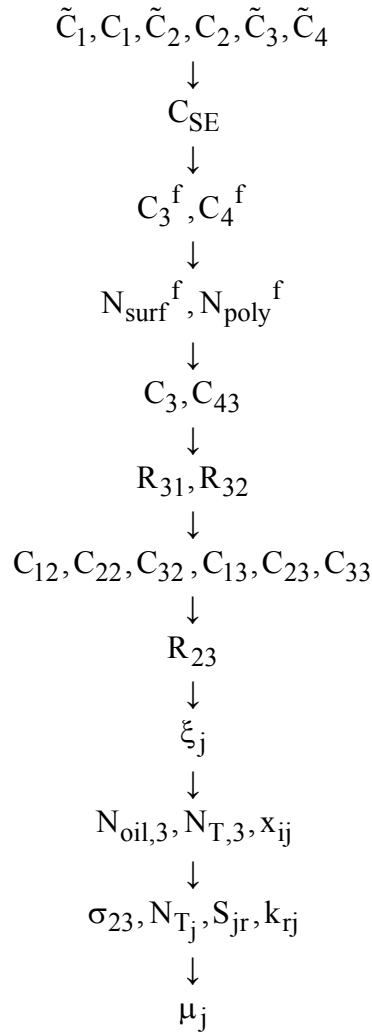
If upstream is I gridblock, Eq. B.5 is rewritten as,

$$\begin{aligned}
& \frac{\partial}{\partial V} \left( T_{i3(I+1/2)} \right) v_{3(I+1/2)} + T_{i3(I+1/2)} \frac{\partial v_{3(I+1/2)}}{\partial V} \\
&= \frac{\partial}{\partial V} \left( T_{i3(I)} \right) v_{3(I+1/2)} + T_{i3(I)} \frac{\partial v_{3(I+1/2)}}{\partial V} \\
&= \frac{\partial}{\partial V} \left( \left( \xi_3 \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I)} \right) v_{3(I+1/2)} + \left( \xi_3 \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I)} \frac{\partial v_{3(I+1/2)}}{\partial V}
\end{aligned} \tag{B.7}$$

where

$$\begin{aligned}
& \frac{\partial}{\partial V} \left( \left( \xi_3 \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I)} \right) \\
& = \begin{cases} \frac{\partial \xi_{3(I)}}{\partial V} \left( \frac{k_{r3}}{\mu_3} x_{i3} \right)_{(I)} + \left( \xi_3 \frac{x_{i3}}{\mu_3} \right)_{(I)} \frac{\partial k_{r3(I)}}{\partial V} & \text{(if } V = V_{(I)} \text{)} \\ - \left( \frac{\xi_3 k_{r3}}{\mu_3^2} x_{i3} \right)_{(I)} \frac{\partial \mu_{3(I)}}{\partial V} + \left( \xi_3 \frac{k_{r3}}{\mu_3} \right)_{(I)} \frac{\partial x_{i3(I)}}{\partial V} & \text{(if } V = V_{(I-1)} \text{ and } V_{(I+1)} \text{)} \\ 0 & \text{(if } V = V_{(I-1)} \text{ and } V_{(I+1)} \text{)} \end{cases}
\end{aligned}$$

The derivatives of molar density, relative permeability, viscosity, and mole fraction of each phase can be calculated using derivatives of dependent variables which can be obtained by following procedure.



A. Derivatives of  $\tilde{C}_1, C_1, \tilde{C}_2, C_2, \tilde{C}_3, \tilde{C}_4$

$$\tilde{C}_1 = \frac{N_{H_2O}}{\xi_{H_2O}} \quad (B.8)$$

where

$$\xi_{H_2O} = \xi_{H_2O}^0 (1 + c_{H_2O}^0 (P - P^0))$$

$$\frac{\partial \tilde{C}_1}{\partial N_i} = \begin{cases} \frac{1}{\xi_{H_2O}} & (\text{if } i = H_2O) \\ 0 & (\text{if } i \neq H_2O) \end{cases} \quad (B.9)$$

$$\frac{\partial \tilde{C}_1}{\partial P} = -\frac{N_{H_2O}}{\xi_{H_2O}^2} (\xi_{H_2O}^0 c_{H_2O}^0) \quad (B.10)$$

$$C_1 = \left( \frac{N_{H_2O}}{\xi_{H_2O}} \right) \frac{1}{1 - \hat{C}_3} \approx \frac{N_{H_2O}}{\xi_{H_2O}} \quad (B.11)$$

Therefore, derivatives of  $C_1$  are the same as those of  $\tilde{C}_1$ .

$$\tilde{C}_2 = \frac{1}{\xi_0} \sum_{i=1}^{n_c} N_i \quad (B.12)$$

$$\frac{\partial \tilde{C}_2}{\partial N_i} = \begin{cases} \frac{1}{\xi_0} \left( 1 - \frac{1}{\xi_0} \frac{\partial \xi_0}{\partial N_i} \sum_{i=1}^{n_c} N_i \right) & (\text{for hydrocarbon components}) \\ 0 & (\text{for aqueous components}) \end{cases} \quad (B.13)$$

$$\frac{\partial \tilde{C}_2}{\partial P} = -\frac{1}{\xi_0^2} \sum_{i=1}^{n_c} N_i \frac{\partial \xi_0}{\partial P} \quad (B.14)$$

$$C_2 = \frac{1}{\xi_0} \sum_{i=1}^{n_c} N_i \left( \frac{1}{1 - \hat{C}_3} \right) \approx \frac{1}{\xi_0} \sum_{i=1}^{n_c} N_i \quad (B.15)$$

Therefore, derivatives of  $C_2$  are the same as those of  $\tilde{C}_2$ .

$$\tilde{C}_3 = \frac{N_{\text{surf}}}{\xi_{\text{surf}}} \quad (\text{B.16})$$

where

$$\xi_{\text{surf}} = \xi_{\text{surf}}^0 (1 + c_{\text{surf}}^0 (P - P^0))$$

$$\frac{\partial \tilde{C}_3}{\partial N_i} = \begin{cases} \frac{1}{\xi_{\text{surf}}} & (\text{if } i = \text{surfactant}) \\ 0 & (\text{if } i \neq \text{surfactant}) \end{cases} \quad (\text{B.17})$$

$$\frac{\partial \tilde{C}_3}{\partial P} = -\frac{N_{\text{surf}}}{\xi_{\text{surf}}^2} (\xi_{\text{surf}}^0 c_{\text{surf}}^0) \quad (\text{B.18})$$

$$\tilde{C}_4 = \frac{N_{\text{poly}}}{\xi_{\text{poly}}} \quad (\text{B.19})$$

where

$$\xi_{\text{poly}} = \xi_{\text{poly}}^0 (1 + c_{\text{poly}}^0 (P - P^0))$$

$$\frac{\partial \tilde{C}_4}{\partial N_i} = \begin{cases} \frac{1}{\xi_{\text{poly}}} & (\text{if } i = \text{polymer}) \\ 0 & (\text{if } i \neq \text{polymer}) \end{cases} \quad (\text{B.20})$$

$$\frac{\partial \tilde{C}_4}{\partial P} = -\frac{N_{\text{poly}}}{\xi_{\text{poly}}^2} (\xi_{\text{poly}}^0 c_{\text{poly}}^0) \quad (\text{B.21})$$

## B. Derivatives of $C_{\text{SE}}$

$$C_{\text{SE}} = 16.037 \xi_{\text{H}_2\text{O}} \frac{N_{\text{salt}}}{N_{\text{H}_2\text{O}}} \quad (\text{B.22})$$

$$\frac{\partial C_{SE}}{\partial N_i} = \begin{cases} 16.037 \frac{\xi_{H_2O}}{N_{H_2O}} & (\text{if } i = \text{salt}) \\ -16.037 \frac{\xi_{H_2O} N_{\text{salt}}}{N_{H_2O}^2} & (\text{if } i = H_2O) \\ 0 & (\text{for other components}) \end{cases} \quad (B.23)$$

$$\frac{\partial C_{SE}}{\partial P} = 16.037 (\xi_{H_2O}^0 c_{H_2O}^0) \frac{N_{\text{salt}}}{N_{H_2O}} \quad (B.24)$$

C. Derivatives of  $C_3^f, C_4^f$

$$C_3^f = \tilde{C}_3 - \hat{C}_3 = \frac{\tilde{C}_1}{2b_3} \left( -a_3 - 1 + b_3 \frac{\tilde{C}_3}{\tilde{C}_1} + \sqrt{\left( a_3 + 1 - b_3 \frac{\tilde{C}_3}{\tilde{C}_1} \right)^2 + 4b_3 \left( \frac{\tilde{C}_3}{\tilde{C}_1} \right)} \right) \quad (\text{see Appendix C}) \quad (B.25)$$

where

$$a_3 = a_{31} + a_{32} C_{SE}$$

$$\frac{\partial C_3^f}{\partial N_i} = \frac{\partial C_3^f}{\partial \tilde{C}_1} \frac{\partial \tilde{C}_1}{\partial N_i} + \frac{\partial C_3^f}{\partial \tilde{C}_3} \frac{\partial \tilde{C}_3}{\partial N_i} + \frac{\partial C_3^f}{\partial C_{SE}} \frac{\partial C_{SE}}{\partial N_i} \quad (B.26)$$

where

$$\frac{\partial C_3^f}{\partial \tilde{C}_1} = \frac{1}{2b_3} \left\{ A - \frac{b_3 \tilde{C}_3}{\tilde{C}_1} \left( 1 - \frac{1}{\sqrt{B}} \left( \frac{b_3 \tilde{C}_3}{\tilde{C}_1} - a_3 - 1 \right) \right) \right\}$$

$$\frac{\partial C_3^f}{\partial \tilde{C}_3} = \frac{1}{2} \left( 1 + \frac{1}{\sqrt{B}} \left( \frac{b_3 \tilde{C}_3}{\tilde{C}_1} - a_3 + 1 \right) \right)$$

$$\frac{\partial C_3^f}{\partial C_{SE}} = \frac{a_{32} \tilde{C}_1}{2b_3} \left( \frac{1}{\sqrt{B}} \left( a_3 + 1 - \frac{b_3 \tilde{C}_3}{\tilde{C}_1} \right) - 1 \right)$$

where

$$A = -a_3 - 1 + b_3 \frac{\tilde{C}_3}{\tilde{C}_1} + \sqrt{\left(a_3 + 1 - b_3 \frac{\tilde{C}_3}{\tilde{C}_1}\right)^2 + 4b_3 \left(\frac{\tilde{C}_3}{\tilde{C}_1}\right)}$$

$$B = \left(a_3 + 1 - b_3 \frac{\tilde{C}_3}{\tilde{C}_1}\right)^2 + 4b_3 \left(\frac{\tilde{C}_3}{\tilde{C}_1}\right)$$

$$\frac{\partial C_3^f}{\partial P} = \frac{\partial C_3^f}{\partial \tilde{C}_1} \frac{\partial \tilde{C}_1}{\partial P} + \frac{\partial C_3^f}{\partial \tilde{C}_3} \frac{\partial \tilde{C}_3}{\partial P} + \frac{\partial C_3^f}{\partial C_{SE}} \frac{\partial C_{SE}}{\partial P} \quad (B.27)$$

$$C_4^f = \tilde{C}_4 - \hat{C}_4 = \frac{\tilde{C}_1}{2b_4} \left( -a_4 - 1 + b_4 \frac{\tilde{C}_4}{\tilde{C}_1} + \sqrt{\left(a_4 + 1 - b_4 \frac{\tilde{C}_4}{\tilde{C}_1}\right)^2 + 4b_4 \left(\frac{\tilde{C}_4}{\tilde{C}_1}\right)} \right) \quad (\text{see Appendix C}) \quad (B.28)$$

where

$$a_4 = a_{41} + a_{42} C_{SE}$$

$$\frac{\partial C_4^f}{\partial N_i} = \frac{\partial C_4^f}{\partial \tilde{C}_1} \frac{\partial \tilde{C}_1}{\partial N_i} + \frac{\partial C_4^f}{\partial \tilde{C}_4} \frac{\partial \tilde{C}_4}{\partial N_i} + \frac{\partial C_4^f}{\partial C_{SE}} \frac{\partial C_{SE}}{\partial N_i} \quad (B.29)$$

where

$$\frac{\partial C_4^f}{\partial \tilde{C}_1} = \frac{1}{2b_4} \left\{ A - \frac{b_4 \tilde{C}_4}{\tilde{C}_1} \left( 1 - \frac{1}{\sqrt{B}} \left( \frac{b_4 \tilde{C}_4}{\tilde{C}_1} - a_4 - 1 \right) \right) \right\}$$

$$\frac{\partial C_4^f}{\partial \tilde{C}_4} = \frac{1}{2} \left( 1 + \frac{1}{\sqrt{B}} \left( \frac{b_4 \tilde{C}_4}{\tilde{C}_1} - a_4 + 1 \right) \right)$$

$$\frac{\partial C_4^f}{\partial C_{SE}} = \frac{a_{42} \tilde{C}_1}{2b_4} \left( \frac{1}{\sqrt{B}} \left( a_4 + 1 - \frac{b_4 \tilde{C}_4}{\tilde{C}_1} \right) - 1 \right)$$

where

$$A = -a_4 - 1 + b_4 \frac{\tilde{C}_4}{\tilde{C}_1} + \sqrt{\left(a_4 + 1 - b_4 \frac{\tilde{C}_4}{\tilde{C}_1}\right)^2 + 4b_4 \left(\frac{\tilde{C}_4}{\tilde{C}_1}\right)}$$



$$B = \left( a_4 + 1 - b_4 \frac{\tilde{C}_4}{\tilde{C}_1} \right)^2 + 4b_4 \left( \frac{\tilde{C}_4}{\tilde{C}_1} \right)$$

$$\frac{\partial C_4^f}{\partial P} = \frac{\partial C_4^f}{\partial \tilde{C}_1} \frac{\partial \tilde{C}_1}{\partial P} + \frac{\partial C_4^f}{\partial \tilde{C}_4} \frac{\partial \tilde{C}_4}{\partial P} + \frac{\partial C_4^f}{\partial C_{SE}} \frac{\partial C_{SE}}{\partial P} \quad (B.30)$$

D. Derivatives of  $N_{\text{surf}}^f, N_{\text{poly}}^f$

$$N_{\text{surf}}^f = \xi_{\text{surf}} C_3^f \quad (B.31)$$

$$\frac{\partial N_{\text{surf}}^f}{\partial N_i} = \xi_{\text{surf}} \frac{\partial C_3^f}{\partial N_i} \quad (B.32)$$

$$\frac{\partial N_{\text{surf}}^f}{\partial P} = (\xi_{\text{surf}}^0 c_{\text{surf}}^0) C_3^f + \xi_{\text{surf}} \frac{\partial C_3^f}{\partial P} \quad (B.33)$$

$$N_{\text{poly}}^f = \xi_{\text{poly}} C_4^f \quad (B.34)$$

$$\frac{\partial N_{\text{poly}}^f}{\partial N_i} = \xi_{\text{poly}} \frac{\partial C_4^f}{\partial N_i} \quad (B.35)$$

$$\frac{\partial N_{\text{poly}}^f}{\partial P} = (\xi_{\text{poly}}^0 c_{\text{poly}}^0) C_4^f + \xi_{\text{poly}} \frac{\partial C_4^f}{\partial P} \quad (B.36)$$

E. Derivatives of  $C_3, C_{43}$

$$C_3 = \frac{N_{\text{surf}}^f}{\xi_{\text{surf}}} \left( \frac{1}{1 - \hat{C}_3} \right) \approx \frac{N_{\text{surf}}^f}{\xi_{\text{surf}}} = C_3^f \quad (B.37)$$

Therefore, derivatives of  $C_3$  are the same as those of  $C_3^f$ .

$$C_{43} = 100 \frac{M_{\text{poly}} N_{\text{poly}}^f}{M_{\text{H}_2\text{O}} N_{\text{H}_2\text{O}}} \quad (B.38)$$

$$\frac{\partial C_{43}}{\partial N_i} = \begin{cases} 100 \frac{M_{\text{poly}}}{M_{\text{H}_2\text{O}}} \frac{1}{N_{\text{H}_2\text{O}}} \left( \frac{\partial N_{\text{poly}}^f}{\partial N_{\text{H}_2\text{O}}} - \frac{N_{\text{poly}}^f}{N_{\text{H}_2\text{O}}} \right) & (\text{if } i = \text{H}_2\text{O}) \\ 100 \frac{M_{\text{poly}}}{M_{\text{H}_2\text{O}}} \frac{1}{N_{\text{H}_2\text{O}}} \frac{\partial N_{\text{poly}}^f}{\partial N_{\text{poly}}} & (\text{if } i = \text{polymer}) \\ 100 \frac{M_{\text{poly}}}{M_{\text{H}_2\text{O}}} \frac{1}{N_{\text{H}_2\text{O}}} \frac{\partial N_{\text{poly}}^f}{\partial N_{\text{salt}}} & (\text{if } i = \text{salt}) \\ 0 & (\text{for other components}) \end{cases} \quad (\text{B.39})$$

$$\frac{\partial C_{43}}{\partial P} = 100 \frac{M_{\text{poly}}}{M_{\text{H}_2\text{O}}} \frac{1}{N_{\text{H}_2\text{O}}} \frac{\partial N_{\text{poly}}^f}{\partial P} \quad (\text{B.40})$$

F. Derivatives of  $R_{31}, R_{32}$

$$R_{31} = \frac{C_3^f}{\tilde{C}_1} \quad (\text{B.41})$$

$$\frac{\partial R_{31}}{\partial V} = \frac{1}{\tilde{C}_1} \frac{\partial C_3^f}{\partial V} - \frac{C_3^f}{\tilde{C}_1^2} \frac{\partial \tilde{C}_1}{\partial V} \quad (\text{B.42})$$

$$R_{32} = \frac{A}{R_{31}} \quad (\text{for } C_{\text{SE}} \leq C_{\text{SEOP}}) \quad (\text{B.43})$$

where

$$A = (A_0 - A_1) \left( 1 - \frac{C_{\text{SE}}}{C_{\text{SEOP}}} \right) + A_1$$

$$\frac{\partial R_{32}}{\partial V} = \frac{1}{R_{31}} \left( \frac{A_1 - A_0}{C_{\text{SEOP}}} \frac{\partial C_{\text{SE}}}{\partial V} - \frac{A}{R_{31}} \frac{\partial R_{31}}{\partial V} \right) \quad (\text{B.44})$$

G. Derivatives of  $C_{12}, C_{22}, C_{32}, C_{13}, C_{23}, C_{33}$

If oleic and microemulsion phases exist ( $C_{33} > C_3$ ),

$$C_{12} = 0, \quad C_{22} = 1, \quad C_{32} = 0 \quad (\text{B.45})$$

$$\frac{\partial C_{12}}{\partial V} = \frac{\partial C_{22}}{\partial V} = \frac{\partial C_{32}}{\partial V} = 0 \quad (\text{B.46})$$

$$C_{13} = \frac{R_{32}}{R_{31} + R_{31}R_{32} + R_{32}} \quad (\text{B.47})$$

$$\frac{\partial C_{13}}{\partial V} = -\frac{1}{(R_{31} + R_{31}R_{32} + R_{32})^2} \left( R_{32}(1 + R_{32}) \frac{\partial R_{31}}{\partial V} - R_{31} \frac{\partial R_{32}}{\partial V} \right) \quad (\text{B.48})$$

$$C_{23} = 1 - (1 + R_{31})C_{13} \quad (\text{B.49})$$

$$\frac{\partial C_{23}}{\partial V} = -C_{13} \frac{\partial R_{31}}{\partial V} - (1 + R_{31}) \frac{\partial C_{13}}{\partial V} \quad (\text{B.50})$$

$$C_{33} = 1 - C_{13} - C_{23} \quad (\text{B.51})$$

$$\frac{\partial C_{33}}{\partial V} = -\frac{\partial C_{13}}{\partial V} - \frac{\partial C_{23}}{\partial V} \quad (\text{B.52})$$

If only microemulsion phase exists ( $C_{33} \leq C_3$ ),

$$C_{13} = C_1, \quad C_{23} = C_2, \quad C_{33} = C_3 \quad (\text{B.53})$$

$$\frac{\partial C_{13}}{\partial V} = \frac{\partial C_1}{\partial V}, \quad \frac{\partial C_{23}}{\partial V} = \frac{\partial C_2}{\partial V}, \quad \frac{\partial C_{33}}{\partial V} = \frac{\partial C_3}{\partial V} \quad (\text{B.54})$$

H. Derivatives of  $R_{23}$

$$R_{23} = \frac{C_{23}}{C_{33}} \quad (\text{B.55})$$

$$\frac{\partial R_{23}}{\partial V} = \frac{\partial C_{23}}{\partial V} \frac{1}{C_{33}} - \frac{C_{23}}{C_{33}^2} \frac{\partial C_{33}}{\partial V} \quad (\text{B.56})$$

I. Derivatives of  $\xi_j$

$$\xi_2 = f(P, T, x_{i2}) \quad (\text{B.57})$$

where

$$x_{i2} = \frac{N_i}{\sum_{i=1}^{n_c} N_i} \text{ (see Appendix C)}$$

Since mole fraction in oleic phase with microemulsion phase is the same as one in oleic phase with pure aqueous phase, we don't have to modify the existing derivatives of  $\xi_2$ .

Molar density of microemulsion phase is assumed to be the same as one of water component, Therefore,

$$\xi_3 \approx \xi_{H_2O} = \xi_{H_2O}^0 (1 + c_{H_2O} (P - P^0)) \quad (B.58)$$

$$\frac{\partial \xi_3}{\partial V} = \begin{cases} 0 & (\text{if } V \neq P) \\ \xi_{H_2O}^0 c_{H_2O}^0 & (\text{if } V = P) \end{cases} \quad (B.59)$$

J. Derivatives of  $N_{oil,3}, N_{T,3}, x_{ij}$

$$x_{i2} = \begin{cases} \frac{N_i}{\sum_{i=1}^{n_c} N_i} & (\text{for hydrocarbon components}) \\ 0 & (\text{for aqueous components}) \end{cases} \quad (B.60)$$

$$\frac{\partial x_{i2}}{\partial N_k} =$$

for hydrocarbon component, i

for hydrocarbon component, k

$$\begin{cases} \frac{1}{\sum_{i=1}^{n_c} N_i} \left( 1 - \frac{N_i}{\sum_{i=1}^{n_c} N_i} \right) & (\text{if } k = i) \\ -\frac{N_i}{\left( \sum_{i=1}^{n_c} N_i \right)^2} & (\text{if } k \neq i) \end{cases} \quad (B.61)$$

for aqueous component, k

$$0 \quad (B.62)$$

for aqueous component, i

$$0 \quad (B.63)$$

$$\frac{\partial x_{i2}}{\partial P} = 0 \quad (B.64)$$

$$x_{i3} = \frac{N_{i,3}}{N_{H_2O} + N_{oil,3} + N_{surf}^f + N_{poly}^f + N_{salt} + N_{trac}} \quad (B.65)$$

Equation B.66 is rewritten for hydrocarbon component, i as

$$x_{i3} = \frac{1}{N_{T,3}} \left( \frac{N_i}{\sum_{i=1}^{n_c} N_i} N_{oil,3} \right) \quad (B.66)$$

where

$$N_{T,3} = N_{H_2O} + N_{oil,3} + N_{surf}^f + N_{poly}^f + N_{salt} + N_{trac}$$

$$N_{oil,3} = \frac{\xi_2 C_{23}}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right) \quad (\text{see Appendix C})$$

for hydrocarbon component, k

if  $C_3 < \text{CMC}$

$$\frac{\partial x_{i3}}{\partial N_k} = 0 \quad (B.67)$$

if  $C_3 \geq \text{CMC}$

$$\begin{aligned} \frac{\partial x_{i3}}{\partial N_k} = & \left( \frac{\partial}{\partial N_k} \left( \frac{N_i}{\sum_{i=1}^{n_c} N_i} \right) N_{oil,3} + \frac{N_i}{\sum_{i=1}^{n_c} N_i} \frac{\partial N_{oil,3}}{\partial N_k} \right) \frac{1}{N_{T,3}} \\ & - \frac{N_i}{\sum_{i=1}^{n_c} N_i} \frac{N_{oil,3}}{N_{T,3}^2} \left( \frac{\partial N_{T,3}}{\partial N_k} \right) \end{aligned} \quad (B.68)$$

where

$$\frac{\partial}{\partial N_k} \left( \frac{N_i}{\sum_{i=1}^{n_c} N_i} \right) = \begin{cases} \frac{1}{\sum_{i=1}^{n_c} N_i} \left( 1 - \frac{N_i}{\sum_{i=1}^{n_c} N_i} \right) & (\text{if } k = i) \\ \frac{-N_i}{\left( \sum_{i=1}^{n_c} N_i \right)^2} & (\text{if } k \neq i) \end{cases}$$

$$\frac{\partial N_{oil,3}}{\partial N_k} = \frac{1}{1 - C_{23}} \left( C_{23} F \frac{\partial \xi_2}{\partial N_k} + \frac{\xi_2 F}{1 - C_{23}} \frac{\partial C_{23}}{\partial N_k} + \xi_2 C_{23} \frac{\partial F}{\partial N_k} \right)$$

where

$$F = \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right)$$

$$\frac{\partial F}{\partial N_k} = \begin{cases} 0 & (\text{if } k = \text{hydrocarbon component, polymer, and tracer}) \\ \frac{1}{\xi_{H_2O}} + \frac{1}{\xi_{surf}} \frac{\partial N_{surf}^f}{\partial N_{H_2O}} & (\text{if } k = H_2O) \\ \frac{1}{\xi_{surf}} \frac{\partial N_{surf}^f}{\partial N_{surf}} & (\text{if } k = \text{surfactant}) \\ \frac{1}{\xi_{surf}} \frac{\partial N_{surf}^f}{\partial N_{salt}} & (\text{if } k = \text{salt}) \end{cases}$$

for aqueous component, k

$$\frac{\partial x_{i3}}{\partial N_k} = \frac{N_i}{\sum_{i=1}^{n_c} N_i} \left( \frac{\partial N_{oil,3}}{\partial N_k} \frac{1}{N_{T,3}} - \frac{N_{oil,3}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} \right) \quad (B.69)$$

where

$$\frac{\partial N_{T,3}}{\partial N_k} = \begin{cases} \frac{\partial N_{oil,3}}{\partial N_k} & (\text{if } k = \text{hydrocarbon component}) \\ 1 + \frac{\partial N_{oil,3}}{\partial N_{H_2O}} + \frac{\partial N_{surf}^f}{\partial N_{H_2O}} + \frac{\partial N_{poly}^f}{\partial N_{H_2O}} & (\text{if } k = H_2O) \\ \frac{\partial N_{oil,3}}{\partial N_{surf}} + \frac{\partial N_{surf}^f}{\partial N_{surf}} & (\text{if } k = \text{surfactant}) \\ \frac{\partial N_{poly}^f}{\partial N_{poly}} & (\text{if } k = \text{polymer}) \\ \frac{\partial N_{oil,3}}{\partial N_{salt}} + \frac{\partial N_{surf}^f}{\partial N_{salt}} + \frac{\partial N_{poly}^f}{\partial N_{salt}} + 1 & (\text{if } k = \text{salt}) \\ 1 & (\text{if } k = \text{tracer}) \end{cases}$$

$$\frac{\partial x_{i3}}{\partial P} = \frac{N_i}{\sum_{i=1}^{n_c} N_i} \left( \frac{\partial N_{oil,3}}{\partial P} \frac{1}{N_{T,3}} - \frac{N_{oil,3}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial P} \right) \quad (B.70)$$

where

$$\frac{\partial N_{oil,3}}{\partial P} = \frac{1}{1 - C_{23}} \left( C_{23} F \frac{\partial \xi_2}{\partial P} + \frac{\xi_2 F}{1 - C_{23}} \frac{\partial C_{23}}{\partial P} + \xi_2 C_{23} \frac{\partial F}{\partial P} \right)$$

where

$$F = \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right)$$

$$\frac{\partial F}{\partial P} = -\frac{N_{H_2O}}{\xi_{H_2O}^2} \xi_{H_2O}^o C_{H_2O}^o + \frac{1}{\xi_{surf}} \frac{\partial N_{surf}^f}{\partial P} - \frac{N_{surf}^f}{\xi_{surf}^2} \xi_{surf}^o C_{surf}^o$$

$$\frac{\partial N_{T,3}}{\partial P} = \frac{\partial N_{oil,3}}{\partial P} + \frac{\partial N_{surf}^f}{\partial P} + \frac{\partial N_{poly}^f}{\partial P}$$

Equation B.66 is rewritten for aqueous component, i as follows:

For  $i = H_2O$ ,

$$x_{i3} = \frac{N_{H_2O}}{N_{T,3}} \quad (B.71)$$

$$\frac{\partial x_{i3}}{\partial N_k} = \begin{cases} \frac{1}{N_{T,3}} - \frac{N_{H_2O}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_{H_2O}} & (\text{if } k = H_2O) \\ -\frac{N_{H_2O}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = \text{others}) \end{cases} \quad (B.72)$$

$$\frac{\partial x_{i3}}{\partial P} = -\frac{N_{H_2O}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial P} \quad (B.73)$$

For  $i = \text{surfactant}$ ,

$$x_{i3} = \frac{N_{surf}^f}{N_{T,3}} \quad (B.74)$$

$$\frac{\partial x_{i3}}{\partial N_k} = \begin{cases} \frac{1}{N_{T,3}} \frac{\partial N_{surf}^f}{\partial N_k} - \frac{N_{surf}^f}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = H_2O, \text{surfactant, salt}) \\ -\frac{N_{surf}^f}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = \text{hydrocarbon component, polymer, tracer}) \end{cases} \quad (B.75)$$

$$\frac{\partial x_{i3}}{\partial P} = \frac{1}{N_{T,3}} \frac{\partial N_{surf}^f}{\partial P} - \frac{N_{surf}^f}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial P} \quad (B.76)$$

For  $i = \text{polymer}$ ,



$$x_{i3} = \frac{N_{\text{poly}}^f}{N_{T,3}} \quad (\text{B.77})$$

$$\frac{\partial x_{i3}}{\partial N_k} = \begin{cases} \frac{1}{N_{T,3}} \frac{\partial N_{\text{poly}}^f}{\partial N_k} - \frac{N_{\text{poly}}^f}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} \\ \quad (\text{if } k = \text{H}_2\text{O, polymer, salt}) \\ - \frac{N_{\text{poly}}^f}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} \\ \quad (\text{if } k = \text{hydrocarbon component, surfactant, tracer}) \end{cases} \quad (\text{B.78})$$

$$\frac{\partial x_{i3}}{\partial P} = \frac{1}{N_{T,3}} \frac{\partial N_{\text{poly}}^f}{\partial P} - \frac{N_{\text{poly}}^f}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial P} \quad (\text{B.79})$$

For  $i = \text{salt}$ ,

$$x_{i3} = \frac{N_{\text{salt}}}{N_{T,3}} \quad (\text{B.80})$$

$$\frac{\partial x_{i3}}{\partial N_k} = \begin{cases} \frac{1}{N_{T,3}} - \frac{N_{\text{salt}}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = \text{salt}) \\ - \frac{N_{\text{salt}}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = \text{others}) \end{cases} \quad (\text{B.81})$$

$$\frac{\partial x_{i3}}{\partial P} = - \frac{N_{\text{salt}}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial P} \quad (\text{B.82})$$

For  $i = \text{tracer}$ ,

$$x_{i3} = \frac{N_{\text{trac}}}{N_{T,3}} \quad (\text{B.83})$$

$$\frac{\partial x_{i3}}{\partial N_k} = \begin{cases} \frac{1}{N_{T,3}} - \frac{N_{\text{trac}}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = \text{tracer}) \\ -\frac{N_{\text{trac}}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial N_k} & (\text{if } k = \text{others}) \end{cases} \quad (\text{B.84})$$

$$\frac{\partial x_{i3}}{\partial P} = -\frac{N_{\text{trac}}}{N_{T,3}^2} \frac{\partial N_{T,3}}{\partial P} \quad (\text{B.85})$$

K. Derivatives of  $\sigma_{23}, N_{T_j}, S_{jr}, k_{rj}$

$$k_{rj}^o = k_{rj}^o \bar{S}_j^{n_j} \quad (\text{B.86})$$

where

$$k_{rj}^o = k_{rj}^{o \text{Low}} + \frac{S_{jr}^{\text{Low}} - S_{jr}^{\text{High}}}{S_{jr}^{\text{Low}} - S_{jr}^{\text{High}}} \left( k_{rj}^{o \text{High}} - k_{rj}^{o \text{Low}} \right)$$

$$\bar{S}_j = \frac{S_j - S_{jr}}{1 - S_{jr} - S_{jr}^{\text{High}}}$$

where

$$S_j = \begin{cases} \frac{1}{\xi_2} \left( \sum_{i=1}^{n_c} N_i - N_{\text{oil},3} \right) & (\text{if } j = 2) \\ \frac{1}{1 - C_{23}} \left( \frac{N_{\text{H}_2\text{O}}}{\xi_{\text{H}_2\text{O}}} + \frac{N_{\text{surf}}^f}{\xi_{\text{surf}}} \right) & (\text{if } j = 3) \end{cases} \quad (\text{see Appendix C})$$

$$S_{jr} = S_{jr}^{\text{High}} + \frac{S_{jr}^{\text{Low}} - S_{jr}^{\text{High}}}{1 + T_j N_{T_j}^{\tau_j}}$$

where

$$N_{T_j} = 2.23247 \times 10^{-8} \frac{1}{\sigma_{jj'}} \times \left\{ \left( k_x \left( -\frac{\partial P_{j'}}{\partial x} \right) \right)^2 + \left( k_y \left( -\frac{\partial P_{j'}}{\partial y} \right) \right)^2 + \left( k_z \left( -\frac{\partial P_{j'}}{\partial z} + \rho_j g \right) \right)^2 \right\}^{1/2}$$

where

$$\sigma_{jj'} = \sigma_{ow} e^{-aR_{23}} + \frac{cF_2}{R_{23}^2} (1 - e^{-aR_{23}^3})$$

where

$$F_2 = \frac{1 - e^{-\sqrt{\sum_{k=1}^3 (C_{k2} - C_{k3})^2}}}{1 - e^{-\sqrt{2}}}$$

$$n_j = n_j^{\text{Low}} + \frac{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}}{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}} (n_j^{\text{High}} - n_j^{\text{Low}})$$

$$\frac{\partial k_{rj}}{\partial V} = \frac{\partial k_{rj}^o}{\partial V} \bar{S}_j^{n_j} + k_{rj}^o \left( n_j \bar{S}_j^{n_j-1} \frac{\partial \bar{S}_j}{\partial V} + \bar{S}_j^{n_j} \ln \bar{S}_j \frac{\partial n_j}{\partial V} \right) \quad (\text{see Appendix C}) \quad (\text{B.87})$$

where

$$\frac{\partial k_{rj}^o}{\partial V} = -\frac{k_{rj}^{o\text{High}} - k_{rj}^{o\text{Low}}}{S_{j'r}^{\text{Low}} - S_{j'r}^{\text{High}}} \frac{\partial S_{j'r}}{\partial V}$$

where

$$\frac{\partial S_{jr}}{\partial V} = -\frac{\tau_j T_j N_{T_j}^{\tau_j-1} (S_{jr}^{\text{Low}} - S_{jr}^{\text{High}})}{(1 + T_j N_{T_j}^{\tau_j})^2} \frac{\partial N_{T_j}}{\partial V}$$

where

$$\frac{\partial N_{T_j}}{\partial V} = 4.9729 \times 10^{-16} \frac{g k_z^2}{\sigma_{jj'}^2 N_{T_j}} \left( \rho_j g - \frac{\partial P_{j'}}{\partial z} \right) \frac{\partial \rho_j}{\partial V} - \frac{N_{T_j}}{\sigma_{jj'}} \frac{\partial \sigma_{jj'}}{\partial V}$$

where

$$\frac{\partial \rho_j}{\partial V} = \frac{\partial (\xi_j \bar{M}_j)}{\partial V} = \frac{\partial \xi_j}{\partial V} \bar{M}_j + \xi_j \frac{\partial \bar{M}_j}{\partial V}$$

where

$$\frac{\partial \bar{M}_j}{\partial V} = \frac{\partial}{\partial V} \left( \sum_{i=1}^{n_c + n_a + 1} x_{ij} M_i \right) = \sum_{i=1}^{n_c + n_a + 1} \left( \frac{\partial x_{ij}}{\partial V} M_i \right)$$

$$\begin{aligned} \frac{\partial \sigma_{jj'}}{\partial V} &= \frac{c}{R_{23}^2} \left( 1 - e^{-a R_{23}^3} \right) \frac{\partial F_2}{\partial V} \\ &+ \left[ -\sigma_{ow} a e^{-a R_{23}^3} + c F_2 \left\{ 3 a e^{-a R_{23}^3} - \frac{2}{R_{23}^3} \left( 1 - e^{-a R_{23}^3} \right) \right\} \right] \frac{\partial R_{23}}{\partial V} \end{aligned}$$

where

$$\frac{\partial F_2}{\partial V} = \frac{1}{1 - e^{-\sqrt{2}}} e^{-f} \frac{\partial f}{\partial V}$$

where

$$f = \sqrt{\sum_{k=1}^3 (C_{k2} - C_{k3})^2}$$

$$\frac{\partial f}{\partial V} = \frac{\partial}{\partial V} (\sqrt{g}) = \frac{1}{2\sqrt{g}} \frac{\partial g}{\partial V}$$

where

$$g = \sum_{k=1}^3 (C_{k2} - C_{k3})^2 \quad (C_{12} = C_{32} = 0, C_{22} = 1)$$

$$\frac{\partial g}{\partial V} = 2C_{13} \frac{\partial C_{13}}{\partial V} - 2(1 - C_{23}) \frac{\partial C_{23}}{\partial V} + 2C_{33} \frac{\partial C_{33}}{\partial V}$$

$$\frac{\partial \bar{S}_j}{\partial V} = \frac{1}{(1 - S_{jr} - S_{jr'})} \left( \frac{\partial S_j}{\partial V} - \frac{\partial S_{jr}}{\partial V} \right) + \frac{S_j - S_{jr}}{(1 - S_{jr} - S_{jr'})^2} \left( \frac{\partial S_{jr'}}{\partial V} + \frac{\partial S_{jr}}{\partial V} \right)$$

where

$$\frac{\partial S_j}{\partial V} = -\frac{1}{\xi_2^2} \frac{\partial \xi_2}{\partial V} \left( \sum_{i=1}^{n_c} N_i - N_{oil,3} \right) + \frac{1}{\xi_2} \left( \frac{\partial}{\partial V} \left( \sum_{i=1}^{n_c} N_i \right) - \frac{\partial N_{oil,3}}{\partial V} \right) \text{ (if } j = 2)$$

where

$$\frac{\partial}{\partial V} \left( \sum_{i=1}^{n_c} N_i \right) = \begin{cases} 1 & \text{(if } V = N_i \text{ (for hydrocarbon component, } i)) \\ 0 & \text{(for others)} \end{cases}$$

$$\frac{\partial S_j}{\partial V} = \frac{F}{(1 - C_{23})^2} \frac{\partial C_{23}}{\partial V} + \frac{1}{1 - C_{23}} \frac{\partial F}{\partial V} \text{ (if } j = 3)$$

where

$$F = \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}}$$

$$\frac{\partial F}{\partial V} = \frac{\partial N_{H_2O}}{\partial V} \frac{1}{\xi_{H_2O}} - \frac{N_{H_2O}}{\xi_{H_2O}^2} \frac{\partial \xi_{H_2O}}{\partial V} + \frac{\partial N_{surf}^f}{\partial V} \frac{1}{\xi_{surf}} - \frac{N_{surf}^f}{\xi_{surf}^2} \frac{\partial \xi_{surf}}{\partial V}$$

where

$$\frac{\partial \xi_i}{\partial V} = \begin{cases} 0 & \text{(if } V = N_j) \\ \xi_i^0 C_i^0 & \text{(if } V = P) \end{cases}$$

$$\frac{\partial n_j}{\partial V} = -\frac{n_j^{High} - n_j^{Low}}{S_{jr}^{Low} - S_{jr}^{High}} \frac{\partial S_{jr}}{\partial V}$$

L. Derivatives of  $\mu_j$

$$\mu_2 = f(P, T, x_{i2}) \quad (\text{B.88})$$

where

$$x_{i2} = \frac{N_i}{\sum_{i=1}^{n_c} N_i}$$

$$\mu_3 = C_{13}\mu_p e^{\alpha_1(C_{23}+C_{33})} + C_{23}\mu_2 e^{\alpha_2(C_{13}+C_{33})} + C_{33}\alpha_3 e^{(\alpha_4 C_{13} + \alpha_5 C_{23})} \quad (\text{B.89})$$

where

$$\begin{aligned} \mu_p &= \mu_w \left( 1 + \left( A_{p1}C_{43} + A_{p2}C_{43}^2 + A_{p3}C_{43}^3 \right) C_{SE}^{S_p} \right) \\ \frac{\partial \mu_3}{\partial V} &= C_{13} e^{\alpha_1(C_{23}+C_{33})} \frac{\partial \mu_p}{\partial V} \\ &+ \left( \mu_p e^{\alpha_1(C_{23}+C_{33})} + C_{23}\alpha_2\mu_2 e^{(\alpha_2 C_{33} + \alpha_2 C_{13})} + C_{33}\alpha_3\alpha_4 e^{(\alpha_5 C_{23} + \alpha_4 C_{13})} \right) \frac{\partial C_{13}}{\partial V} \\ &+ \left( \mu_p C_{13}\alpha_1 e^{\alpha_1(C_{23}+C_{33})} + \mu_2 e^{\alpha_2(C_{13}+C_{33})} + C_{33}\alpha_3\alpha_5 e^{(\alpha_4 C_{13} + \alpha_5 C_{23})} \right) \frac{\partial C_{23}}{\partial V} \\ &+ \left( \mu_p C_{13}\alpha_1 e^{\alpha_1(C_{23}+C_{33})} + \mu_2 C_{23}\alpha_2 e^{\alpha_2(C_{13}+C_{33})} + \alpha_3 e^{(\alpha_4 C_{13} + \alpha_5 C_{23})} \right) \frac{\partial C_{33}}{\partial V} \\ &+ C_{23} e^{\alpha_2(C_{13}+C_{33})} \frac{\partial \mu_2}{\partial V} \end{aligned} \quad (\text{B.90})$$

where

$$\frac{\partial \mu_p}{\partial V} = \begin{cases} \text{if } S_p \neq 0 \\ \mu_w C_{SE}^{S_p-1} \left[ C_{SE} \left( A_{p1} + 2A_{p2}C_{43} + 3A_{p3}C_{43}^2 \right) \frac{\partial C_{43}}{\partial V} \right. \\ \left. + S_p \left( A_{p1}C_{43} + A_{p2}C_{43}^2 + A_{p3}C_{43}^3 \right) \frac{\partial C_{SE}}{\partial V} \right] \\ \text{if } S_p = 0 \\ \mu_w \left( A_{p1} + 2A_{p2}C_{43} + 3A_{p3}C_{43}^2 \right) \frac{\partial C_{43}}{\partial V} \end{cases}$$

## Derivatives for Source/Sink Term

For a constant rate injection,

For  $i = \text{H}_2\text{O}$ ,

$$\frac{\partial q_i}{\partial V} = \frac{q_{\text{H}_2\text{O}}|_{\text{surface}}}{A^2} (A - a) \frac{\partial a}{\partial V} \quad (\text{B.91})$$

where

$$A = \sum_{m=z_t}^{z_b} (\text{PI}_t)_m$$

$$a = (\text{PI}_t)_z$$

$$\frac{\partial a}{\partial V} = c_z \sum_{j=2}^3 \left( \frac{1}{\mu_j} \frac{\partial k_{rj}}{\partial V} - \frac{k_{rj}}{\mu_j^2} \frac{\partial \mu_j}{\partial V} \right)$$

For  $i = \text{surfactant}$ ,

$$\frac{\partial q_i}{\partial V} = c_{i,\text{inj}} \frac{\xi_{\text{surf}}}{\xi_{\text{H}_2\text{O}}|_{\text{surface}}} \frac{\partial q_{\text{H}_2\text{O}}}{\partial V} \quad (\text{B.92})$$

For  $i = \text{polymer}$ ,

$$\frac{\partial q_i}{\partial V} = \frac{c_{i,\text{inj}}}{100} \frac{M_{\text{H}_2\text{O}}}{M_{\text{poly}}} \frac{\partial q_{\text{H}_2\text{O}}}{\partial V} \quad (\text{B.93})$$

if  $i = \text{salt or tracer}$ ,

$$\frac{\partial q_i}{\partial V} = \frac{c_{i,\text{inj}}}{16.037 \xi_{\text{H}_2\text{O}}} \frac{\partial q_{\text{H}_2\text{O}}}{\partial V} \quad (\text{B.94})$$

For a constant pressure injection well,

For  $i = \text{H}_2\text{O}$ ,

$$\frac{\partial q_i}{\partial V} = \begin{cases} \frac{c_z}{1 + c_{\text{surf}, \text{inj}}} \bullet \xi_{\text{H}_2\text{O}} \Big|_{\text{inj}} (P_{\text{wf}} - P) \left( \frac{1}{\mu_3} \frac{\partial k_{r3}}{\partial V} - \frac{k_{r3}}{\mu_3^2} \frac{\partial \mu_3}{\partial V} \right) & (\text{if } V \neq P) \\ \frac{c_z}{1 + c_{\text{surf}, \text{inj}}} \bullet \xi_{\text{H}_2\text{O}} \Big|_{\text{inj}} \left( \left( \frac{1}{\mu_3} \frac{\partial k_{r3}}{\partial V} - \frac{k_{r3}}{\mu_3^2} \frac{\partial \mu_3}{\partial V} \right) (P_{\text{wf}} - P) - \frac{k_{r3}}{\mu_3} \right) & (\text{if } V = P) \end{cases} \quad (\text{B.95})$$

For  $i = \text{surfactant}$ ,

$$\frac{\partial q_i}{\partial V} = c_{i, \text{inj}} \frac{\xi_{\text{surf}}}{\xi_{\text{H}_2\text{O}}} \Big|_{\text{surface}} \frac{\partial q_{\text{H}_2\text{O}}}{\partial V} \quad (\text{B.96})$$

For  $i = \text{polymer}$ ,

$$\frac{\partial q_i}{\partial V} = \frac{c_{i, \text{inj}}}{100} \frac{M_{\text{H}_2\text{O}}}{M_{\text{poly}}} \frac{\partial q_{\text{H}_2\text{O}}}{\partial V} \quad (\text{B.97})$$

For  $i = \text{salt or tracer}$ ,

$$\frac{\partial q_i}{\partial V} = \frac{c_{i, \text{inj}}}{16.037 \xi_{\text{H}_2\text{O}}} \frac{\partial q_{\text{H}_2\text{O}}}{\partial V} \quad (\text{B.98})$$

For a constant pressure production well,

$$\frac{\partial q_i}{\partial V} = \begin{cases} c_z (P_{\text{wf}} - P) \sum_{j=2}^3 \frac{\partial}{\partial V} \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right) & (\text{if } V \neq P) \\ c_z \sum_{j=2}^3 \left( (P_{\text{wf}} - P) \frac{\partial}{\partial V} \left( \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right) - \xi_j \frac{k_{rj}}{\mu_j} x_{ij} \right) & (\text{if } V = P) \end{cases} \quad (\text{B.99})$$



## APPENDIX C—Derivations of the Equations in Appendix B

1)

$$C_i^f = \frac{\tilde{C}_1}{2b_i} \left( -a_i - 1 + b_i \frac{\tilde{C}_i}{\tilde{C}_1} + \sqrt{\left( a_i + 1 - b_i \frac{\tilde{C}_i}{\tilde{C}_1} \right)^2 + 4b_i \left( \frac{\tilde{C}_i}{\tilde{C}_1} \right)} \right) \quad (\text{if } i=3 \text{ (surf) or } 4 \text{ (poly)}) \quad (\text{B.25})$$

### Derivation

The adsorbed concentration of i component is given by a Langmuir-type isotherm,

$$\frac{\hat{C}_i}{\tilde{C}_1} = \frac{a_i \left( \frac{\tilde{C}_i}{\tilde{C}_1} - \frac{\hat{C}_i}{\tilde{C}_1} \right)}{1 + b_i \left( \frac{\tilde{C}_i}{\tilde{C}_1} - \frac{\hat{C}_i}{\tilde{C}_1} \right)} \quad (\text{C.1})$$

where

$$a_i = (a_{i1} + a_{i2}C_{SE})$$

We can rewrite Eq. C.1 as follows.

$$\frac{\tilde{C}_i}{\tilde{C}_1} - \frac{C_i^f}{\tilde{C}_1} = \frac{a_i \frac{C_i^f}{\tilde{C}_1}}{1 + b_i \frac{C_i^f}{\tilde{C}_1}} \quad (\text{C.2})$$

Let

$$\frac{C_i^f}{\tilde{C}_1} = X, \text{ and } \frac{\tilde{C}_i}{\tilde{C}_1} = \alpha$$

Then, Eq. C.2 can be rewritten as

$$b_i X^2 + (a_i + 1 - \alpha b_i) X - \alpha = 0 \quad (\text{C.3})$$

Equation C.3 has one physically reasonable root, which is

$$X = \frac{1}{2b_i} \left( -a_i - 1 + \alpha b_i + \sqrt{(a_i + 1 - \alpha b_i)^2 + 4\alpha b_i} \right) \quad (C.4)$$

Equation C.4 can be rewritten using the definition of X and  $\alpha$  as

$$C_i^f = \frac{\tilde{C}_1}{2b_i} \left( -a_i - 1 + b_i \frac{\tilde{C}_i}{\tilde{C}_1} + \sqrt{\left( a_i + 1 - b_i \frac{\tilde{C}_i}{\tilde{C}_1} \right)^2 + 4b_i \left( \frac{\tilde{C}_i}{\tilde{C}_1} \right)} \right) \quad (B.25)$$

2)

$$x_{i2} = \frac{N_i}{\sum_{i=1}^{n_c} N_i} \quad (\text{for hydrocarbon component, } i)$$

#### Derivation

By definition,

$$x_{i3} = \frac{N_{i,3}}{N_{T,3}} \quad (C.5)$$

where

$N_{i,j}$ : mole/PV of hydrocarbon component  $i$  in phase  $j$

$$N_{T,3} = N_{H_2O} + N_{oil,3} + N_{surf}^f + N_{poly}^f + N_{salt} + N_{trac}$$

where

$N_{oil,j}$ : total mole/PV of hydrocarbon components in phase  $j$

Assume that

$$\frac{N_{i,2}}{N_{oil,2}} = \frac{N_{i,3}}{N_{oil,3}} \quad (C.6)$$

where

$$N_{oil,j} = \sum_{i=1}^{n_c} N_{i,j}$$

Eq. C.6 is rewritten as,

$$N_{i,2}N_{oil,3} = N_{i,3}N_{oil,2} \quad (C.7)$$

and we know that

$$N_i = N_{i,2} + N_{i,3} \quad (C.8)$$

From Equations C.7 and C.8,

$$N_{i,2} = \frac{N_i N_{oil,2}}{(N_{oil,2} + N_{oil,3})} = \frac{N_i N_{oil,2}}{\sum_{i=1}^{n_c} N_i} \quad (C.9)$$

Therefore,

$$x_{i2} = \frac{N_{i,2}}{N_{oil,2}} = \frac{N_i}{\sum_{i=1}^{n_c} N_i} \quad (C.10)$$

**3)**

$$N_{oil,3} = \frac{\xi_2 C_{23}}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right)$$

Derivation

$$\begin{aligned}
C_{23} &= \frac{V_{\text{oil},3}}{V_{\text{oil},3} + V_{\text{H}_2\text{O}} + V_{\text{surf},3}} \\
&= \frac{\frac{n_{\text{oil},3}}{\xi_{\text{oil},3}}}{\frac{n_{\text{oil},3}}{\xi_{\text{oil},3}} + \frac{n_{\text{H}_2\text{O}}}{\xi_{\text{H}_2\text{O}}} + \frac{n_{\text{surf}}^f}{\xi_{\text{surf}}}} \\
&= \frac{\frac{N_{\text{oil},3}}{\xi_{\text{oil},3}}}{\frac{N_{\text{oil},3}}{\xi_{\text{oil},3}} + \frac{N_{\text{H}_2\text{O}}}{\xi_{\text{H}_2\text{O}}} + \frac{N_{\text{surf}}^f}{\xi_{\text{surf}}}}
\end{aligned} \tag{C.11}$$

where

$V_{\text{oil},j}$  : bulk volume of hydrocarbon components in phase j

$V_{\text{surf},j}$  : bulk volume of surfactant in phase j

$n_{\text{oil},j}$  : total mole of hydrocarbon components in phase j

$\xi_{\text{oil},j}$  : bulk molar density of hydrocarbon components in phase j

Rewrite Eq. C.11,

$$\left( \frac{1}{\xi_{\text{oil},3}} - \frac{C_{23}}{\xi_{\text{oil},3}} \right) N_{\text{oil},3} = C_{23} \left( \frac{N_{\text{H}_2\text{O}}}{\xi_{\text{H}_2\text{O}}} + \frac{N_{\text{surf}}^f}{\xi_{\text{surf}}} \right) \tag{C.12}$$

Eq. C.12 is rewritten as,

$$N_{\text{oil},3} = \frac{\xi_{\text{oil},3} C_{23}}{1 - C_{23}} \left( \frac{N_{\text{H}_2\text{O}}}{\xi_{\text{H}_2\text{O}}} + \frac{N_{\text{surf}}^f}{\xi_{\text{surf}}} \right) \tag{C.13}$$

Assume that

$$\xi_{\text{oil},3} \approx \xi_2 \tag{C.14}$$

Then,

$$N_{oil,3} = \frac{\xi_2 C_{23}}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right) \quad (C.15)$$

4)

$$S_j = \begin{cases} \frac{1}{\xi_2} \left( \sum_{i=1}^{n_c} N_i - N_{oil,3} \right) & (\text{if } j = 2) \\ \frac{1}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right) & (\text{if } j = 3) \end{cases}$$

Derivation

$$S_2 = \frac{V_{oil,2}}{V_p} = \frac{N_{oil,2}}{\xi_2} = \frac{1}{\xi_2} \left( \sum_{i=1}^{n_c} N_i - N_{oil,3} \right) \quad (C.16)$$

$$\begin{aligned} S_3 &= \frac{1}{V_p} (V_{oil,3} + V_{H_2O} + V_{surf,3}) = \frac{N_{oil,3}}{\xi_{oil,3}} + \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \\ &= \frac{1}{\xi_{oil,3}} \frac{\xi_2 C_{23}}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right) + \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \end{aligned} \quad (C.17)$$

Using Eq. C.14,

$$S_3 = \frac{(C_{23} + 1 - C_{23})}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right) = \frac{1}{1 - C_{23}} \left( \frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}^f}{\xi_{surf}} \right) \quad (C.18)$$

5)

$$\frac{\partial k_{rj}}{\partial V} = \frac{\partial k_{rj}^o}{\partial V} \bar{S}_j^{n_j} + k_{rj}^o \left( n_j \bar{S}_j^{n_j-1} \frac{\partial \bar{S}_j}{\partial V} + \bar{S}_j^{n_j} \ln \bar{S}_j \frac{\partial n_j}{\partial V} \right) \quad (B.87)$$

Derivation

$$k_{rj} = k_{rj}^o \bar{S}_j^{n_j} \quad (B.86)$$

$$\frac{\partial k_{rj}}{\partial V} = \frac{\partial k_{rj}^o}{\partial V} \bar{S}_j^{n_j} + k_{rj}^o \frac{\partial \left( \bar{S}_j^{n_j} \right)}{\partial V} \quad (C.19)$$

To calculate  $\frac{\partial \left( \bar{S}_j^{n_j} \right)}{\partial V}$  in (A.19), let  $F = \bar{S}_j^{n_j}$  and apply natural log to both sides, then

$$\ln F = n_j \ln \bar{S}_j \quad (C.20)$$

$$\frac{\partial}{\partial V} \ln F = \frac{1}{F} \frac{\partial F}{\partial V} = \frac{\partial n_j}{\partial V} \ln \bar{S}_j + n_j \frac{1}{\bar{S}_j} \frac{\partial \bar{S}_j}{\partial V} \quad (C.21)$$

$$\begin{aligned} \therefore \frac{\partial F}{\partial V} &= \frac{\partial \bar{S}_j^{n_j}}{\partial V} \\ &= F \left( \frac{\partial n_j}{\partial V} \ln \bar{S}_j + n_j \frac{1}{\bar{S}_j} \frac{\partial \bar{S}_j}{\partial V} \right) \\ &= \bar{S}_j^{n_j} \left( \frac{\partial n_j}{\partial V} \ln \bar{S}_j + n_j \frac{1}{\bar{S}_j} \frac{\partial \bar{S}_j}{\partial V} \right) \\ &= n_j \bar{S}_j^{n_j-1} \frac{\partial \bar{S}_j}{\partial V} + \bar{S}_j^{n_j} \ln \bar{S}_j \frac{\partial n_j}{\partial V} \end{aligned} \quad (C.22)$$

$$\therefore \frac{\partial k_{rj}}{\partial V} = \frac{\partial k_{rj}^o}{\partial V} \bar{S}_j^{n_j} + k_{rj}^o \left( n_j \bar{S}_j^{n_j-1} \frac{\partial \bar{S}_j}{\partial V} + \bar{S}_j^{n_j} \ln \bar{S}_j \frac{\partial n_j}{\partial V} \right) \quad (B.87)$$

## APPENDIX D—3-D Surfactant/Polymer Input Data

```
TITLE(2)="96x96x11 (101,376 gridblocks) run"

DESCRIPTION( )=
"THICKNESS (FT) : 66 "
"LENGTH (FT) : 2400 "
"WIDTH (FT) : 2400 "
"GRID BLOCKS : 96X96x11"

COMPOSITIONAL_MODEL

TIMEEND = 5000

$ I/O OPTIONS
OUTLEVEL = 1
PROCOUT
OUTPUT_PRE
OUTPUT_SAT
OUTPUT_VIS
OUTPUT_AQ
OUTPUT_IFT
OUTPUT_HIS
TDPVOPT

OUTPUT_TIME( ) = 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
                  1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0

$ FAULT BLOCK AND MESH DATA
METHOD = 2
DOWN( ) = 0 0 1
NX(1) = 96  NY(1) = 96  NZ(1) = 11
DX( ) = 25  DY( ) = 25  DZ( ) = 6

$ COMPOUND NAMES
COMPOUND(1) = "C10"

$ COMPOUND CRITICAL TEMPERATURES
CRIT( ) 1111.8

$ COMPOUND CRITICAL PRESSURES
CRIP( ) 304.0

$ COMPOUND CRITICAL VOLUMES
CRIV( ) 12.087

$ COMPOUND ACEN
ACEN( ) 0.488

$ COMPOUND MOL WEIGHTS
MOLW( ) 142.3

$ COMPOUND PARA
PARA( ) 431.0

$ MAX NUMBER OF PHASES
NPHASE = 3

$ Initial rock & water properties
ROCKZ = 0.00000  ROCKP = 14.7
H2OZ = 0.000003  H2OP = 14.7  H2OD = 3.467
```

```

SURTF = 60.0  SURPS = 14.7
RESTF = 130.0

$ TOLERANCE
TOL_FLASH = 0.0001
TOL_VOLUME = 0.0001
TOL_MASS = 0.0001
TOL_WATER = 0.0001
MAXNEWT = 100

$ POROSITY
POROSITY1() = 0.336

$ PERMEABILITIES
XPERM1() = 540
YPERM1() = 540
ZPERM1() = 540

$ INITIAL WATER SATURATION
SWINI1() = 0.70

$ INITIAL WATER CELL PRESSURE
PINI1() = 900.0

$ INITIAL PHASE VISCOSITIES AT EACH CELL
VIS1() = 0.75

$ INITIAL COMPOSITIONS
ZXY1(,,1) = 1.000

$TRACER DATA
IOILVIS
OILVIS 2.6

IMPAQCOMP
NAQCOMP 3
AQCOMPNAM() = "SALT" "SURFACTANT" "POLYMER"
AQCOMPTYPE() = 4 3 2
AQCOMPINIT() = 0.17 0.00 0.00

$ SURFACTANT PARAMS
EPSME 0.0001
HBNC70 0.07
HBNC71 0.04
CSEL7 0.177
CSEU7 0.250

$ IFT PARAMS
AHUH 9
CHUH 0.2
XIFTW 1.3

$POLYMER
AP1 81
AP2 2500
AP3 2700
AD41 0
SLOPP 0

$ ADSORPTION PARAMETERS
AD31 1.5
AD32 0.5
B3D 1000

```



```

$ RELPERM DATA
RELPERM 2
NRELPERM 1
ITRAP
XIFTRW = 1.3
ENDPTLOW() = 0.106 0.800 0.000
ENDPTHIGH() = 1.000 1.000 0.000
SRLOW() = 0.14 0.25 0.00
SRHIGH() = 0.00 0.00 0.00
EXPLOW() = 2.10 1.70 0.00
EXPHIGH() = 0.48 1.50 0.00
TL() = 364.2 59074 364.2
TAUL() = 1 1 1

NUMWELL=13

WELLNAME(1) = "INJ1"
KINDWELL(1) = 2
WELLTOP(1 TO 3,1,1) = 12 12 0
WELLBOTTOM(1 TO 3,1,1) = 12 12 100
DIAMETER(1,1) = 1.0
PRLIMIT(1) = 2500
WELLPQ(1) Block
Interpolation Linear
Extrapolation Constant
Data 0. 1000
EndBlock

WELLNAME(2) = "INJ2"
KINDWELL(2) = 2
WELLTOP(1 TO 3,1,2) = 1188 12 0
WELLBOTTOM(1 TO 3,1,2) = 1188 12 100
DIAMETER(1,2) = 1.0
PRLIMIT(2) = 2500
WELLPQ(2) Block
Interpolation Linear
Extrapolation Constant
Data 0. 1000
EndBlock

WELLNAME(3) = "INJ3"
KINDWELL(3) = 2
WELLTOP(1 TO 3,1,3) = 2388 12 0
WELLBOTTOM(1 TO 3,1,3) = 2388 12 100
DIAMETER(1,3) = 1.0
PRLIMIT(3) = 2500
WELLPQ(3) Block
Interpolation Linear
Extrapolation Constant
Data 0. 1000
EndBlock

WELLNAME(4) = "PROD5"
KINDWELL(4) = 3
WELLTOP(1 TO 3,1,4) = 612 612 0
WELLBOTTOM(1 TO 3,1,4) = 612 612 100
DIAMETER(1,4) = 1.0
WELLPQ(4) Block
Interpolation Linear
Extrapolation Constant
Data 0. 500
EndBlock

WELLNAME(5) = "PROD6"

```

```

KINDWELL(5) = 3
WELLTOP(1 TO 3,1,5) = 1812 612 0
WELLBOTTOM(1 TO 3,1,5) = 1812 612 100
DIAMETER(1,5) = 1.0
WELLPQ(5) Block
Interpolation Linear
Extrapolation Constant
Data 0. 500
EndBlock

WELLNAME(6) = "INJ8"
KINDWELL(6) = 2
WELLTOP(1 TO 3,1,6) = 12 1188 0
WELLBOTTOM(1 TO 3,1,6) = 12 1188 100
DIAMETER(1,6) = 1.0
PRLIMIT(6) = 2500
WELLPQ(6) Block
Interpolation Linear
Extrapolation Constant
Data 0. 1000
EndBlock

WELLNAME(7) = "INJ9"
KINDWELL(7) = 2
WELLTOP(1 TO 3,1,7) = 1188 1188 0
WELLBOTTOM(1 TO 3,1,7) = 1188 1188 100
DIAMETER(1,7) = 1.0
PRLIMIT(7) = 2500
WELLPQ(7) Block
Interpolation Linear
Extrapolation Constant
Data 0. 1000
EndBlock

WELLNAME(8) = "INJ10"
KINDWELL(8) = 2
WELLTOP(1 TO 3,1,8) = 2388 1188 0
WELLBOTTOM(1 TO 3,1,8) = 2388 1188 100
DIAMETER(1,8) = 1.0
PRLIMIT(8) = 2500
WELLPQ(8) Block
Interpolation Linear
Extrapolation Constant
Data 0. 1000
EndBlock

WELLNAME(9) = "PROD12"
KINDWELL(9) = 3
WELLTOP(1 TO 3,1,9) = 612 1812 0
WELLBOTTOM(1 TO 3,1,9) = 612 1812 100
DIAMETER(1,9) = 1.0
WELLPQ(9) Block
Interpolation Linear
Extrapolation Constant
Data 0. 500
EndBlock

WELLNAME(10) = "PROD13"
KINDWELL(10) = 3
WELLTOP(1 TO 3,1,10) = 1812 1812 0
WELLBOTTOM(1 TO 3,1,10) = 1812 1812 100
DIAMETER(1,10) = 1.0
WELLPQ(10) Block
Interpolation Linear

```

Extrapolation Constant  
Data 0. 500  
EndBlock

WELLNAME(11) = "INJ15"  
KINDWELL(11) = 2  
WELLTOP(1 TO 3,1,11) = 12 2388 0  
WELLBOTTOM(1 TO 3,1,11) = 12 2388 100  
DIAMETER(1,11) = 1.0  
PRLIMIT(11) = 2500  
WELLPQ(11) Block  
Interpolation Linear  
Extrapolation Constant  
Data 0. 1000  
EndBlock

WELLNAME(12) = "INJ16"  
KINDWELL(12) = 2  
WELLTOP(1 TO 3,1,12) = 1188 2388 0  
WELLBOTTOM(1 TO 3,1,12) = 1188 2388 100  
DIAMETER(1,12) = 1.0  
PRLIMIT(12) = 2500  
WELLPQ(12) Block  
Interpolation Linear  
Extrapolation Constant  
Data 0. 1000  
EndBlock

WELLNAME(13) = "INJ17"  
KINDWELL(13) = 2  
WELLTOP(1 TO 3,1,13) = 2388 2388 0  
WELLBOTTOM(1 TO 3,1,13) = 2388 2388 100