DE-FC-26-00BC15314

A NEW GENERATION CHEMICAL FLOODING SIMULATOR

Semi-annual Report for the Period
April 1, 2002 – September 30, 2002

By

Gary A. Pope, Kamy Sepehrnoori, and Mojdeh Delshad

Center for Petroleum and Geosystems Engineering

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

A NEW GENERATION CHEMICAL FLOODING SIMULATOR

Contract No. DE-FC-26-00BC15314

The University of Texas
Austin, TX 78712

Contract Date: October 1, 2001
Anticipated Completion: August 31, 2004
Government Award: \$930,801 (Total funding)

Principal Investigators:
Gary A. Pope
Kamy Sepehrnoori
Mojdeh Delshad

Contract Officer:
Richard Rogus
National Petroleum Technology Office

Reporting Period: April 1, 2002- September 30, 2002

TABLE OF CONTENTS

	Page
ABSTRACT	4
SUMMARY	4
Task 1: Formulation and development of Solution Scheme	6
Task 2: Formulation and Implementation of Chemical Module	35
Task 3: Validation and Application	50
REFERENCES	63
TABLES	65
FIGURES	84
Appendix A	100

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 computing 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.

SUMMARY

During the past several years, an advanced and efficient reservoir simulation framework named IPARS, Integrated Parallel Accurate Reservoir Simulator (Parashar *et al.*, 1997) has been developed by the research team at the University of Texas. IPARS includes a number of advanced features such as providing all the necessary infrastructure for physical models, from message passing and input/output to solvers and well handling, run on a range of platforms, from a single PC with Linux Operating System to massively parallel machines or clusters of workstation. The framework supports isothermal, three-dimensional, multiple phases containing multiple species plus an immobile solid rock phase with adsorbing components. Phase densities and viscosities may be arbitrary functions of pressure and composition. Porosity and permeability may vary with location. Physical properties such as relative permeability and capillary pressure are functions of saturations and rock type. An arbitrary number of wells may be completed at arbitrary locations. Finite difference formulations with a block-centered grid are assumed 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 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).

The goal of this project is 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 as illustrated in **Figure 1.1**. The progress in the second half of the first year was focused on the implementation of the chemical module reported in Task 2 and the validation of chemical module reported under Task 3. In this report, we will detail our progress on Tasks 1 through 3 for the first year of the project.

Task 1: Formulation and Development of Solution Scheme

The effort on this task was directed towards the formulation of tracer and polymer options. For completeness, we first give a brief review of the mathematical formulation of the compositional model. 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 CO₂, N₂ or H₂S.
- Instantaneous local thermodynamic equilibrium between hydrocarbon phases.
- Negligible capillary pressure effects on hydrocarbon phase equilibrium.
- Water is slightly compressible and water viscosity is constant.

Mass conservation equation

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

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

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

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

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 Equation 3 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 ϕ is the porosity, ξ_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:

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

Multiplying both sides of Equation 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

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

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 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 μ_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 ΔG 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

$$\ln Y_i + \ln \phi_i(\bar{Y}) - h_i = 0 \quad (1.10)$$

for $i = 1, \dots, n_c$

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

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

$$h_i = \ln Z_i + \ln \phi_i(\bar{Z}) \quad (1.12)$$

for $i = 1, \dots, n_c$

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 says 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 is 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

$$f_{ij} = f_{il} \quad \text{for } i = 1, \dots, n_c \quad (1.13)$$

$$j = 2, \dots, n_p (j \neq l)$$

where phase l 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,

$$f_{ij} = f_{ij}(P, \overline{x_j}) \quad \text{for } i = 1, \dots, n_c \quad (1.14)$$

$$j = 2, \dots, n_p (j \neq l)$$

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

$$x_{ij} = \frac{n_{ij}}{n_j} \quad (1.16)$$

for $i = 1, 2, \dots, n_c, j = 1, 2, \dots, n_p$

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

$$v_l = \frac{1}{1 - K_{\max}} < 0 \quad (1.20)$$

$$v_r = \frac{1}{1 - K_{\min}} > 0$$

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 Equations **1.18** and **1.19** for v . However, round-off errors occur when solving Equation **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 Equation **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

$$v_{\min} = \max \left(\frac{Z_i K_i - 1}{K_i - 1} \right) \quad (1.22)$$

for $K_i > 1$ and $Z_i < 0$

$$v_{\max} = \min \left(\frac{Z_i - 1}{K_i - 1} \right) \quad (1.23)$$

for $K_i < 1$ and $Z_i > 0$

Also, the Newton procedure is used to solve Equation **1.21** for v .

Equation of State

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

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

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

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

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

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

$$\kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2, \quad (1.28)$$

if $\omega < 0.49$

$$\kappa = 0.379640 + 1.485030\omega - 0.164423\omega^2 + 0.016666\omega^3 \quad (1.29)$$

if $\omega \geq 0.49$

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

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

and

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

where for component i , the a_i is computed from Equation 1.25, and b_i is computed from Equation 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.* (1996). 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

$$\Omega_a = 0.457235529$$

$$\Omega_b = 0.077796074$$

Dividing Equation 1.37 by Equation 1.38 gives:

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

where α is defined in Equation 1.26.

A fluid is assumed to be in single-phase if $T > T_c$, which also implies

$\alpha \leq 1$. From Equation 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

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

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, $\overline{v_j}$ 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,

$$P_{cref,j} = P_{cref,j}(\vec{S}, \vec{x}) \quad (1.45)$$

for $j = 1, \dots, n_p$

Formation porosity

The porosity is a function of pressure.

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

where ϕ_{ref} is the porosity at the reference pressure, P_{ref} and ϕ 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 combined several viscosity correlations. The steps involved in calculating the phase viscosity are given below:

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

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

or

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

where

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

- 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 ψ_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

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

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 solution approach is identical to that outlined earlier in the solution procedure with the exception that 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, 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 Equation 1.54 by Equation 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, Δx , Δy and Δz are gridblock sizes (ft) in x, y and z directions, λ_{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 Equation 1.59, the radius r_o is given by Peaceman's equation:

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

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 γ_{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

$$(q_i)_z = [1 - (f_1)_{inj}] (z_i)_{inj} (q_t)_z \quad (1.62)$$

for $i = 1, \dots, n_c$

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

and

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

where

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

And

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

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

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

where ξ_1^o is the aqueous phase molar density at the reference pressure P_1^o .

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

$$(q_i)_z = \sum_{j=2}^{n_p} (\xi_j x_{ij} PI_j)_z (P_{wf} - P_j)_z \quad (1.69)$$

for $i = 1, \dots, n_c$

And

$$(q_{n_c+1})_z = (\xi_1 PI_1)_z (P_{wf} - P_1)_z \quad (1.70)$$

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 Jacobian term 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, 1998; 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⁺⁺. 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 lead to the improvement in the performance of the matrix routines.

Task 2: Formulation and Implementation of Chemical Module

The first set of chemical species added to the GPAS simulator is the tracer. Thus, although all the formulation in the aqueous species modeling applies to tracer, polymer and surfactant, each of these species modeling can involve, in addition, its own assumptions, formulations and special properties. Henceforth, the aqueous species refers to a species present in trace quantities in the aqueous phase and this includes a tracer, polymer or surfactant, the main subroutine XAQCOMP solving for the aqueous species concentration as a function of space and time is referred as the chemical subroutine and the whole module encompassing all the separate subroutines modeling the tracer, polymer and surfactant features is referred as the chemical module. The final output of the chemical subroutine XAQCOMP is the dimensionless aqueous species concentration.

In GPAS simulator, 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.

Maroongroge (1994) used the following approach for tracer calculations in UTCOMP:

- Solution of the pressure equation.
- Solution of the mass conservation equations for species other than tracers
- Flash calculation to calculate the phase compositions for species other than tracers
- Computation of the phase densities, saturations, relative permeabilities, and capillary pressures.
- Solution of the mass conservation equations for tracer components from the independent variables. This includes the re-determination of the upstream locations from the potentials, the calculation of the convection terms from the transmissibility coefficients, phase mole fractions, densities and fluxes, and the recalculation of the well flow rates in each layer.
- Return to the first step, the solution of the pressure equation for a new time step.

In GPAS, the parameters such as the phase flux, phase saturation, phase density and upstream locations are already being calculated in EOSCOMP. Because the aqueous species do not alter the phase behavior, the equation-of-state flash calculations are performed only for the non-aqueous species and subsequently the phase saturations, phase densities and phase fluxes are determined. In order to make the code efficient, these fluid flow and rock

parameters were transferred from EOSCOMP at the last Newtonian iteration of each timestep to the chemical subroutine. The grid-block pressures are no longer required as the other parameters have already accounted for the pressure term.

The simplified method used in GPAS is as follows:

- Transfer of phase saturations, phase densities, phase flux, upstream locations and the well molar flow rates from EOSCOMP to the chemical subroutine.
- Solution of the mass conservation equation for the aqueous species incorporating the convection terms and the source/sink terms.
- Return to Step 1 for a new time step

A notable feature of this method is that the chemical module calculations are not performed for each Newtonian iteration in a timestep, since the equations for aqueous species flow are decoupled and can be solved explicitly. This saves considerable computing time for reservoir simulations with a large number of gridblocks.

Mathematical formulation

Aqueous Species Mass Conservation Equation

The differential form for the species mass conservation equation is expressed as:

$$\frac{\partial}{\partial t} \left(\phi \sum_{j=1}^{Np} \rho_j S_j \omega_{ij} + (1 - \phi) \rho_s \omega_{is} \right) + \vec{\nabla} \cdot \left(\sum_{j=1}^{Np} (\rho_j \omega_{ij} \vec{u}_j - \phi S_j \rho_s \vec{K}_{ij} \cdot \vec{\nabla} \omega_{ij}) \right) = R_i, i = 1, \dots, Nc \quad (2.1)$$

where ρ_j is the mass density of phase j , S_j is the saturation of phase j , ω_{ij} is the mass fraction of species i in j , ρ_s is the mass density of the stationary phase s , $\overleftrightarrow{K}_{ij}$ is the dispersion tensor, ω_{is} is the mass fraction of species i in the stationary phase s , $\vec{u}_j$ is the flux of phase j , and R_i is the source/sink term.

The species mass conservation equation was simplified to model the aqueous species flow using the below assumptions:

- Aqueous species do not occupy any volume
- Physical dispersion is neglected
- Aqueous species stay in the aqueous phase and do not partition to the oil or gas phase with the exception of partitioning tracers

The additional specific assumptions that apply only to tracers are

- Tracers do not change the physical properties of the fluids
- Tracer adsorption does not take place
- Tracers do not undergo chemical reactions

Thus, the surfactant and polymer component modeling option will include the capability of a surfactant and polymer to change the physical properties of the fluids, to adsorb on the rock surfaces and to undergo chemical reactions. But these surfactant and polymer specific features are added as adjunct subroutines and thus they do not change the aqueous species flow modeling equation. Thus the assumptions of a tracer flow can be applied to the aqueous species flow equation.

Applying these assumptions, the mass conservation Equation **2.1** reduces to:

$$\frac{\partial}{\partial t}(\phi \rho_1 S_1 \omega_{i1}) + \vec{\nabla} \bullet (\rho_1 \omega_{i1} \vec{u}_1) = R_i \quad (2.2)$$

The above differential equation was discretized as follows:

$$\begin{aligned} & \left(\frac{(\phi^{n+1} \rho_1^{n+1} S_1^{n+1}) w_{i1}^{n+1} - (\phi^n \rho_1^n S_1^n) w_{i1}^n}{\Delta t} \right) = \\ & - \left\{ \frac{(\rho_1^{n+1} U_{x1}^{n+1} w_{i1}^n)_{x+1/2} - (\rho_1^{n+1} U_{x1}^{n+1} w_{i1}^n)_{x-1/2}}{\Delta x} \right\} \\ & - \left\{ \frac{(\rho_1^{n+1} U_{y1}^{n+1} w_{i1}^n)_{y+1/2} - (\rho_1^{n+1} U_{y1}^{n+1} w_{i1}^n)_{y-1/2}}{\Delta y} \right\} \\ & - \left\{ \frac{(\rho_1^{n+1} U_{z1}^{n+1} w_{i1}^n)_{z+1/2} - (\rho_1^{n+1} U_{z1}^{n+1} w_{i1}^n)_{z-1/2}}{\Delta z} \right\} \\ & + \frac{(\rho_1^{n+1} q_1^{n+1}) w_{i1, inj}}{V_b^{n+1}} \end{aligned} \quad (2.3)$$

The one point upstream weighting method is used in this finite-difference formulation. In the absence of the physical dispersion, all the dispersion in the tracer output is contributed by the numerical dispersion. The upstream locations and the phase fluxes are transferred from the subroutine MRESIPW and the well flow rates are obtained from the subroutine XWELL.

The subroutines involved with the implementation of the aqueous species flow modeling in GPAS are

Xstep.df

Xaqcomp.df

Xarray.df

Xisdat.df

Xprint.f

Xchemdat.dh

Xarydat.dh

The flowchart of the aqueous species code explaining the role of each subroutine is given in **Figure 2.1**. GPAS was designed using dynamic memory allocation feature to run on parallel computers. The same parallelization concepts as used in EOSCOMP were used in the chemical subroutine. The memory for the variables is allocated dynamically and the grid indexes are in conformance with the parallelization requirement. The purpose, features and the coding details of each of the subroutines involved with the aqueous species code are discussed in the following section.

Description of the Subroutines

XSTEP

The XSTEP subroutine is the main subroutine for the compositional model EOSCOMP. For each Newtonian iteration, XSTEP calls all the other subroutines involved with the compositional model calculations. The chemical subroutine, XAQCOMP is also called from XSTEP. The variables required for the chemical subroutine are defined in an argument list, IAQCOMP. Each element of the argument list contains the pointer of the variable to be passed. This is how the dynamic memory allocation is used. The pointers are specified in XARRAY and they are stored in an include file XARYDAT using a COMMON statement. The argument list used for the chemical subroutine is given below

$$\text{IAQCOMP}(1) = 8$$

IAQCOMP(2) = N_VB
IAQCOMP(3) = N_UPD
IAQCOMP(4) = N_FLX
IAQCOMP(5) = N_POR
IAQCOMP(6) = N_SAT
IAQCOMP(7) = N_DEN
IAQCOMP(8) = N_CDTRA
IAQCOMP(9) = N_VP

The chemical subroutine is called on at the last Newtonian iteration when the flag IFLAGT or IFLAGP is turned on. The last Newtonian iteration is reached when the convergence is reached. This requirement for convergence can be bypassed when the flag NOCRASH is used in the input file. This ensures that even if the convergence is not reached before the maximum number of iterations, the run proceeds to the next time step. The call for the chemical subroutine is as follows:

```
CALL CALLWORK (XAQCOMP, IAQCOMP)
```

The subroutine CALLWORK is provided by the framework. It manages and facilitates the passing of the variables in the form of an argument list to the subroutine specified.

XAQCOMP

This is the main chemical subroutine. The name stands for aqueous component subroutine. The aqueous components implemented so far are tracer and polymer. The surfactant is not yet implemented. The output from

XAQCOMP is the aqueous species concentration as a function of time and space. The upstream locations, phase fluxes, saturation, porosity, density and well molar flow rates are obtained from the EOSCOMP subroutines. The dimensionless aqueous species concentration CDTRA is saved as a pointer variable. The algorithm used for the XAQCOMP subroutine is given below:

- Initialization The aqueous species concentration in each cell at the beginning of the simulation is initialized to zero
- Initial aqueous species mass For the present timestep, the mass of the aqueous species initially in each cell is determined from its previous concentration.
- Convection Based on the upstream locations and the phase flux, the amount of aqueous species moving into and out of a cell in X, Y and Z directions is computed.
- Source/Sink The well locations and the type of well constraints are identified. If the well is an injector, based on the injected aqueous species concentration, the corresponding mass of aqueous species is added to the already existing mass. If the well is a producer well, then based on the aqueous species concentrations determined in the cell containing

the well, the corresponding mass of the species is removed from the existing species mass in the well grid blocks.

- Mass balance check

To verify the correctness of the mass balance, the cumulative amounts of injected, produced and remaining mass of the aqueous species in the reservoir is determined for every timestep. The results are stored and printed out at the user-specified times.

XARRAY

For the dynamic memory allocation, the memory for the grid element array variable needs to be allocated by the framework using the subroutine ALCGEA. The memory for the upstream locations, the fluxes and the aqueous species concentrations are allocated in this subroutine as follows.

KE = 40

CALL ALCGEA ('UPD ',2,3,N_UPD,MEMSUM,KERR)

CALL ALCGEA ('FLX ',2,3,N_FLX,MEMSUM,KERR)

CALL ALCGEA ('CDTRA ',2,0,N_CDTRA,MEMSUM,KERR)

The significance of the above statements is described in detail below:

- KE is used to number or identify the location of wrong initialization.
- The name ALCGEA stands for allocated grid element array.
- The first argument is the grid element array variable name.

- The second argument signifies the data type of the variable wherein each number stands for a specific data type as follows

1 → Real*4	2 → Real*8
3 → Integer*2	4 → Integer*4
5 → Logical*2	6 → Logical*4

- The third argument stands for the product of the fourth and higher dimensions. Zero signifies no higher dimensions. The default first three dimensions are the grid block indexes in the X, Y and Z directions.
- The fourth argument stands for the array variable pointer or number.
- The fifth argument, MEMSUM, signifies the cumulative memory usage in bytes.
- The last argument KERR stands for the error number. Each error number corresponds to a specific error message as follows:

0 → no error
464 → too many arrays
462 → insufficient memory
465 → invalid data type

Thus, the upstream locations, aqueous phase flux and the aqueous species concentration are declared as real grid element array variables.

XISDAT

The input scalar data for the compositional model is read in XISDAT subroutine. It cannot be used to read the memory managed arrays. The aqueous

species model scalar input comprises the flag for turning the tracer model on, IFLAGT, the number of injected aqueous species, NCTR, the injected aqueous species concentration, WFINJ (mass fraction of species in aqueous phase), the aqueous species injection start time, TRA_TIMS (days) and the aqueous species injection end time, TRA_TIME (days). These keywords are specified in the input file with the appropriate units. An option to write out the chemical subroutine output using the flag OUTPUT_TRA is also included. The framework facilitates the reading of the input using the subroutine GETVAL. The commands used to read in the chemical subroutine input is as follows:

```
CALL GETVAL('NCTR ',NCTR,'I4',0,0,0,0,NDUM,NERR)
CALL GETVAL('WFINJ ',WFINJ,'R8',10,0,0,0,NDUM,NERR)
CALL GETVAL('TRA_TIMS ',TIMS,'R8',0,0,0,0,NDUM,NERR)
CALL GETVAL('TRA_TIME ',TIME,'R8',0,0,0,0,NDUM,NERR)
```

The general form of GETVAL statement is given as

```
GETVAL(VNAM, VAL, VTYP, NDIM1, NDIM2, NDIM3, NDIM4,
NUMRET, NERR)
```

where the significance of each argument is as follows:

- The first argument VNAM is the variable name with the option to assign the units. The variable name is terminated with a blank or the left bracket of the units specification.
- The second argument VAL is the Fortran code variable to which the input data is assigned.
- The third argument R8 is the variable type.

- The fourth argument NDIM1 indicates the dimensions of the variable.
- The fifth argument NDIM2 is assigned zero for unused dimensions.
- The sixth argument NDIM3 indicates character and block variables.
- The seventh argument NDIM4 stands for the maximum characters of the character variable.
- NUMRET indicates the presence of values returned in VAL().
- NERR stands for the error number. It is stepped by one for each data error encountered.

XPRINT

The XPRINT subroutine outputs the solution at one time level. If the option OUTPUT_TRA is included in the input file, then the aqueous species concentration profile at the specified times is written out. The aqueous species mass balance check is also written out at each of the user specified times. The OUTPUT_HIS option included in the input file invokes the output of the well history. The produced aqueous species concentration is also written out along with the other history parameters when the OUTPUT_TRA option is turned on. The framework handles the output of the aqueous species concentration variable.

XCHEMDAT

The aqueous species scalar input read in the subroutine XISDAT is stored in common blocks in the XCHEMDAT header file as follows:

```
COMMON /TRACR/  NCTR, WFINJ(10), TIMS, TIME, MASINJ, MASOUT,  
MASREM, IFLAGT, OUTPUT_TRA, CDTRAOUT
```

```
REAL*8 WFINJ,TIMS,TIME,MASINJ,MASOUT,MASREM,CDTRAOUT  
INTEGER NCTR  
LOGICAL*4 IFLAGT, OUTPUT_TRA
```

These variables are accessible when XCHEMDAT is attached to the subroutine XAQCOMP as the include file.

XARYDAT

The pointer variables defined in the subroutine XARRAY are stored in common blocks in XARYDAT subroutine as follows:

```
COMMON /AQCOMP/ N_UPD, N_FLX, N_CDTRA, N_VISCW
```

These pointer variables are used in the index argument list in XSTEP and then passed from one subroutine to another.

Parallelization of the chemical subroutine

The performance evaluation of the parallel subroutines is measured by speedup and scaleup, efficiency and communication expenditure. The parallel efficiency depends on three factors: degree of parallelism, amount of load transfer and load balance. The degree of parallelism is a property of the program. The amount of data transfers and load balance depends on the execution model.

The model used in GPAS is a static model. Each processor operates on a portion of the reservoir, which has been distributed across all processors prior to executing the program. The grid distribution is fixed and cannot be altered at run time. Each processor performs the same calculations but on a specific "slice" of

the reservoir. The static model can significantly reduce data transfers and allow "large" reservoirs to be distributed across the processors.

The message passing is a mechanism transferring data from one process to another. MPI is the message passing utility used in GPAS. For three-dimensional reservoir models, the number of gridblocks in the Y direction should be divisible by the number of processors. The gridblocks are divided into blocks and each processor is assigned a block. All the communications between the various processors takes place through the ghost layers around the blocks.

The steps followed to parallelize the chemical subroutine are described below:

- The first requirement is to define the new global grid element variables as dynamically memory allocated variables. The variables are defined in the subroutine XARRAY and the pointers are stored in XARYDAT.
- An argument list IAQCOMP is defined in XSTEP and the variables are assigned to each argument in the argument list. The XARRAY header file is already included in the XSTEP subroutine.
- The framework facilitates the passing of the index list variables to the subroutine XAQCOMP.
- In the XAQCOMP subroutine, the arguments order is as follows:
SUBROUTINE XAQCOMP (IDIM, JDIM, KDIM, LDIM, IL1, IL2, JL1V, JL2V, KL1, KL2, KEYOUT, NBLK, VB, UPD, FLX, POR, SAT, DEN, CDTRA, VP).

The first twelve arguments are fixed for any subroutine. The additional

variables assigned in the argument index list follow later. The grid block index in the x, y and z direction goes from IL1 to IL2, JL1V to JL2V and KL1 to KL2 respectively. The slice of the reservoir assigned to a processor has the dimensions ranging from IL1 to IL2 in the x direction, JL1V to JL2V in the y direction, and KL1 to KL2 in z direction.

- In order to share the grid element array variable as the aqueous species concentration values among the multi processors, the following statements are defined in XSTEP:

```
CALL UPDATE(N_CDTRA,1)
```

- The scalar variables as the cumulative mass of the aqueous species injected are determined by summing the mass injected on all the processors using the following statement in XSTEP subroutine:

```
CALL SUMIT(1,MASINJ)
```

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), 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 on GPAS and UTCOMP. The comparison of these results from GPAS with UTCOMP for the modified SPE fifth comparative solution project is discussed.

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 κ (Equation 1.29) for the acentric factor values greater than 0.49 while VLMU uses the same expression for all values of acentric factors (Equation 1.28). 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 taken as zero in UTCOMP.

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.

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 **Figure 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 **Figure 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 **Figure 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 weighting method is used.

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 and the pressure, the 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 **Figure 3.5**. The gas saturation profile along the x-axis was plotted for every Y grid index in **Figure 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**.

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.

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 **Figure 3.7**. The oil saturation profile

and the gas saturation profile along the x-axis for every y index is plotted in **Figure 3.8** and **Figure 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.

Three Dimensional Case

A 10x10x5 finite difference grid was used as shown in **Figure 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 on 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 **Figure 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 **Figure 3.12** and **Figure 3.13** respectively. The average reservoir pressure history from GPAS and UTCOMP were compared in

Figure 3.14. The overall production gas oil ratio comparison plots are shown in **Figure 3.15.**

Validation of the chemical module

In order to verify the correctness of the chemical subroutine, test cases were run on GPAS and the results were compared with the output from UTCHEM simulator and an analytical solution. The scope of testing covered the one-, two- and three-dimensional cases with both continuous and slug tracer injection in a single-phase, homogenous reservoir. The results compared were the tracer concentration profile and the effluent tracer concentration history. UTCHEM has been extensively tested on tracer problems, so it was chosen as the basis for testing the new modules of GPAS.

One-Dimensional Tracer Test **Continuous Injection**

The simulation results of a one-dimensional tracer flow test case were compared with the analytical solution of the convection-diffusion equation. The summary of the input data is given in **Table 3.11**. The simulation domain was 2000 ft in the x-direction, 1 ft in the y-direction and 1 ft in the z-direction. The porosity is 0.25 and the permeability is 50 md in all the three directions. A 50 x 1 x 1 grid is used with an injector at (1,1,1) and a producer at (50,1,1). The only fluid in the reservoir is water. Water tagged with the tracer is injected into the reservoir at a constant pressure. **Figures 3.16** and **3.17** show a good agreement between the analytical solution of the convection-diffusion equation and the numerical results from GPAS.

Another one-dimensional tracer flow problem was simulated with GPAS and the tracer concentration result was compared with the result from UTCHEM. The input data summary is given in **Table 3.12**. The simulation domain is 300 ft in the x-direction, 10 ft in the y-direction and 10 ft in the z-direction with a porosity of 0.25. The reservoir is homogenous with a permeability of 30 md. Water containing tracer is injected into the reservoir. The reservoir contains only the water phase. A 30x1x1 grid was used with an injector at (1,1,1) and a producer at (30,1,1). The injector bottom hole pressure is maintained at 2000 psi and the production pressure is maintained at 1000 psi. The reservoir initial pressure is 1500 psi. The total injection time of the tracer slug is 500 days. A constant time step size of 0.01 days was used for this simulation. The tracer concentration production history obtained from GPAS is compared with that of UTCHEM. **Figure 3.18** shows a good agreement between these results.

Slug Injection

The simulation domain is 300 ft in the x-direction, 10 ft in the y-direction and 10ft in the z-direction with a porosity of 0.25. The input data summary is given in **Table 3.13**. The permeability of the homogenous single-phase reservoir is 30 md. A One-dimensional 30x1x1 grid was used for this simulation problem. The only injection well is located at grid (1,1,1) and the only production well is located at gridblock (30,1,1). The aqueous tracer slug was injected from 10 days to 14.658 days. The purpose of this simulation was to compare the GPAS results

with UTCHEM. The results are shown in **Figure 3.19** and a good agreement was obtained.

Further, another problem similar to that of the previous case as summarized in **Table 3.13** was simulated but with a much smaller slug size of 2 days injection time. As expected, the GPAS results compared well with the UTCHEM result as shown in **Figure 3.20**. For the UTCHEM results, both the single point upstream and the third-order total variation diminishing (TVD) options were used for this problem.

Two-Dimensional Tracer Test

The simulation domain is 100 ft in the x-direction, 100 ft in the y-direction and 10 ft in the z-direction. A 10x10x1 grid is used and the only injector is located at (1,1,1) and the only producer is located at (10,10,1). The homogenous reservoir has a porosity of 0.25 and permeability of 10 md. Water is injected continuously at a bottomhole pressure of 2000 psi and the producer is maintained at 1000 psi bottomhole pressure. The aqueous tracer slug is injected from 10 to 14.658 days. The resident fluid in the reservoir is also water. The same problem was simulated using GPAS and UTCHEM and the results were compared. A good agreement was obtained as shown in **Figure 3.21**. A summary of the input data is given in **Table 3.14**. The problem was slightly altered by changing the permeability to 30 md and a comparison between the effluent tracer concentration from GPAS with UTCHEM is shown in **Figure 3.22**.

Three-Dimensional Tracer Test

For this three-dimensional tracer flow simulation, the input data are given in **Table 3.156**. A 5x5x5 grid was used. The only injector is located at (1,1) and is completed in all the 5 layers and the only producer is located at (5,5) and is completed in all 5 layers. A constant pressure constraint is applied to the wells. The injection pressure is held at 2000 psi while the production pressure is held at 1000 psi. The aqueous tracer slug is injected from 10 to 15 days followed by a waterflood. A summary of the input data is provided in **Table 3.15**. The tracer concentration history result of GPAS compared well with that of the UTCHEM as shown in **Figure 3.23**.

GPAS performance on PC clusters

We evaluated the performance of GPAS simulator on various Intel architecture-based PC clusters with different networks. A total of eight clusters were tested, three from the Center for Petroleum and Geosystems Engineering (CPGE), one from Texas Institute of Computational and Applied Mathematics (TICAM), and four from Dell Computer Corporation (Dell). All clusters run the Linux operation system.

Three clusters from CPGE, all of which were donated by Intel, are called Fuji, Rocky, and Ararat. The Fuji cluster consists of 16 nodes of Intel Pentium II (PII) 300 MHz processors with a 66 MHz front side bus (FSB) and 384 MB of memory. The Rocky cluster consists of 8 nodes of dual PII Xeon 400MHz processors with a 100 MHz FSB and 512 MB of memory. The most recent

cluster at CPGE is the Ararat workstation cluster. It consists of 12 nodes of PIII Xeon 1.0 GHz processors with a 400 MHz FSB and 256 MB of memory. While all cluster nodes for this study use the synchronous dynamic random access memory (SDRAM) as main memory, Ararat cluster nodes are equipped with a dual-channel Rambus DRAM (RDRAM). All compute nodes are interconnected by switched 100 Mbps Fast Ethernet at the CPGE site.

The Longhorn cluster at TICAM consists of 64 nodes of PII 400 MHz processors with a 66 MHz FSB and 512 MB of memory. Longhorn is equipped with Fast Ethernet and Myrinet network interface cards and switches.

The PowerEdge (PE) 6350 cluster includes 8 Dell PowerEdge 6350 servers, each consisting of quad PII Xeon 450 MHz processors with a 100 MHz FSB and 4.0 GB of memory. The PE 2450 cluster includes 8 Dell PowerEdge 2450 *“rack-dense, highly available”* servers, each consisting of dual PIII 733 MHz processors with a 133 MHz FSB and 512 MB of memory. The PE 4400 cluster includes 8 Dell PowerEdge 4400 *“high performance, departmental”* servers, each consisting of dual PIII Xeon 866 MHz processors with a 133 MHz memory bus and 1.0 GB of memory. The PE 1550 cluster includes 16 Dell PowerEdge 1550 *“high-performance 1U Internet and network application”* servers, each consisting of dual PIII 1.0 GHz processors with a 133 MHz memory

bus and 1.0 GB of memory. All clusters at Dell are connected by Fast Ethernet, Gigabit Ethernet, GigaNet, or Myrinet.

We created four benchmark problems based on a modified version of the Society of Petroleum Engineer's Fifth Comparative Simulation Project (Killough and Kossack, 1987), a standard reservoir simulation benchmark. In this problem, gas is injected into a reservoir containing oil and water, with one injection well and one production well. The reservoir fluid is described using three components and the simulation period is set to 100 days. Initially, we set the grid dimensions to be fairly small so that the simulation could be performed on a single processor. Later, we increased the problem size by enlarging the reservoir size but keeping the grid block size the same as in the small-scale simulation to examine the performance of the clusters in parallel processing mode, and to demonstrate the capability of the clusters for solving large-scale reservoir simulation problems.

Based on these preliminary computations, we found

1. Fast processors mainly contributed to the reduction in wall-clock time. Machines with faster memory subsystems also improved the performance of the simulator.
2. Even with the relatively limited performance of a Fast Ethernet network, we were able to achieve good scalability in most cases. With up to eight processors, Fast Ethernet scaled almost as well as other

high-speed networks, namely Myrinet and GigaNet. Myrinet performed better than GigaNet in all our benchmarks.

3. The simulator scaled very well for the homogeneous permeability model, in which the linear solver took roughly 40% of the total execution time. Modeling the heterogeneous permeability field increased the amount of computation, in which the linear solver spent more than 60% of the total execution time. The heterogeneous permeability model did not achieve as good scalability as was observed in the homogeneous model. The Inter-processor communication was larger for the heterogeneous permeability model than for the homogeneous model. Therefore, the use of a high-speed network became as important as having fast processors to reduce wall-clock time.
4. A PC cluster with 16 processors could be used effectively for large-scale reservoir simulation studies. Increasing the number of processors and allocating larger amounts of memory per processor should enable simulation of large-scale and high-resolution reservoirs with millions of gridblocks.

The complete manuscript is given in Appendix A.

REFERENCES

- 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.
- Chang, Y.B.: Development and Application of an Equation of State Compositional Simulator, Ph.D. Dissertation, 1990.
- 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.: Trapping of Micellar Fluids in Berea Sandstone, Ph.D. Dissertation, The University of Texas at Austin, 1990.
- Gosset, H. and Kalitventzeff, I.: "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.
- 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.
- Liu J., Pope, G.A., and Sepehrnoori, K.: "A High Resolution, Fully Implicit Method for Enhanced Oil Recovery Simulation," SPE 29098 presented at the 13th SPE Symposium on Reservoir Simulation held in San Antonio, TX, February 12-15, 1995.

- Lohrenz, J., Bray, B.G. and Clark, C.R.: "Calculating Viscosities of Reservoir Fluids from their Compositions," Trans., AIME, 1964.
- Maroongroge, V.: Modeling and application of tracers for reservoir characterization, Ph.D. Dissertation, The University of Texas at Austin, 1994.
- Michelsen, J.L.: "The Isothermal Flash Problem. Part I. Stability," Fluid Phase Equilibrium, 1982.
- Parashar M., Wheeler, J.A., Pope G.A., Wang, K., and Wang, P.: "A New Generation EOS Compositional Reservoir Simulator: Part II - Framework and Multiprocessing," SPE 37977 presented at the 1997 SPE Reservoir Simulation Symposium, June 8-11, USA.
- 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.
- Reid, R.C., Prausnitz, J.M. and Poling, B.E.: *The Properties of Gases and Liquids*, Fourth Edition, McGraw-Hill, Inc., NY, 1987.
- Smith, J.M. and Van Ness, H.C.: *Introduction to Chemical Engineering Thermodynamics*, McGraw-Hill, Inc., NY, 1975.
- Sandler, Stanley I.: *Chemical and Engineering Thermodynamics*, Third Edition, John Wiley & Sons, Inc., 1999.
- Trangenstein, J.A.: "Customized Minimized Techniques for Phase Equilibrium Computations in Reservoir Simulation," Chem. Eng. Sci., 42, No.12, 1987.
- 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 15th Reservoir Simulation Symposium, Houston, TX, Feb. 14-17, 1999.
- Wang, P., Yotov, I., Wheeler, M., Arbogast, T., Dawson, C., Parashar, M. and Sepehrnoori, K.: "A New Generation EOS Compositional Reservoir Simulator: Part I - Formulation and Discretization," SPE 37979 presented at the 1997 SPE Reservoir Simulation Symposium, USA.
- Wheeler, J., *et al.*: *IPARS user's manual*, October 8, 2000, University of Texas at Austin.

Table 3.1: Summary of input data for the 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 ₂ H ₆	0.5
	C ₃ H ₆	0.5
Simulation Time (day)		1.0
Time Step Size (day)		0.1

Component Critical Properties:

Component	T_c (°R)	P_c (psi)	V_c (ft ³ /lb-mole)	ω_i (dyne ^{1/4} cm ^{11/4} /gm-mole)	MW_i (lbm/lb-mole)	ψ_i
C ₂ H ₆	549.72	708.36	2.3703	0.098	30.070	111
C ₃ H ₆	657.00	670.07	2.899	0.148	42.081	40.0

Binary Interaction Coefficients:

Component	C ₂ H ₆	C ₃ H ₆
C ₂ H ₆	0	0
C ₃ H ₆	0	0

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	0.3512584	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 3.2: Comparison of the phase compositions from GPAS with UTCOMP and VLMU in the 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

Table 3.3: Comparison of the phase molar densities from GPAS with UTCOMP and VLMU in the ethane propylene batch - flash test case

Table 3.4: Summary of input data for the 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 ₁₀ H ₂₂	1.0
Water viscosity (cp)		1.0
Oil viscosity (cp)		3.1527
Water Compressibility (1/psi)		3.0x10 ⁻⁵
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 ₁₀ H ₂₂	0.0
	H ₂ 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

Table 3.4 (Cont.)

Component Critical Properties:

Component	T_c (°R)	P_c (psi)	V_c (ft³/lb-mole)	ω_i (dyne ^{1/4} cm ^{11/4} /gm-mole)	MW_i (lbm/lb-mole)	ψ_i
C ₁₀ H ₂₂	1111.8	304.0	12.087	0.488	142.3	431.0

Relative Permeability functions used for generating the tables

$K_{r1}^0 = 1$ $m = 4$ $S_{wr} = 0.1$	$K_{r2}^0 = 1$ $n = 4$ $S_{or} = 0$
$S = \frac{S_w - S_{wr}}{1 - S_{or} - S_{wr}}$ $K_{r1} = K_{r1}^o (S)^m$	$K_{r2} = K_{r2}^o (1 - S)^n$

Table 3.5: Summary of input data for the 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 ₂	0.002
	H ₂ O	0.998
Water viscosity (cp)		0.79
Water Compressibility (1/psi)		6.7e-5
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 ₂	0.999
Producer Well Location (I,J,K)		(5,5,1)
Production Bottomhole Pressure (psi)		1000

Table 3.5 (Cont.)

Component Critical Properties:

Component	T_c (°R)	P_c (psi)	V_c (ft ³ /lb-mole)	ω_i (dyne ^{1/4} cm ^{11/4} /gm-mole)	MW_i (lbm/lb-mole)	ψ_i
CO ₂	547.5	1071.0	1.505	0.23	44.01	49.0
H ₂ O	1165.14	3207.4	0.79890	0.34400	18.02	100.0

Binary Interaction Coefficients:

Component	CO ₂	H ₂ O
CO ₂	0	-0.085
H ₂ O	-0.085	0

Relative Permeability functions used for generating the tables

$$K_{r1} = 0.4089 \left(\frac{S_1 - 0.2}{0.5} \right)^3$$

$$K_{r21} = \left(\frac{0.7 - S_1}{0.5} \right)^2$$

$$K_{r3} = 0.39 \left(\frac{S_3 - 0.05}{0.6} \right)^3$$

$$K_{r23} = 0.83886 \left(\frac{0.65 - S_3}{0.6} \right)^{2.1952}$$

Capillary Pressure functions used for generating the tables:

$$P_{c21} = 0.05 + 255.5328 \exp(-8.689S_1)$$

$$P_{c23} = 6.00524 \times 10^{-4} \exp(13.544618S_3)$$

GPAS RESULT				
CO2 molefraction in the Mobile Water Phase				
LX/LY(ft)	50	150	250	350
50	0.0215	0.0211	0.0207	0.0201
150	0.0211	0.0208	0.0204	0.0134
250	0.0207	0.0204	0.0193	0.0053
350	0.0201	0.0134	0.0053	0.0029
450	0.0049	0.0031	0.0024	0.0022
450	0.0049	0.0031	0.0024	0.0022
H2O molefraction in the Mobile Water Phase				
LX/LY(ft)	50	150	250	350
50	0.9785	0.9789	0.9793	0.9799
150	0.9789	0.9792	0.9796	0.9866
250	0.9793	0.9796	0.9807	0.9947
350	0.9799	0.9866	0.9947	0.9971
450	0.9951	0.9969	0.9976	0.9978
450	0.9951	0.9969	0.9976	0.9978
CO2 molefraction in the Gas Phase				
LX/LY(ft)	50	150	250	350
50	0.9808	0.9838	0.9863	0.9885
150	0.9838	0.9855	0.9875	0
250	0.9863	0.9875	0	0
350	0.9885	0	0	0
450	0	0	0	0
450	0	0	0	0
H2O molefraction in the Gas Phase				
LX/LY(ft)	50	150	250	350
50	0.0192	0.0162	0.0137	0.0115
150	0.0162	0.0145	0.0125	0
250	0.0137	0.0125	0	0
350	0.0115	0	0	0
450	0	0	0	0
450	0	0	0	0

UTCMP RESULT				
CO2 molefraction in the Mobile Water Phase				
LX/LY(ft)	50	150	250	350
50	0.021576	0.02113	0.020671	0.020119
150	0.02113	0.020841	0.020395	0.013835
250	0.020671	0.020395	0.019807	0.005362
350	0.020119	0.013835	0.005362	0.002924
450	0.004952	0.003108	0.002413	0.002178
450	0.004952	0.003108	0.002413	0.002178
H2O molefraction in the Mobile Water Phase				
LX/LY(ft)	50	150	250	350
50	0.97842	0.97887	0.97933	0.97988
150	0.97887	0.97916	0.9796	0.98616
250	0.97933	0.9796	0.98019	0.99464
350	0.97988	0.98616	0.99464	0.99708
450	0.99505	0.99689	0.99759	0.99782
450	0.99505	0.99689	0.99759	0.99782
CO2 molefraction in the Gas Phase				
LX/LY(ft)	50	150	250	350
50	0.98077	0.98388	0.98641	0.98853
150	0.98388	0.98556	0.98758	0
250	0.98641	0.98758	0.98938	0
350	0.98853	0	0	0
450	0	0	0	0
450	0	0	0	0
H2O molefraction in the Gas Phase				
LX/LY(ft)	50	150	250	350
50	0.019229	0.01612	0.013588	0.011473
150	0.01612	0.014439	0.012419	0
250	0.013588	0.012419	0.010622	0
350	0.011473	0	0	0
450	0	0	0	0
450	0	0	0	0

Table 3.6: Comparison of the mobile water phase and gas phase compositions from GPAS with UTCOMP for the carbon dioxide sequestration problem.

Table 3.7: The absolute differences in the mobile water phase and gas phase compositions from GPAS with UTCOMP for the carbon dioxide sequestration problem at the end of 100 days.

**ABSOLUTE DIFFERENCE IN VALUES BETWEEN
GPAS AND UTCOMP RESULTS**

CO2 molefraction in the Mobile Water Phase

LX/LY(ft)	50	150	250	350	450
50	-7.6E-05	-3E-05	2.9E-05	-1.9E-05	-5.2E-05
150	-3E-05	-4.1E-05	5E-06	-0.00044	-7.6E-06
250	2.9E-05	5E-06	-0.00051	-6.2E-05	-1.3E-05
350	-1.9E-05	-0.00044	-6.2E-05	-2.4E-05	2.17E-05
450	-5.2E-05	-7.6E-06	-1.3E-05	2.17E-05	2.56E-05

H2O molefraction in the Mobile Water Phase

LX/LY(ft)	50	150	250	350	450
50	8E-05	3E-05	-3E-05	2E-05	5E-05
150	3E-05	4E-05	0	0.00044	1E-05
250	-3E-05	0	0.00051	6E-05	1E-05
350	2E-05	0.00044	6E-05	2E-05	-2E-05
450	5E-05	1E-05	1E-05	-2E-05	-3E-05

CO2 molefraction in the Gas Phase

LX/LY(ft)	50	150	250	350	450
50	3E-05	-8E-05	-0.00011	-3E-05	0
150	-8E-05	-6E-05	-8E-05	0	0
250	-0.00011	-8E-05	-0.98938	0	0
350	-3E-05	0	0	0	0
450	0	0	0	0	0

H2O molefraction in the Gas Phase

LX/LY(ft)	50	150	250	350	450
50	-2.9E-05	8E-05	0.000112	2.7E-05	0
150	8E-05	6.1E-05	8.1E-05	0	0
250	0.000112	8.1E-05	-0.01062	0	0
350	2.7E-05	0	0	0	0
450	0	0	0	0	0

Table 3.8: Summary of the component critical properties and binary interaction coefficients for the two-dimensional six component simulation example.

Component Critical Properties:

Component	T_c (°R)	P_c (psi)	V_c (ft ³ /lb-mole)	ω_i (dyne ^{1/4} cm ^{11/4} /gm-mole)	MW_i (lbm/lb-mole)	ψ_i
CH ₄	343.0	667.8	1.599	0.013	16.0	71.00
C ₃ H ₈	665.7	616.3	3.211	0.152	44.1	151.0
C ₆ H ₁₄	913.4	436.9	5.923	0.301	86.2	271.0
C ₁₀ H ₂₂	1111.8	304.0	10.087	0.488	142.3	431.0
C ₁₅ H ₃₂	1270.0	200.0	16.696	0.650	206.0	631.0
C ₂₀ H ₄₂	1380.0	162.0	21.484	0.850	282.0	831.0

Binary Interaction Coefficients:

Component	CH ₄	C ₃ H ₈	C ₆ H ₁₄	C ₁₀ H ₂₂	C ₁₅ H ₃₂	C ₂₀ H ₄₂
CH ₄	0	0	0	0	0.05	0.05
C ₃ H ₈	0	0	0	0	0.005	0.005
C ₆ H ₁₄	0	0	0	0	0	0
C ₁₀ H ₂₂	0	0	0	0	0	0
C ₁₅ H ₃₂	0.05	0.005	0	0	0	0
C ₂₀ H ₄₂	0.05	0.005	0	0	0	0

Table 3.9: Summary of input data for the 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 ₄	0.5
	C ₃ H ₈	0.03
	C ₆ H ₁₄	0.07
	C ₁₀ H ₂₂	0.2
	C ₁₅ H ₃₂	0.15
	C ₂₀ H ₄₂	0.05
Water viscosity (cp)		0.337
Water Compressibility (1/psi)		3.0e-6
Rock Compressibility (1/psi)		1.0e-6
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 ₄	0.77
	C ₃ H ₈	0.20
	C ₆ H ₁₄	0.03
Producer Well Location (I,J,K)		(5,5,1)
Production Bottomhole Pressure (psi)		1000

Table 3.9 (Cont.)

Relative Permeability functions used for generating the tables

$$K_{r1} = 0.4089 \left(\frac{S_1 - 0.2}{0.5} \right)^3$$

$$K_{r21} = \left(\frac{0.7 - S_1}{0.5} \right)^2$$

$$K_{r3} = 0.39 \left(\frac{S_3 - 0.05}{0.6} \right)^3$$

$$K_{r23} = 0.83886 \left(\frac{0.65 - S_3}{0.6} \right)^{2.1952}$$

Capillary Pressure functions used for generating the tables:

$$P_{c21} = 0.05 + 255.5328 \exp(-8.689S_1)$$

$$P_{c23} = 6.00524 \times 10^{-4} \exp(13.544618S_3)$$

Table 3.10: Summary of input data for the 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 ₄	0.5
	C ₃ H ₈	0.03
	C ₆ H ₁₄	0.07
	C ₁₀ H ₂₂	0.2
	C ₁₅ H ₃₂	0.15
	C ₂₀ H ₄₂	0.05
Water viscosity (cp)		0.337
Water Compressibility (1/psi)		3.0x10 ⁻⁵
Rock Compressibility (1/psi)		1.0e-6
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 ₄	0.77
	C ₃ H ₈	0.20
	C ₆ H ₁₄	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

Table 3.10 (Cont.)

Relative Permeability functions used for generating the tables:

$$K_{r1} = 0.4089 \left(\frac{S_1 - 0.2}{0.5} \right)^3$$

$$K_{r21} = \left(\frac{0.7 - S_1}{0.5} \right)^2$$

$$K_{r3} = 0.39 \left(\frac{S_3 - 0.05}{0.6} \right)^3$$

$$K_{r23} = 0.83886 \left(\frac{0.65 - S_3}{0.6} \right)^{2.1952}$$

Capillary Pressure functions used for generating the tables:

$$P_{c21} = 0.05 + 255.5328 \exp(-8.689S_1)$$

$$P_{c23} = 6.00524 \times 10^{-4} \exp(13.544618S_3)$$

Table 3.11: Summary of input data for the convection-diffusion equation.

Dimensions (ft):	Length	2000
	Width	1
	Thickness	1
Number of gridblocks in X direction	NX	50
Number of gridblocks in Y direction	NY	1
Number of gridblocks in Z direction	NZ	1
Porosity (fraction)		0.25
Absolute Permeability (md)		50
Initial Water Saturation (fraction)		0.99
Reservoir Temperature (°F)		130
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	H ₂ O	1.0
Water viscosity (cp)		0.79
Water Compressibility (1/psi)		6.7x10 ⁻⁵
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		10,000
Tracer Injection Time (days)		10,000
Constant Time Step Size (day)		0.1
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		2000
Norm. Injection Tracer Concentration		1.0
Injection Fluid Concentration	H ₂ O	1.0
Producer Well Location (I,J,K)		(50,1,1)
Production Bottom Hole Pressure (psi)		1000

Component Critical Properties:

Component	T_c (°R)	P_c (PSI)	V_c (ft ³ /lb-mole)	ω_i (dyne ^{1/4} cm ^{11/4} /gm- mole)	MW_i (lbm/lb- mole)	ψ_i
H ₂ O	1165.14	3207.4	1.0215	0.34400	18.02	100.0

Table 3.12. Summary of input data for the one-dimensional continuous tracer flow.

Dimensions (ft):	Length	300
	Width	10
	Thickness	10
Number of gridblocks in X direction	NX	30
Number of gridblocks in Y direction	NY	1
Number of gridblocks in Z direction	NZ	1
Porosity (fraction)		0.25
Permeability (md)		30
Initial Water Saturation (fraction)		0.99
Reservoir Temperature (°F)		130
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	H ₂ O	1.0
Water viscosity (cp)		0.79
Water Compressibility (1/psi)		6.7x10 ⁻⁵
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		500
Tracer Injection Time (days)		500
Constant Time Step Size (day)		0.01
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		2000
Norm. Injection Tracer Concentration		1.0
Injection Fluid Concentration	H ₂ O	1.0
Producer Well Location(I,J,K)		(30,1,1)
Production Bottom Hole Pressure (psi)		1000

Table 3.13: Summary of input data for the one-dimensional tracer slug flow.

Dimensions (ft):	Length	300
	Width	10
	Thickness	10
Number of gridblocks in X direction	NX	30
Number of gridblocks in Y direction	NY	1
Number of gridblocks in Z direction	NZ	1
Porosity (fraction)		0.25
Permeability (md)		30
Initial Water Saturation (fraction)		0.9999
Reservoir Temperature (°F)		130
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	H ₂ O	1.0
Water viscosity (cp)		0.52
Water Compressibility (1/psi)		3.0x10 ⁻⁶
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		300
Tracer Injection Time (days)		4.658
Constant Time Step Size (day)		0.001
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		2000
Norm. Injection Tracer Concentration		1.0
Injection Fluid Concentration	H ₂ O	1.0
Producer Well Location(I,J,K)		(30,1,1)
Production Bottom Hole Pressure (psi)		1000

Table 3.14: Summary of input data for the two-dimensional tracer slug flow

Dimensions (ft):	Length	100
	Width	100
	Thickness	10
Number of gridblocks in X direction	NX	10
Number of gridblocks in Y direction	NY	10
Number of gridblocks in Z direction	NZ	1
Porosity (fraction)		0.25
Absolute Permeability (md)		10
Initial Water Saturation (fraction)		0.99
Reservoir Temperature (°F)		150
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	H ₂ O	1.0
Water viscosity (cp)		0.44
Water Compressibility (1/psi)		6.7x10 ⁻⁵
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		400
Tracer Injection Time (days)		5.0
Constant Time Step Size (day)		0.001
Injector Well Location (I,J,K)		(1,1,1)
Injection Bottomhole Pressure (psi)		2000
Norm. Injection Tracer Concentration		1.0
Injection Fluid Concentration	H ₂ O	1.0
Producer Well Location (I,J,K)		(10,10,1)
Production Bottom Hole Pressure (psi)		1000

Table 3.15: Summary of input data for the three-dimensional tracer slug flow

Dimensions (ft):	Length	100
	Width	100
	Thickness	100
Number of gridblocks in X direction	NX	5
Number of gridblocks in Y direction	NY	5
Number of gridblocks in Z direction	NZ	5
Porosity (fraction)		0.25
Permeability (md)		30
Initial Water Saturation (fraction)		0.99
Reservoir Temperature (°F)		150
Initial Reservoir Pressure (psi)		1500
Initial Composition (mole fraction)	H ₂ O	1.0
Water viscosity (cp)		0.44
Water Compressibility (1/psi)		6.7x10 ⁻⁵
Rock Compressibility (1/psi)		0.0
Simulation Time (days)		500
Tracer Injection Time (days)		5.0
Constant Time Step Size (day)		0.001
Injector Well Location (I,J,K)		(1,1,1 to 5)
Injection Bottomhole Pressure (psi)		2000
Injection Tracer Concentration		1.0
Injection Fluid Concentration	H ₂ O	1.0
Producer Well Location (I,J,K)		(5,5,1 to 5)

Figure 1.1: Flow chart showing the general structure of GPAS simulator

Figure 2.1: Flowchart of the subroutines involved with the tracer code.

Figure 3.1: Oil/Water relative permeability curves used in the Buckley Leverett Problem.

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.

Figure 3.4: Schematic reservoir geometry (quarter five spot) for the CO₂ sequestration problem.

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.

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.

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.

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.

Figure 3.17: Comparison of GPAS result with analytical solution for the convection-diffusion equation at different injected pore volumes

Figure 3.18: Comparison of GPAS result with analytical solution for the convection-diffusion equation at 0.5 injected pore volumes

Figure 3.19: Comparison of the produced tracer concentration history from GPAS with UTCHEM for a one-dimensional continuous tracer flow.

Figure 3.20: Comparison of the produced tracer concentration history from GPAS with UTCHEM result for a one-dimensional tracer slug flow in a single-phase, homogenous reservoir with a slug size of 4.65 days.

Figure 3.21: Comparison of the produced tracer concentration history from GPAS with UTCHEM for a one-dimensional tracer slug flow in a single-phase, homogenous reservoir with a slug size of 2 days.

Figure 3.22: Comparison of the produced tracer concentration history from GPAS with UTCHEM for a two-dimensional tracer slug flow in a single-phase, homogenous reservoir.

Figure 3.23: Comparison of the produced tracer concentration history from GPAS with UTCHEM for a two-dimensional tracer slug flow in a single-phase, homogenous reservoir with an absolute permeability of 30md.

Figure 3.24: Comparison of the produced tracer concentration history from GPAS with UTCHEM for a three-dimensional slug tracer.

Appendix A

PARALLEL RESERVOIR SIMULATION ON HIGH PERFORMANCE CLUSTERS

Takaaki Uetani, Baris Guler, Kamy Sepehrnoori
Department of Petroleum and Geosystems Engineering
The University of Texas at Austin
Austin, TX 78712, USA

ABSTRACT

Numerical simulation of large complex petroleum reservoirs is computationally expensive and time-consuming. Traditionally, such simulations were performed using powerful workstations, mainframes, and in some cases, supercomputers. More recently, some developers have addressed development of reservoir simulators for massively parallel supercomputers. Due to high cost, usage of these supercomputers has been mainly limited to major oil companies and national laboratories. However, recent developments in hardware and software have made application of such parallel reservoir simulators affordable using clusters of PCs. This paper evaluates the performance of various Intel architecture-based PC clusters with different networks using a parallel compositional reservoir simulator with up to 1.5 million gridblocks for a gas injection process. Results indicate that fast processors and good computer architecture are essential for high-performance computing. In addition, high-speed networks contribute effectively to reduce the wall-clock time of the simulation on clusters of PCs.

Keywords: Reservoir Simulation, High Performance Computing, Clusters, Parallel Processing, and Interconnects

RESERVOIR SIMULATION AND HIGH PERFORMANCE COMPUTING

Reservoir simulation includes complex geological and physical models to forecast oil and gas recovery and to provide underground reservoir characterization. Mathematically, this is done by solving partial differential equations whereby Darcy's law is applied to express the fluid transport term due to pressure gradients. The partial differential equations are discretized in time and space. An iterative method is used to solve the resulting system of equations. Typical compositional reservoir simulation on a single-processor workstation is able to calculate up to 80,000 unknowns. On the other hand, to do an accurate full-field simulation could require tens of millions of unknowns. This is why making use of high-performance parallel computers is desirable to support the wide range of computational demands in reservoir simulation modeling. Many researchers have done studies of reservoir simulations on parallel architectures [1,4-7]. A parallel-computing environment is preferable not only to reduce wall-clock time but, more importantly, to solve large-scale problems which are much larger than the ones that may feasibly run on a single-processor workstation.

OBJECTIVE

The objective of this work was to evaluate the performance of several clusters of PCs, workstations, and servers with different networks using a parallel compositional reservoir simulator. Furthermore, since solving large-scale problems is as important as reduction of wall-clock time in practical reservoir simulations, this work addresses simulation of such problems on clusters of PCs as well.

DESCRIPTION OF CLUSTERS

A total of eight clusters were tested for this study, three from the Center for Petroleum and Geosystems Engineering (CPGE), one from the Texas Institute of Computational and Applied Mathematics (TICAM), and four from Dell Computer Corporation (Dell) [3]. All clusters run the Linux operating system.

Three clusters from CPGE, all of which were donated by Intel, are called Fuji, Rocky, and Ararat, each named after popular mountains in Japan, the United States and Turkey, respectively. The Fuji cluster consists of 16 nodes of Intel Pentium II (PII) 300 MHz processors with a 66 MHz front side bus (FSB) and 384 MB of memory. The Rocky cluster consists of 8 nodes of dual PII Xeon 400MHz processors with a 100 MHz FSB and 512 MB of memory. The most recent cluster at CPGE is the Ararat workstation cluster. It consists of 12 nodes of PIII Xeon 1.0 GHz processors with a 400 MHz FSB and 256 MB of memory. While all cluster nodes for this study use the synchronous dynamic random access memory (SDRAM) as main memory, Ararat cluster nodes are equipped with a dual-channel Rambus DRAM (RDRAM). All compute nodes are interconnected by switched 100 Mbps Fast Ethernet at the CPGE site.

The Longhorn cluster at TICAM consists of 64 nodes of PII 400 MHz processors with a 66 MHz FSB and 512 MB of memory. Longhorn is equipped with Fast Ethernet and Myrinet network interface cards and switches.

The PowerEdge (PE) 6350 cluster includes 8 Dell PowerEdge 6350 servers, each consisting of quad PII Xeon 450 MHz processors with a 100 MHz FSB and 4.0 GB of memory. The PE 2450 cluster includes 8 Dell PowerEdge 2450 "rack-dense, highly available" servers, each consisting of dual PIII 733 MHz processors with a 133 MHz FSB and 512 MB of memory. The PE 4400 cluster includes 8 Dell PowerEdge 4400 "high performance, departmental" servers, each consisting of dual PIII Xeon 866 MHz processors with a 133 MHz memory bus and 1.0 GB of memory. The PE 1550 cluster includes 16 Dell PowerEdge 1550 "high-performance 1U

Internet and network application” servers, each consisting of dual PIII 1.0 GHz processors with a 133 MHz memory bus and 1.0 GB of memory. All clusters at Dell are connected by Fast Ethernet, Gigabit Ethernet, GigaNet, or Myrinet.

DESCRIPTION OF THE RESERVOIR SIMULATOR

The reservoir simulator used in this report is a fully implicit, parallel, equation-of-state (EOS) compositional simulator. The Peng-Robinson EOS is used for the hydrocarbon phase-behavior calculations. The solutions for linear equations are obtained using PETSc (Portable Extensible Toolkit for Scientific Computation) linear solver package [2]. The simulator solves the equations that describe the fluid flow phenomena coupled with the phase behavior in a porous medium. The framework approach was used to separate the physical reservoir model development from parallel processing. This separation was done by IPARS (Integrated Parallel Accurate Reservoir Simulator) developed at The University of Texas at Austin [5,7].

THE BENCHMARK

We created four benchmark problems based on a modified version of the Society of Petroleum Engineer’s Fifth Comparative Simulation Project [4], a standard reservoir simulation benchmark. In this problem, gas is injected into a reservoir containing oil and water, with one injection well and one production well. The reservoir fluid is described using three components and the simulation period is set to 100 days. Initially, we set the grid dimensions to be fairly small so that the simulation could be performed on a single processor. Later, we increased the problem size by enlarging the reservoir size but keeping the grid block size the same as in the small-scale simulation to examine the performance of the clusters in parallel processing mode, and to demonstrate the capability of the clusters for solving large-scale reservoir simulation problems.

SIMULATION CASE STUDIES

The efficiency in parallel applications is usually measured by speedup. The term “*speedup*” of a cluster with N processors is defined as

$$Speedup = \frac{t_1}{t_N} \quad \text{Eq. (1)}$$

where t_1 is the amount of execution time spent with 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.

Small Scale Simulation

In order to evaluate the scalability performance of the simulator, we first propose a small-scale reservoir simulation benchmark that is small enough to be run on a single-processor machine. An 800 ft by 11,200 ft by 160 ft thick reservoir which contains two wells, a gas injection well perforated at one corner of the top layer and a production well located at the opposite corner of the bottom layer, was divided into 28,672 (16×224×8) gridblocks with 229,376 unknowns. The reservoir was assumed to have a homogeneous permeability and porosity field.

Single processor performance results:

Initially, the benchmark was performed using one processor. The simulator requires at least 300 MB of memory to simulate the benchmark. Since the compute node performance depends only on the computation time consumed by the simulator, the results indicate the internal hardware performance of each cluster node.

As shown in Figure 1, an Ararat cluster node with PIII Xeon 1.0 GHz processor achieved the fastest execution time with 309.3 seconds, while the Fuji cluster node with PII 300 MHz processor took the longest with 1030.3 seconds. Longhorn and the Rocky cluster nodes have 400 MHz processors. However, Rocky performed better than Longhorn because the Rocky nodes have more advanced Xeon-series processors, which offer twice the size of L2 cache memory of 512 KB running at the speed of the processor, and 1.5 times faster FSB than the Longhorn nodes. Although both the PowerEdge 1550 and Ararat clusters have processors running at 1.0 GHz, Ararat’s execution time was slightly faster. This could also be attributed to the fact that Ararat has a more advanced Xeon-series processor, RDRAM type memory and 400 MHz FSB, while the PowerEdge 1550 has SDRAM type memory and 133 MHz FSB. Clearly, better processor, memory and overall system design improve system performance.

We found that a computing node with faster processor and memory subsystem performs the best. The execution time is roughly inversely proportional to the clock-speed of the processor.

Two processors performance results:

We ran the same benchmark on two processors. There are two different modes of operations when using two processors:

- 1) Two processors from different nodes.
- 2) Two processors from the same node (SMP).

The first architecture is called distributed memory access and the latter is shared memory access, also known as dual processing or SMP as shown in Figure 2. Since every component as well as processors is installed “all-in-one”, SMP computers are less expensive and require less space than two uni-processor computers. However, access to a single memory unit from two processors

during the simulation causes memory contention. The distributed memory architecture, using two processors from two nodes – i.e., one processor from each node – will not have this problem but instead data must be exchanged through the network, hence creating latency problems.

In order to examine the dual processing performance of the Ararat cluster, which has only one processor per node, an additional processor was placed in one of Ararat's nodes. The results of the execution time with both memory-access types as well as the execution time results using one processor for speedup comparisons are shown in Figures 3 through 5. The results show that memory contention has a bigger impact than high-latency when simulating this particular benchmark using two processors. Xeon processors with faster memory and FSB performed better than the ordinary Pentium III processors in terms of both execution time and speedup. In particular, the PowerEdge 4400 showed excellent dual-processing performance as it has two-way interleaving memory architecture with dual memory controllers on board. Fast Ethernet was used for inter-processor communication in these runs. MPI is used to handle all the communication among the processors in both shared and distributed memory configurations.

Scalability performance with Fast Ethernet:

In this section, the performance of the clusters interconnected with a Fast Ethernet network is compared. Execution time results are shown in Figure 6. Fast Ethernet is the most popular and probably the least expensive network architecture providing 100 Mbps bandwidth. When using MPICH over TCP/IP on Fast Ethernet, the P4_SOCKBUFSIZE environment variable should be tuned to establish the maximum communication performance between processors. The socket buffer size influences the control flow of the TCP connection. The desired socket buffer size changes according to the computing environment. For example, we found that the optimized socket buffer size was 4,096 bytes for the PE1550 cluster when 16 processors are used. This value yields 22.2 seconds to execute the above benchmark, while it took 143.5 seconds when 8,192 bytes was selected as socket buffer size.

Figure 7 suggests that all the clusters showed very good scalability up to eight processors. This is because computation times scaled pretty well and the communication times spent by the MPICH library were insignificant in the clusters.

The clusters behaved differently when 16 processors are used. The Fuji and Longhorn clusters achieved almost ideal speedup. This is because they have the slowest processors, PII 300 MHz and 400 MHz, respectively, and have only one processor per compute node. Since slower processors spend the majority of the execution time on computation, the communication-to-computation ratio remains almost constant. The PE 1550 cluster also has 16 compute nodes and showed good speedup of 14.0 but less

than what the Fuji and the Longhorn clusters achieved. This is because the clock-speed of the PE 1550 cluster, 1.0 GHz, is much faster than the throughput from the memory subsystem and interconnects. Other clusters did not show as good a speedup as Fuji, since they use two processors from each node, thus creating memory contention.

Communication hardware performance:

To understand the effect of the communications hardware, we also performed the benchmark with GigaNet and Myrinet, two of the most popular high-speed networks for PC clusters. Both networks provide much higher bandwidth and lower latency than the Fast Ethernet. Link speeds for Giganet and Myrinet are 1.25 Gbps and 1.28 Gbps, respectively. The latency on Giganet and Myrinet hardware and communication protocols are about 7 μ s.

Figures 8 through 10 show that all the three networks achieved excellent speedup performance with up to 8 processors. We noticed a drop in the performance while using 32 processors. This shows that the low latency and high bandwidth interconnect hardware become very important beyond 16 processors. Myrinet was the fastest interconnect of all the simulations.

Constant Problem Size Per Processor

Unlike the previous benchmark, where the number of processors was doubled for the simulation while the grid dimension was fixed, the benchmark introduced in this section simultaneously doubles both the number of processors and the grid dimensions for the simulation. The benchmark starts with $40 \times 40 \times 12$ (19,200) gridblocks on a single processor. The reservoir properties and the production pattern are the same as in the previous small-scale benchmark. The simulator requires almost 256 MB of memory per processor to perform this simulation. In this benchmark, the computational intensity on each processor should be maintained constant at all times, which is equivalent to $40 \times 40 \times 12$ gridblocks case. Also, we expected the execution time for each physical problem to be reasonably close to the result of the $40 \times 40 \times 12$ -gridblock simulation done on one processor.

The results showed a relatively small performance drop as seen in Figure 11. Thus, domain decomposition was successfully achieved and the code showed good scalability behavior. From 8 to 16 processors, the performance drop was rather obvious with Fast Ethernet due to the increase in communication time. While all the clusters used the Fast Ethernet network in this benchmark, the Longhorn cluster also used Myrinet. The Longhorn cluster using Myrinet showed a very small performance drop up to 32 processors. There was no significant difference observed between Myrinet and Fast Ethernet up to 8 processors due to the low communications. The Rocky cluster did not perform very well with 16 processors mainly because of the memory contention problem. It uses 2 processors from each node to perform 16-processor parallel simulations.

Large Scale Reservoir Simulation

Scalability analysis of small-scale reservoir simulations was discussed in the previous sections. Although it was interesting from a theoretical point of view, the problem size was rather small. Generally, if a problem could be solved on a single processor machine within a matter of minutes, investing in an interconnect switch and extra compute nodes to set up a parallel environment is not considered economically worthwhile. This section presents the scalability test with a much larger number of gridblocks and unknowns. The effect of reservoir heterogeneity is also investigated.

In addition to Fast Ethernet and Myrinet networks, Gigabit Ethernet was studied in the next two sections as well. Gigabit Ethernet is an extension of the Ethernet standard. While Gigabit Ethernet theoretically provides about 10 times faster bandwidth than Fast Ethernet, at 1.0 Gbps, it has about the same latency as Fast Ethernet.

A 5,000 ft by 8,000 ft by 400 ft thick reservoir was divided into 320,000 ($100 \times 160 \times 20$) gridblocks with 2,560,000 unknowns. This reservoir has the same fluid properties and production pattern as in the small-scale reservoir simulation benchmark. Two different models have been created in order to investigate the effect of heterogeneity. This benchmark was intended to fulfill the CPU load and the memory capacity of the PE 4400 and 1550 clusters using 4 processors.

Homogeneous Problem:

In the first model, permeability and porosity values are kept uniform. When running the benchmark with 16 processors, the execution time on the PE 1550 cluster with Gigabit Ethernet was almost 6 minutes, while it took 7 minutes and 17 seconds on the PE 4400 cluster with Myrinet.

Overall, the PE 1550 cluster with Gigabit Ethernet was the fastest and obtained the best speedup in the homogeneous model as seen in Figure 12. The speedup of the PE 1550 cluster with Gigabit Ethernet was 15.5, almost close to the ideal value, while the speedup of the PE 4400 cluster with Myrinet was 13.2. The PE 1550 cluster with Gigabit Ethernet performed better than the PE 4400 cluster because the PE 1550 cluster has slightly faster processors (PIII 1.0 GHz) than the PE 4400 cluster (PIII Xeon 866 MHz) although Myrinet is a lower latency. Although PE 4400 cluster has higher speed interconnect (Myrinet) than PE 1550 cluster's Gigabit Ethernet, memory contention is an important factor effecting the execution and speedup performance. The PE 1550 cluster used only one processor per compute node, because the cluster has a total of 16 nodes available. By contrast, the PE 4400 cluster was required to use two processors from each node, creating memory access problems. The communication-to-computation time ratio was small but measurable, and therefore it was the clock-speed of the processors that affected the final performance rather than the high-bandwidth and low-latency network.

As seen in Figure 13, the Longhorn clusters also scaled pretty well. It took much more time for the Longhorn clusters to execute the simulation than for the PowerEdge clusters because of slower processors and memory subsystems. On the Longhorn cluster, Myrinet also performed better than Fast Ethernet. The simulation on 4 processors was not performed on the Longhorn cluster due to lack of memory.

In this large-scale simulation, the largest computational tasks were in the linear solver and in updating the Jacobian, the two of which account for about 80% of the total execution time, as seen in Figures 14 through 16. In all clusters, the linear solver occupied around 40%, and updating the Jacobian around 35 to 40% of the computing time. The differences are mainly attributed to the linear solver and updating ghost-region tasks whose execution times can be reduced by using a high-speed network. All the major computational tasks scaled very well on the PE 1550 cluster with Gigabit Ethernet. However, it was the linear-solver time that did not scale well on the PE 4400 cluster with Myrinet although the rest of the computational tasks scaled pretty well.

Heterogeneous Problem:

In this case, we simulated a layered reservoir. Permeability values in the layers vary from 1 to 200 md while porosity is kept uniform. The clusters behaved differently in the heterogeneous case than when simulating the homogeneous case. It took almost twice the computation time to simulate the heterogeneous problem than to simulate the homogeneous problem. The speedup results are seen in Figures 17 and 18. The PE 1550 cluster with Gigabit Ethernet and the PE 4400 cluster with Myrinet performed similarly but showed different behavior than the simulation with the homogeneous case, in which the PE 1550 with Gigabit Ethernet performed better than the PE 4400 cluster, throughout. In the simulation with 4 processors, the PE 1550 cluster with Gigabit Ethernet performed better than the PE 4400 cluster. Using 8 processors, the PE 4400 cluster with Myrinet performed better than the PE 1550 cluster with Gigabit Ethernet. Finally, the PE 1550 cluster with Gigabit Ethernet performed better than the PE 4400 cluster with 16 processors.

As shown in Figure 18, the Longhorn cluster scaled pretty well up to 16 processors but the execution time could not be reduced effectively with 32 processors. From 16-processor to 32-processor parallel simulation, Myrinet only reduced the execution time by 3.5% and with Fast Ethernet the execution time increased. None of the major computational tasks scaled well with 32 processors. It was updating the Jacobian and dependent variables tasks that took even longer to execute than did the simulation with 16 processors.

Time spent on individual computational tasks, such as updating the Jacobian, updating viscosity and updating dependent variables, increased from the homogeneous cases by about 10%. The computation time for the linear-

solver more than doubled and resulted in accounting for more than 60% of the total execution time, as seen in Figures 19 through 20. The communication time spent by the MPICH library on Fast Ethernet with up to 8 processors was more than triple that with the homogeneous problem, thus increasing the communication-to-computation ratio of the final performance. This is mainly attributed to the fact that the linear system of equations arising from linearization of the nonlinear system of equations solved in the simulation is much more difficult to solve for the heterogeneous permeability case than for the homogeneous permeability case. As a consequence, the solver in the simulator will require a larger number of iterations to converge for solving the problem with the heterogeneous permeability field.

Very-Large-Scale Reservoir Simulation

One of the challenges in reservoir simulation is to simulate a reservoir using a very large number of gridblocks in a reasonable amount of time. In the previous sections, we demonstrated the potential of solving large problems using clusters of PCs. In this section we address simulating a larger problem consisting of 1.5 million gridblocks using 16 processors. A 15,000 ft by 12,800 ft by 400 ft thick reservoir was divided into 1,536,000 (300×256×20) gridblocks with over 12 million unknowns. The reservoir has a homogeneous porosity and permeability field that is simulated for 100 days. This reservoir has the same fluid properties and production pattern as in the small-scale reservoir simulation benchmark.

This significant computational problem took approximately 31 minutes to execute on 16 PowerEdge 1550 servers connected by Gigabit Ethernet network. The PETSc linear solver took the largest portion, about 44% of the total computational time, followed by updating the Jacobian, which took 31%, as seen in Figure 21. During the entire simulation, we observed that all the nodes were using all the available memory in the system.

SUMMARY

Based on our computational results, we found the following:

1. Fast processors mainly contributed to the reduction in wall-clock time. Machines with faster memory subsystems also improved the performance of the simulator.
2. Even with the relatively limited performance of a Fast Ethernet network, we were able to achieve good scalability in most cases. With up to eight processors, Fast Ethernet scaled almost as well as other high-speed networks, namely Myrinet and GigaNet. Myrinet performed better than GigaNet in all our benchmarks.
3. The simulator scaled very well for the homogeneous permeability model, in which the linear solver took roughly 40% of the total execution time. Modeling the heterogeneous permeability field increased the amount of computation, in which the linear solver spent more than 60% of the total execution time. The heterogeneous

permeability model did not achieve as good scalability as was observed in the homogeneous model. The Inter-processor communication was larger for the heterogeneous permeability model than for the homogeneous model. Therefore, the use of a high-speed network became as important as having fast processors to reduce wall-clock time.

4. A PC cluster with 16 processors could be used effectively for large-scale reservoir simulation studies. Increasing the number of processors and allocating larger amounts of memory per processor should enable simulation of large-scale and high-resolution reservoirs with millions of gridblocks.

ACKNOWLEDGEMENTS

This research was supported by the U.S. Department of Energy and the sponsors of the Joint Industry Project on Reservoir Simulation at the Center for Petroleum and Geosystems Engineering at The University of Texas at Austin. We would like to thank Intel for donating the clusters, and Dell for the use of their clusters, and providing technical support.

REFERENCES

- [1] Abate, J., Wang, P., and Sepehrnoori, K. (2001). "Parallel Compositional Reservoir Simulation on Clusters of PCs," The International Journal of High Performance Computing Applications Volume 15, No. 1, Spring 2001, pp. 13-21.
- [2] Balay, S., Gropp, W.D., McInnes, L.C., and Smith, B. (2001). "PETSc Users Manual," ANL-95/11 - Revision 2.1.0, Argonne National Laboratory, Argonne, IL.
- [3] Dell Computer Corporation. (2001, January 22). "Dell: PowerEdge Servers," Advertisement in The New York Times, C16, Round Rock, TX.
- [4] Killough, J.E., and Kossack, C.A. (1987). "Fifth Comparative Simulation Project: Evaluation of Miscible Flood Simulators," paper SPE 16000 presented at the 1987 Society of Petroleum Engineers Symposium on Reservoir Simulation. San Antonio, TX.
- [5] Parashar, M., Wheeler, J.A., Pope, G.A., Wang, K., and Wang, P. (1997). "A New Generation EOS Compositional Reservoir Simulator: Part II – Framework and Multiprocessing," paper SPE 37977 presented at the 1997 Society of Petroleum Engineers Reservoir Simulation Symposium. San Antonio, TX.
- [6] Rame, M., and Delshad, M. (1995). "A Compositional Reservoir Simulator on Distributed Memory Parallel Computers," paper SPE 29103 presented at the 1995 Society of Petroleum Engineers Symposium on Reservoir Simulation. San Antonio, TX.
- [7] Wang, P., Balay, S., Sepehrnoori, K., Wheeler, J., Abate, J., Smith, B., and Pope, G.A. (1999). "A Fully Implicit Parallel EOS Compositional Simulator for Large Scale Reservoir Simulation," paper SPE 51885 presented at the 1999 Society of Petroleum Engineers Symposium on Reservoir Simulation. Houston, Texas.

Figure 1: Execution time results simulating the small-scale reservoir simulation problem using a single processor.

Figure 2: Shared memory access types.

Figure 3: Dual processing performance on the PowerEdge 1550 cluster.

Figure 4: Dual processing performance on the Ararat cluster.

Figure 5: Dual processing performance on the PowerEdge 4400 cluster.

Figure 6: Execution time results for the small-scale reservoir simulation problem using Fast Ethernet.

Figure 7: Speedup results simulating the small-scale reservoir simulation problem using Fast Ethernet.

Figure 8: Interconnect speedups simulating the small-scale reservoir simulation problem on the Longhorn cluster.

Figure 11: Execution time results for the constant problem size per processor problem.

Figure 9: Interconnect speedups simulating the small-scale reservoir simulation problem on the PowerEdge 6350 cluster

Figure 12: Speedup results for the large-scale reservoir simulation problem with homogeneous field on the PowerEdge clusters.

Figure 10: Interconnect speedups simulating the small-scale reservoir simulation problem on the PowerEdge 2450 cluster

Figure 13: Speedup results for the large-scale reservoir simulation problem with homogeneous field on the Longhorn cluster.

Figure 14: Computational breakdowns for the large-scale reservoir simulation problem with homogeneous field using 8 processors.

Figure 15: Computational breakdowns for the large-scale reservoir simulation problem with homogeneous field using 16 processors.

Figure 16: Computational breakdowns for the large-scale reservoir simulation problem with homogeneous field using 32 processors.

Figure 17: Speedup results for the large-scale reservoir simulation problem using the heterogeneous case on the PowerEdge clusters.

Figure 18: Speedup results for the large-scale reservoir simulation problem with heterogeneous field on the Longhorn cluster.

Figure 19: Computational breakdowns for the large-scale reservoir simulation problem with heterogeneous field using 8 processors.

Figure 20: Computational breakdowns for the large-scale reservoir simulation problem with heterogeneous field using 16 processors.

Figure 21: Computational tasks breakdown for the very large-scale reservoir simulation problem on the PowerEdge 1550 cluster with Gigabit Ethernet.