



SPE 100079

A Fully Implicit, Compositional, Parallel Simulator for IOR Processes in Fractured Reservoirs

Reza Naimi-Tajdar¹, SPE, Choongyong Han, SPE, Kamy Sepehrnoori, SPE, Todd J. Arbogast, SPE, and Mark A. Miller², SPE, The University of Texas at Austin. ¹now with BP, ²now with Lucid Reservoir Technologies.

Copyright 2006, Society of Petroleum Engineers

This paper was prepared for presentation at the 2006 SPE/DOE Symposium on Improved Oil Recovery held in Tulsa, Oklahoma, U.S.A., 22–26 April 2006.

This paper was selected for presentation by an SPE Program Committee following review of information contained in an abstract submitted by the author(s). Contents of the paper, as presented, have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material, as presented, does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Papers presented at SPE meetings are subject to publication review by Editorial Committees of the Society of Petroleum Engineers. Electronic reproduction, distribution, or storage of any part of this paper for commercial purposes without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of where and by whom the paper was presented. Write Librarian, SPE, P.O. Box 833836, Richardson, TX 75083-3836, U.S.A., fax 01-972-952-9435.

Abstract

Naturally fractured reservoirs contain a significant amount of the world oil reserves. A number of these fields contain several billion barrels of oil. Accurate and efficient reservoir simulation of naturally fractured reservoirs is one of the most important, challenging, and computationally intensive problems in reservoir engineering. Parallel reservoir simulators developed for naturally fractured reservoirs can effectively address the computational problem.

A new accurate parallel simulator for large-scale naturally fractured reservoirs, capable of modeling fluid flow in both rock matrix and fractures, has been developed. The simulator is a parallel, 3D, fully implicit, equation-of-state compositional model that uses numerical tools for solving very large, sparse linear systems arising from discretization of the governing partial differential equations. A generalized dual porosity model, the multiple-interacting-continua (MINC), has been implemented in this simulator. The matrix blocks are discretized into subgrids in both horizontal and vertical directions to offer a more accurate transient flow description in matrix blocks. We believe this implementation has led to a unique and powerful reservoir simulator that can be used by small and large oil producers to help them in design and prediction of complex gas and waterflooding processes on their desktops or a cluster of computers. Some features of this simulator, such as modeling both gas and water processes with the ability of two-dimensional matrix subgridding for naturally fractured reservoirs, to the best of our knowledge are not available in any commercial simulator. The development was performed on a cluster of processors, which has proven to be a very efficient and convenient resource for developing parallel programs.

The results were successfully verified against analytical solutions and commercial simulators (ECLIPSE and GEM). Excellent agreement was achieved for a variety of reservoir case studies. Applications of this model for several IOR processes (including gas and water injection) are demonstrated. Simulation results using the simulator on a cluster of processors are also presented. Excellent speedups were obtained using the simulator in conjunction with solving a variety of problems.

Introduction

The so-called dual porosity model is one of the most widely used conceptual models for naturally fractured reservoirs. In the dual porosity model, two types of porosity are in a rock volume: fracture and matrix. Matrix blocks are surrounded by fractures and the system is visualized as a set of stacked cubes, representing matrix blocks separated by fractures (Fig.1). There is no communication between matrix blocks in this model, and the fracture network is continuous. Matrix blocks do communicate with the fractures that surround them. A mass balance for each of the media yields two continuity equations that are connected by so-called matrix-fracture transfer functions. The transfer functions characterize fluid flow between matrix blocks and fractures. The performance of dual porosity simulators is determined by the accuracy of this transfer function.

The dual porosity continuum approach was first proposed by Barenblat *et al.*¹ for a single-phase system. Later, Warren and Root² used this approach to develop a pressure-transient analysis method for naturally fractured reservoirs. Kazemi *et al.*³ extended the Warren and Root method to multiphase flow using a two-dimensional, two-phase, black-oil formulation and derived two flow equations, one for the matrix and another for the fracture. The two equations were then linked by means of a matrix-fracture transfer function. Since the publication of Kazemi *et al.*³, the dual porosity approach has been widely used in the industry to develop field-scale reservoir simulation models for modeling fractured reservoir performance⁴⁻⁸.

In simulating a fractured reservoir, we are faced with the following distinct problem: matrix blocks may contain over 90 percent of the total oil reserve, and the problem of oil recovery from a fractured reservoir is essentially that of extracting oil from these matrix blocks and not necessarily from the fractures. Therefore, any effort to understand the mechanisms that take place in matrix blocks and to simulate these

processes within their container as accurately as possible is very crucial. The main aim should be to simulate the matrix-block behavior as accurately as possible so the fracture pressure, which is used essentially as the boundary condition for matrix blocks, can be known accurately. Further discretizing the matrix blocks into subgrids or subdomains seems to be a very good solution to accurately take into account transient flow in the matrix blocks. The resulting finite-difference equations are solved along with the fracture equations to calculate matrix-fracture transfer flow. The way that matrix blocks are discretized varies in the proposed models, but the objective is to accurately model pressure and saturation gradients in the matrix blocks^{9,5,10-16}.

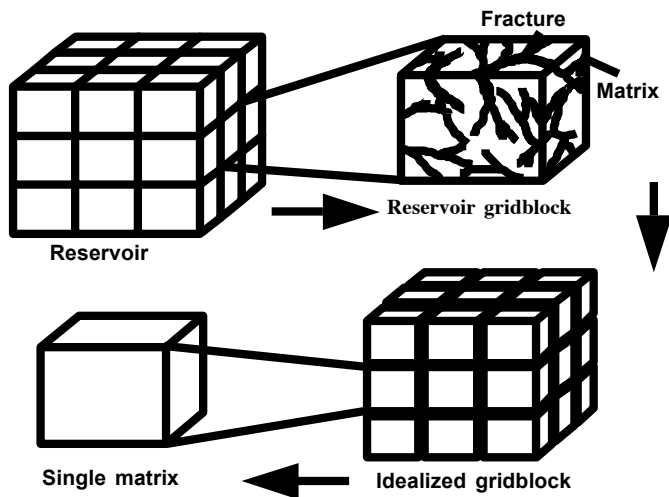


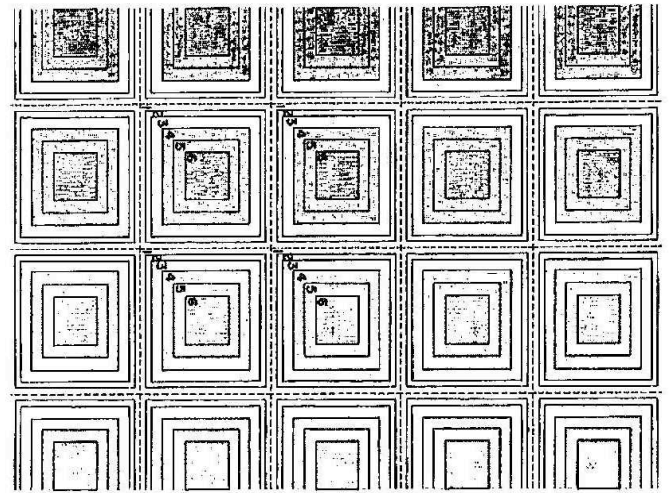
Fig. 1. Idealized representation of fractured reservoirs².

As a generalization of the dual porosity concept, Pruess and Narasimhan¹¹ developed a “Multiple Interacting Continua” method (MINC), which treats the multiphase and multidimensional transient flow in both fractures and matrix blocks by a numerical approach. The transient interaction between matrix and fracture is treated in a realistic way. The main assumption in the MINC method is that thermodynamic conditions in the matrix depend primarily on the distance from the nearest fracture¹¹. Therefore, one can partition the flow domain into compositional volume elements in such a way that all interfaces between volume elements in the matrix are parallel to the nearest fracture. Subgridding of matrix blocks on the basis of distance from the fractures gives rise to a pattern of nested volume elements. For the 2D case, it is shown in Fig. 2. Each volume element has a defined thermodynamic state assigned to it. The basic MINC concept of partitioning matrix blocks according to distance from the fracture faces can be extended readily to arbitrary irregular block shapes and sizes.

Subgridding dramatically increases computer time and storage, especially since all the flow equations are solved implicitly. While the serial computing and simulation technology may be adequate for typical reservoirs, naturally fractured reservoirs need more gridblocks to adequately define the transient flow in matrix media. Parallel reservoir simulation technology places a powerful tool in the hands of

reservoir engineers and geologists for determining accurate fluid in place, sweep, and reservoir performance.

The primary objectives of this study are (1) to develop a new, parallel, equation-of-state compositional, fully implicit simulator to model three-phase fluid flow in naturally fractured oil reservoirs for IOR processes, (2) to verify and test the developed model against analytical solutions and



commercial simulators, and (3) to investigate the parallel efficiency of the developed model.

Fig. 2 Basic 2D computational mesh for fractured reservoir¹¹.

Mathematical Formulation

In the dual porosity model, two overlapping continua, one corresponding to the fracture medium and one corresponding to the matrix medium, are considered. Thus two values of most variables and parameters are attributed to each point. The equations of motion and of component mole conservation are written independently for each medium and should hold at every point of the fracture and matrix medium and at all times. The four most important transport mechanisms occurring in permeable media are viscous forces, gravity forces, dispersion (diffusion), and capillary forces¹⁷. Transfer of fluids between the two media is taken into consideration by a source/sink (transfer) function.

Isothermal multicomponent and multiphase flow in a porous medium can be described using three different types of equations: (1) Component conservation equations; (2) Equations constraining volume and component moles; and (3) Phase equilibrium equations dealing with equilibrium component mass transfer between phases, in which flash calculations using an EOS are performed to determine amounts and compositions of equilibrium phases.

Neglecting dispersion term and mutual solubility between water and hydrocarbon phases, for a system consisting of n_c hydrocarbon components and n_p fluid phases (excluding the aqueous phase), the above three types of equations are mathematically expressed for a control volume in the following sections.

Component Mole Conservation Equations. Darcy's law is a fundamental relationship describing the flow of fluids in permeable media. The differential form of Darcy's law can be used to treat multiphase unsteady-state flow, non-uniform permeability, and 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. In terms of moles per unit time, the hydrocarbon component conservation equations, for both the fracture and matrix systems are the following:

Fracture system (subscript f):

$$V_b \frac{\partial}{\partial t} (\phi_f N_{fi}) - V_b \vec{\nabla} \cdot \sum_{j=1}^{n_p} \lambda_{fj} \xi_{fj} x_{fij} (\vec{\nabla} P_{fj} - \gamma_{fj} \vec{\nabla} D_f) \quad (1)$$

$$-q_i + \tau_{mfi} = 0$$

Matrix system (subscript m):

$$V_b \frac{\partial}{\partial t} (\phi_m N_{mi}) - V_b \vec{\nabla} \cdot \sum_{j=1}^{n_p} \lambda_{mj} \xi_{mj} x_{mij} (\vec{\nabla} P_{mj} - \gamma_{mj} \vec{\nabla} D_m) \quad (2)$$

$$= 0$$

For $i = 1, 2, \dots, n_c$

These equations also hold for water by inserting the properties of the aqueous phase.

Volume Constraint Equations. The volume constraint states that the pore volume in each cell must be filled completely by the total fluid volume. The volume constraint equations for both fracture and matrix media are the same and are as follows:

$$\sum_{i=1}^{n_c+1} N_{fi} \sum_{j=1}^{n_p} L_j \bar{V}_j = 1 \quad (\text{fracture system}) \quad (3)$$

$$\sum_{i=1}^{n_c+1} N_{mi} \sum_{j=1}^{n_p} L_j \bar{V}_j = 1 \quad (\text{matrix system}) \quad (4)$$

Phase Equilibrium Equations. The equilibrium solution must satisfy three conditions¹⁸. First, the molar balance constraint must be preserved. Second, the chemical potentials for each component must be the same in all phases. Third, the Gibbs free energy at constant temperature and pressure must be a minimum. With the assumption of local thermodynamic equilibrium for the hydrocarbon phases, the criterion of phase equilibrium applies:

$$\ln(f_{fi}^g) - \ln(f_{fi}^o) = 0 \quad (\text{fracture system}) \quad (5)$$

$$\ln(f_{mi}^g) - \ln(f_{mi}^o) = 0 \quad (\text{matrix system}) \quad (6)$$

Independent Variables. Equations 1 through 6 describe the fluid flow through porous media in naturally fractured reservoirs. There are $2(2n_c+2)$ equations. Independent unknowns are chosen as $\ln K_b$, N_b , P_w , N_w (N = moles per unit pore volume) in each medium, fracture and matrix, which

gives $2(2n_c+2)$ primary variables. This set of independent variables is likely to be the best choice largely because it makes the fugacity equations nearly linear¹⁹. All the fluid-related properties and variables in Eqs. 1 through 6 can be expressed as a function of the selected independent variables.

Transfer Function and Boundary Conditions. The transfer function for each component (hydrocarbon components and water), τ_{mfi} are evaluated at the boundary between the matrix and fracture media and have the following forms:

$$\tau_{mfi} = NM \cdot V_b \sum_{l=1}^{N_b} \frac{\partial}{\partial t} (\phi_m N_{mi})_l \quad (7)$$

where NM is the total number of matrix blocks within a fracture gridblock, and N_b is the number of matrix subgrids. No-flow boundary conditions for component mole conservation equations in the fracture system are considered. The boundary condition for matrix blocks is the continuity of all phase pressures.

Solution Approach

The material balance equations (Eqs. 1 and 2) need to be discretized using a proper scheme for a given grid system that represents the geometry of the reservoir. A fully implicit solution method is used to solve the governing equations. This method treats each term in Eqs. 1 and 2 implicitly. These 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. An analytical method is used to calculate the elements of the Jacobian matrix. The term Jacobian refers to the mathematical matrix that forms the system of linear equations and whose elements are the derivatives of the governing equations with respect to the independent variables. Using the Schur complement method (described in the next section), the matrix equations, in each fracture cell, are condensed and added to the diagonal elements of the fracture system. Finally, the nonlinear fracture equations are solved by one of the linear solvers of PETSc (Portable Extensible Toolkit for Scientific Computation). To solve the governing equations for the independent variables over a timestep, we take the following steps:

1. Initialization of fluids at both fracture and matrix cells. The pressure, overall composition, and temperature are specified;
2. Determination of phase properties and phase state. Flash calculations are performed for each cell to determine phase compositions and densities. The states of all the phases present are then labeled as gas, oil, or aqueous phase. Phase viscosities and relative permeabilities are then computed;
3. Linearization of the governing equations for both fracture and matrix medium. All the governing equations are linearized in terms of the independent variables, and the Jacobian matrices for both fracture and matrix medium are formed. The elements of the Jacobian are computed;
4. Decomposition of the matrix medium from the fracture medium and reduction of the linear system.

The Schur complement method is used to decouple the matrix system from the fracture system (for more details, see the next section). This will reduce the linear system to only the fracture system;

5. Solution of the reduced system of the linear equations (fracture system) for the independent variables in the fracture medium;
6. Solution of the decoupled equations for the independent variables in the matrix medium;
7. Updating the physical properties;
8. Check for convergence. The residuals of the linear system obtained in Step 3, which contain both fracture and matrix media, 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.

Schur Complement Method. Let the number of fracture unknowns be I , and denote them by F and associated to each gridblock $i=1,2,\dots,I$, we have a series of matrix unknowns M . After linearization by the Newton method, fracture and matrix equations can be summarized by

$$A_{ff}F + B_{fm}M = B_f \quad (\text{fracture system}) \quad (8)$$

$$C_{mf}F + D_{mm}M = B_m \quad (\text{matrix system}) \quad (9)$$

By solving for M in the matrix equations (Eq. 9), we obtain

$$M = D_{mm}^{-1} (B_m - C_{mf} F) \quad (10)$$

By substitution of M (Eq. 10) into the fracture equations (Eq. 8), we obtain

$$A_{ff}F + B_{fm}D_{mm}^{-1} (B_m - C_{mf} F) = B_f \quad (11)$$

And by solving Eq. 11 for F we obtain

$$(A_{ff} - B_{fm}D_{mm}^{-1}C_{mf})F = B_f - B_{fm}D_{mm}^{-1}B_m \quad (12)$$

In Eq. 12, there are no matrix unknowns (M), and they have been reduced to a system of fracture equations only (i.e., single porosity model). In Eq. 12, the challenge is to calculate terms $(B_{fm}D_{mm}^{-1}C_{mf})$ and $(B_{fm}D_{mm}^{-1}B_m)$. The most time- and memory-consuming part of the calculations is the solution of the inverse of the Jacobian matrix for the matrix media D_{mm}^{-1} . By considering Eqs. 10 and 12, we actually need $D_{mm}^{-1}C_{mf}$ and $D_{mm}^{-1}B_m$ rather than D_{mm}^{-1} alone. Suppose $X_m = D_{mm}^{-1}B_m$ and $Y_{mf} = D_{mm}^{-1}C_{mf}$. Then, to calculate these terms, we need to solve the following linear equations for X_m and Y_{mf} :

$$D_{mm}X_m = B_m \quad (13)$$

$$D_{mm}Y_{mf} = C_{mf} \quad (14)$$

B_m is a vector of size $(2n_c+2)n_Hn_v$ and C_{mf} is a matrix of size $(2n_c+2)n_Hn_v$ by $(2n_c+2)$. Hence, by combining these two systems of linear equations, there will be only $(2n_c+3)$ right sides compared to $(2n_c+2)n_Hn_v$ right sides in the first method.

Fluid-Related Calculations Using EOS

The phase equilibrium calculations play a critical role in both development of an EOS compositional simulator and its efficiency. One of the major convergence problems of EOS compositional simulators is caused by inefficient treatment of the fluid-related calculations. These calculations determine the number, amounts, and compositions of the phases in equilibrium. The sequence of phase equilibrium calculations in the simulator is as follows:

- Using the phase stability analysis, the number of phases in each gridblock is determined.
- Next, the compositions of each equilibrium phase are calculated.
- The phases in each gridblock are saved for the next timestep calculations.

The key equations used to calculate the fluid properties such as viscosities, relative permeability, capillary pressure, and phase molar density are described in Chang²⁰.

Parallel Implementation

Increased oil and gas production from naturally fractured reservoirs using enhanced oil-recovery processes involves numerical modeling of such processes to minimize the risk involved in development decisions. The production from naturally fractured reservoirs requires much more detailed analyses with a greater demand for reservoir simulations with geological and physical models of much more detail than for conventional reservoirs. The computational work required to produce accurate simulations is very intensive for these problems, and thus there is a great need for parallel computing.

To develop the compositional dual porosity model in a parallel processing platform, we used a framework approach to handle the complicated tasks associated with parallel processing. The goal was to separate the physical model development from parallel processing. To achieve this, we employed an Integrated Parallel Accurate Reservoir Simulator (IPARS) framework^{21,22}. The IPARS framework includes a number of advanced features such as providing the entire 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 workstations. The framework allows the representation of heterogeneous reservoirs with variable porosity and permeability and allows the reservoir to consist of one or more fault blocks.

Spatial decomposition of the reservoir gridblocks is applied for parallel processing. The original reservoir simulation domain is divided vertically into several subdomains equal to the number of processors required by the run. The computations are decomposed on parallel machines to achieve computational efficiency. The gridblocks assigned to each processor are surrounded by one or more layers of grid

columns representing gridblocks of the neighboring processors. This is referred to as the boundary or ghost layer. The framework provides a routine that updates data in the communication layer. The message passing interface (MPI) is used in the framework to handle necessary message sending/receiving between processors.

The fully implicit EOS compositional formulation has already been implemented into the IPARS framework and successfully tested on a variety of computer platforms such as IBM SP and a cluster of PCs¹⁹. The goal of this work was to add a dual porosity module to the existing compositional Peng-Robinson cubic equation-of-state module. The simulator includes the IPARS framework and the compositional and dual porosity modules. The entire software which includes the framework and several modules for IOR processes is called General Purpose Adaptive Simulator (GPAS) and is illustrated in Fig. 3.

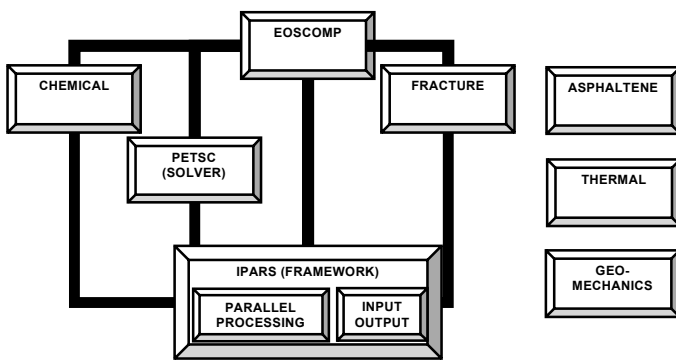


Fig. 3. General schematic of GPAS.

Verification Studies

Numerous verification case studies were performed to verify and test the newly developed dual porosity option of the GPAS simulator to model naturally fractured reservoirs. Here, we present only few of them. First, the single porosity option of the GPAS was tested after implementation of the dual porosity option using a series of waterflood processes in the fracture and matrix media. Then a modified version of Kazemi's³ quarter-five-spot waterflood case (2D and 3D) was used to validate. Finally, the compositional option of the GPAS for naturally fractured reservoirs was verified using a 3D gas-injection process.

2D Waterflood. Modified Kazemi's³ quarter-five-spot waterflood was used to verify the 2D option of the developed dual porosity option of GPAS simulator. The results were compared against the results of the UTCHEM simulator²³. Aldejain¹⁶ showed "a very close match" between the results of UTCHEM compared against the ECLIPSE simulator²⁴ for a similar case. In this case, water is injected into a quarter-five-spot model at a rate of 200 STB/D and liquids are produced from the other end at a constant pressure of 3900 psia. The reservoir is 600 ft long, 600 ft wide, and 30 ft thick. The fracture media is discretized into 8x8 uniform gridblocks in the x and y directions, respectively, and has one 30-ft-thick gridblock in the z direction. The input parameters are given in

Table 1. Zero capillary pressure is used for both fracture and matrix media, and the relative permeability curves in the fracture and matrix media are shown in Fig. 4.

Table 1. Input parameters used for 2D waterflood.

Description	Fracture	Matrix
Number of gridblocks	8 x 8 x 1	4x4 & 16x16
Size of gridblocks	75x75x30 ft ³	10x10x30 ft ³
Porosity	0.01	0.19
Permeability	500 md	1.0 md
Initial water saturation	0.0001	0.25
Water viscosity	0.5 cp	0.5 cp
Oil viscosity	2.0 cp	2.0 cp
Residual oil saturation	0.0	0.3
Oil endpoint relative perm	1.0	0.92
Corey's exponent for oil	2.15	1.8
Residual water saturation	0.0	0.25
Water endpoint relative perm	1.0	0.2
Corey's exponent for water	1.46	1.18
Initial reservoir pressure	4000 psia	4000 psia
Water injection rate	200 STB/D (constant)	
Production well pressure	3900 psia (constant)	

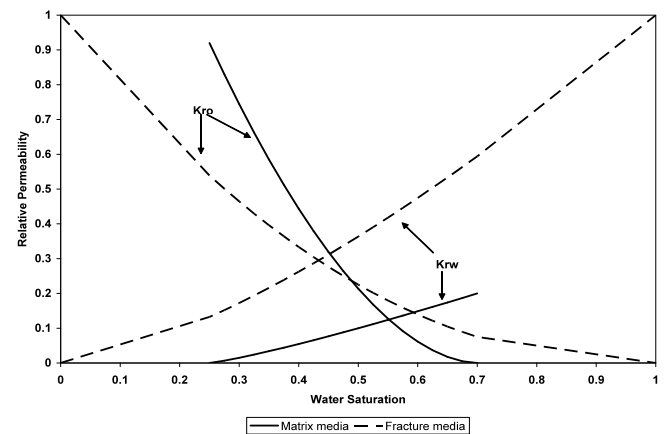


Fig. 4. Fracture and matrix relative permeabilities used in the quarter-five-spot waterflood

Since ECLIPSE does not have an option to subgrid in the vertical direction, we used a 4x1 subgridding for the matrix media. The oil recovery and oil and water production rates are shown in Figs. 5 and 6, respectively, which represent an excellent agreement when we compare the results of GPAS, UTCHEM and ECLIPSE. Note that the oil recovery is very low (less than 6% after 1200 days, which equals 0.625 water-injected pore volumes). Subgridding in the vertical direction was investigated in the next series of runs. Figures 7 and 8 show the comparison between the results of GPAS, UTCHEM, and ECLIPSE for cases of 4x1 and 4x4 subgrids. Note that due to the lack of vertical subgridding option in the ECLIPSE, there are no results for ECLIPSE run of 4x4 case. A very interesting fact can be noticed by looking closely at these figures. The oil recovery has increased to about 20% from 6% with vertical subgridding. Also, the water breakthrough is delayed to about 50 days in the 4x4 subgrids case. These results show the importance of vertical

subgridding when gravity drainage is dominant in oil recovery from naturally fractured reservoirs. Hence, the results of the simulators without vertical subgridding could be very misleading.

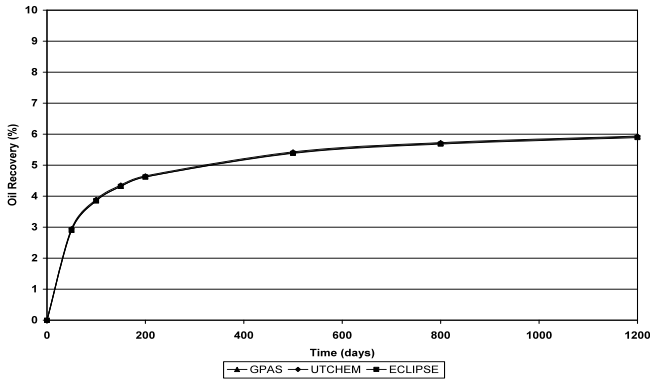


Fig. 5. Oil recovery vs. time for the quarter-five-spot waterflood (4x1 subgrids)

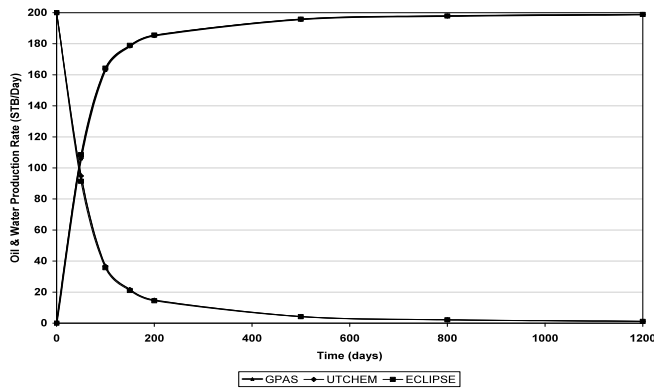


Fig. 6. Oil and water production rates for the quarter-five-spot waterflood (4x1 subgrids)

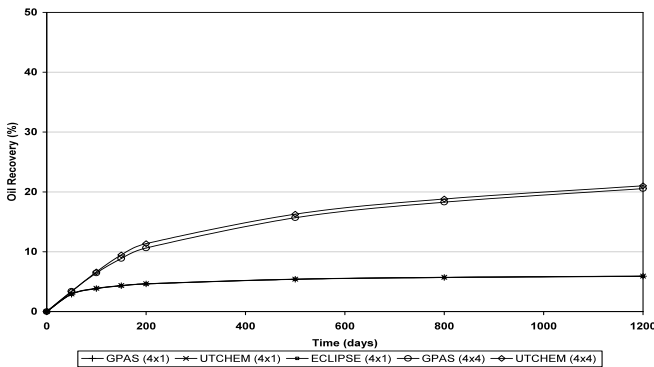


Fig. 7. Oil recovery for the quarter-five-spot waterflood (4x1 and 4x4 subgrids)

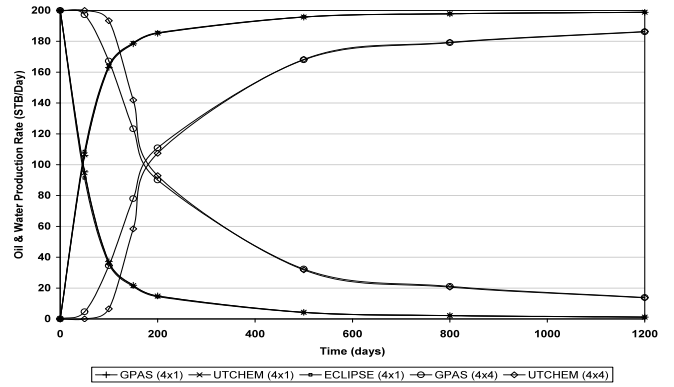


Fig. 8. Oil and water production rates for the quarter-five-spot waterflood (4x1 and 4x4 subgrids)

To investigate the effect of capillary in the matrix media, we used the same input parameters of these cases and added the capillary pressure option of GPAS. Figure 9 shows the capillary pressure curve used in the matrix media. Again, zero capillary pressure is assumed for the fracture media. Figures 10 and 11 show the oil recovery and the oil and water production rates for the 4x4 subgrids case, with and without the capillary pressure option. There are very interesting features in these figures. First, the oil recovery has increased by a factor of two from 20% to more than 40% after 1200 days (which equals 0.625 water-injected pore volumes) and second, the water breakthrough has been delayed from 50 days to almost 300 days. These behaviors show that in addition to gravity drainage, we have a very strong imbibition mechanism due to capillary pressure (Fig. 9) in this case. In other words, water is imbibed into the rock matrix from the fracture and oil comes out of the matrix and is produced through fracture channels. Hence, both gravity drainage and capillary pressure play very important mechanisms in oil recovery in naturally fractured reservoirs.

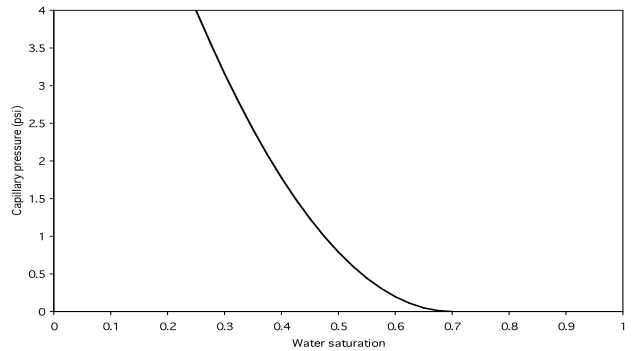


Fig. 9. Capillary pressure curve used in the quarter-five-spot waterflood (4x4 subgrids)

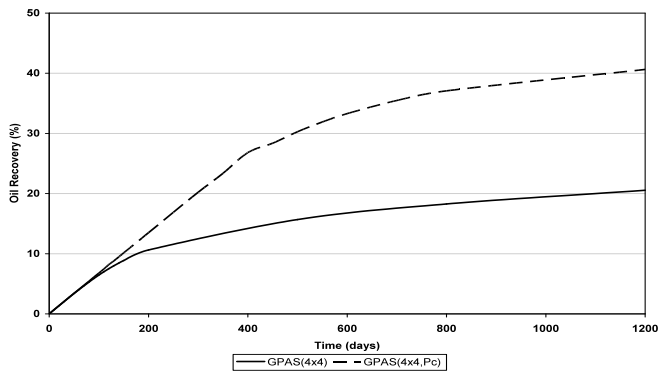


Fig. 10. Oil recovery for the quarter-five-spot waterflood (4x4 subgrids, with and without capillary pressure option)

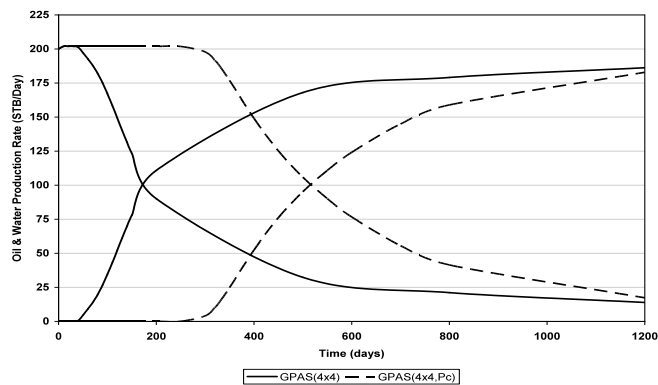


Fig. 11. Oil and water production rates for the quarter-five-spot waterflood (4x4 subgrids, with and without capillary pressure option)

3D Gas Injection. A six-component, constant-rate, gas-injection process (a modified version of the SPE fifth comparative solution problem²⁵) was used to verify the compositional option of the developed dual porosity module of GPAS. The simulation domain is 560x560x100 ft³ and is discretized in 7x7x3 fracture gridblocks with the matrix blocks size of 10x10x10 ft³ discretized in 2x1 subgrids. The three layers have 20, 30, and 50 ft thickness, respectively from the top of the reservoir. The initial reservoir pressure is 1500 psi. The only injection well is located at gridblocks (1,1,1) through (1,1,3) and injects at the constant rate of 1 Mscf/D, and the production well is located at gridblocks (7,7,1) through (7,7,3) and produces at a constant bottomhole pressure of 1300 psia. The reservoir description of this case is presented in Table 2. Straight lines three-phase relative permeabilities are used for the fracture network and the three-phase relative permeabilities used in matrix media are shown in Fig. 12. The hydrocarbon phase behavior is given using a six-component application of the Peng-Robinson equation of state. Table 3 shows the initial composition and properties of the six components and binary interaction coefficients used in the fracture and matrix media, respectively.

Table 2. Input parameters used for 3D gas injection.

Description	Fracture	Matrix
Number of gridblocks	7 x 7 x 3	2 x 1
Size of gridblocks in layer 1	80 x 80 x 20 ft ³	10x10x10 ft ³
Size of gridblocks in layer 2	80 x 80 x 30 ft ³	10x10x10 ft ³
Size of gridblocks in layer 3	80 x 80 x 50 ft ³	10x10x10 ft ³
Porosity	0.02	0.35
Permeability (X,Y,Z)	50,50,5 md	1,1,1 md
Initial water saturation	0.01	0.17
Water viscosity	1.0 cp	1.0 cp
Residual oil saturation	0.0	0.1
Oil endpoint relative Perm	1.0	0.9
Corey's exponent for oil	1.0	2.0
Residual gas saturation	0.0	0.0
Gas endpoint relative Perm	1.0	0.9
Corey's exponent for gas	1.0	2.0
Residual water saturation	0.0	0.30
Water endpoint relative Perm	1.0	0.4
Corey's exponent for water	1.0	3.
Initial reservoir pressure	1500 psia	1500 psia
Gas injection rate	1.0 Mscf/D (constant)	
Production well pressure	1300 psia (constant)	

Table 3. Initial composition and properties of the components used for 3D gas injection.

Properties	C ₁	C ₃	C ₆	C ₁₀	C ₁₅	C ₂₀
T _c (°R)	343.0	665.7	913.4	1111.8	1270.0	1380.0
P _c (psia)	667.8	616.3	439.9	304.0	200.0	162.0
V _c (ft ³ /lb-mole)	1.599	3.211	5.923	10.087	16.696	21.484
Z _c (no unit)	0.29	0.277	0.264	0.257	0.245	0.235
MW	16.0	44.1	86.2	142.3	206.0	282.0
Acentric factor	0.013	0.152	0.301	0.488	0.650	0.850
Parachors	71.0	151.0	271.0	431.0	631.0	831.0
Initial composition	0.50	0.03	0.07	0.20	0.15	0.05
Injected gas composition	0.77	0.20	0.01	0.01	0.005	0.005

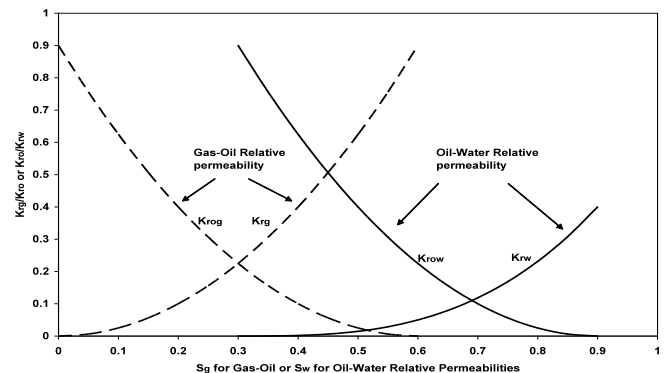


Fig. 12. Three-phase matrix relative permeabilities used in the gas injection.

To test the results of the GPAS simulation runs, the dual porosity option of the GEM simulator²⁶ (compositional mode) was used. Since the GEM simulator did not have an option to discretize the matrix media in compositional mode at the time of this study, we used Kazemi-Gilman's shape factor to calculate the matrix-fracture transfer function. The results of the GPAS and GEM simulation runs for oil recovery and oil and gas production rates vs. time are shown in Figs. 13 through 15, respectively. The results of the GPAS and GEM

showed a good agreement with a slight difference at early time due to the difference in flash calculation between GPAS and GEM.

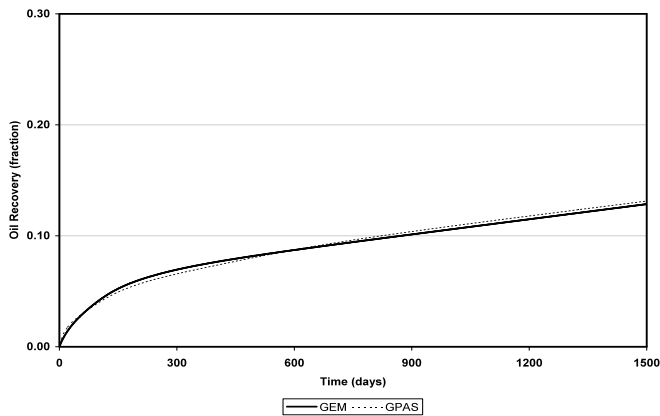


Fig. 13. Oil recovery vs time for gas injection.

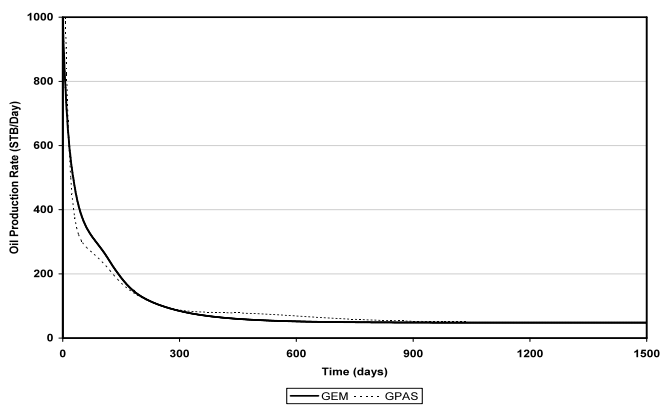


Fig. 14. Oil production rate vs time for gas injection.

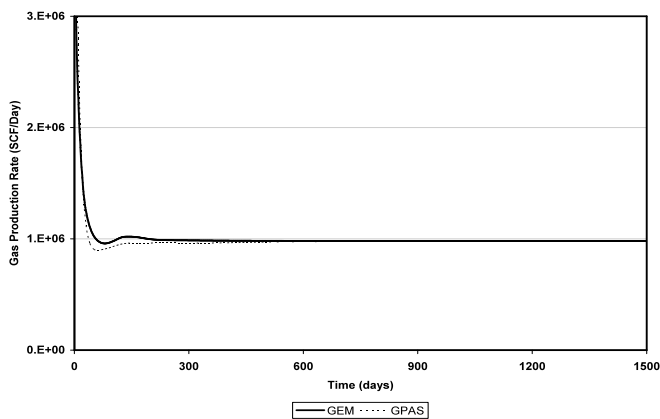


Fig. 15. Gas production rate vs time for gas injection.

Parallel Dual Porosity Reservoir Simulation

The performance of the developed dual porosity option of GPAS simulations in a parallel processing platform is presented in this section. The speedup is one of the ways to

measure parallel processing performance efficiency and is defined as: $\text{Speedup} = t_1 / t_n$.

Where t_1 is the execution time on a single processor and t_n is the execution time on n processors. The ideal speedup of parallel simulation with n processors is n , which means the program runs n times faster. However, in reality, as the number of processors becomes larger, a speedup less than n is usually observed. This performance reduction is due to increasing inter-processor communication. Also, it can be due to an unfavorable programming style, in which a program does not decompose the application evenly (load-balance issue). A Dell PowerEdge 1750 cluster system (Lonestar) was used for these parallel processing cases. The Lonestar cluster system is a Cray-Dell Linux cluster located in the Texas Advanced Computing Center (TACC) of The University of Texas at Austin. The TACC Cray-Dell PowerEdge Xeon Cluster contains 768 Xeon processors at 3.06GHz speed and 256 Xeon processors at 3.2GHz speed. A Myrinet-2000 switch fabric, employing PCI-X interfaces, interconnects the nodes (I/O and compute) with a point-to-point bandwidth of 250MB/sec (500MB/sec bidirectional).

2D Waterflood Case Study. The 2D quarter-five-spot waterflood problem was used and scaled up in size to investigate the performance of the developed dual porosity option of GPAS simulations in a parallel processing platform. The described model used consists of 16,384 gridblocks (4,096 fracture gridblocks and 2x2 matrix subgrids in each fracture gridblock) and two operating wells.

This case was run in serial and parallel modes on the Lonestar cluster system using two, four, eight, and sixteen processors. Identical results for oil recovery and water production were obtained for the serial (single processor) and the parallel (multiprocessors) simulations. Table 4 lists the speedups and execution times on the Lonestar cluster, and Fig. 16 shows plot of the speedups for different numbers of processors. As expected, by increasing the number of processors, the execution time for running the case is decreased. The speedups of 3.66 and 6.24 obtained for four and eight processors, respectively, are not too far from the ideal speedup line, whereas the speedup of 10.49 for 16 processors is much farther from the ideal speedup line. The main reason for this is that up to eight processors, the system is large enough for each processor (from a computational point of view). However, the system becomes smaller for each processor using 16 processors, and most of the time is consumed in message passing among the processors. To better understand this issue, the CPU times for major sections of the code such as Linear Solver, Updating Jacobian and Residuals, and calculating Dependent Variables were recorded. These sections were timed individually during the simulation and reported at the end. Figures 17 shows the pie charts for one and 16 processors, respectively. Note that most of the inter-processor communications are included in the solver calculation. By looking at Fig. 17, it can be seen that the solver takes a large amount of CPU time using 16 processors (28%) compared to one processor (13%), which confirms the increase in the inter-processor communications for the 16 processors run. Also, the CPU time to update the Jacobian

matrix using 16 processors (54%) is much smaller than for one processor (67%).

Table 4. Speedups and execution times for parallel simulations (2D waterflood).

No. of Processors	CPU Time (sec)	Speedup
1	22,647	1
2	11,273	2.01
4	6,192	3.66
8	3,627	6.24
16	2,159	10.49

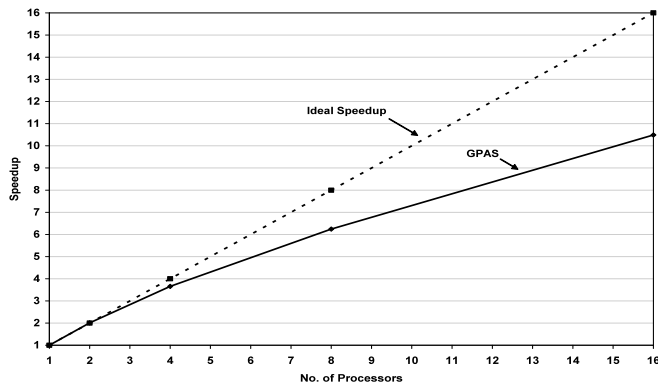
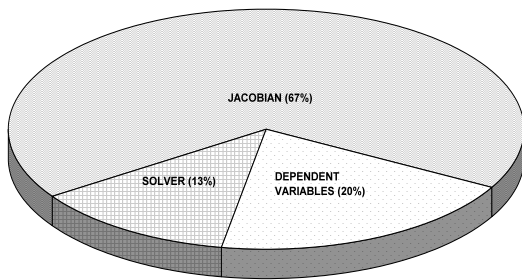


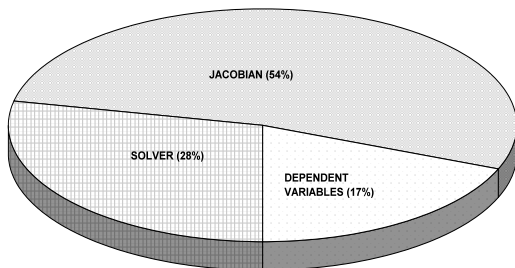
Fig. 16. Speedups for parallel simulations (2D waterflood).

3D Waterflood Case Study. To illustrate the performance of the developed dual porosity option of GPAS in a parallel processing platform in 3D with more gridblocks and unknowns, simulation of a large 3D five-spot waterflood model with five operating wells was performed. In this case, the total number of gridblocks is 98,304 (64x64x6 fracture gridblocks and 2x2 matrix subgrids in each fracture gridblock). This leads to 393,216 unknowns in the fracture and matrix media solved at each timestep. There are four injection wells injecting at constant rates of 250 STB/D at the corners of the reservoir model and one production well producing at the constant bottomhole pressure of 3500 psia in the center. The initial reservoir pressure in both fracture and matrix media is 4000 psia.

Because the computational model is large, the simulation was run on the Lonestar cluster system in parallel mode using 2, 4, 8, 16, and 32 processors. The results of simulation runs show the same oil recovery and oil and water production rates regardless of the number of processors used. Table 5 lists the speedups and execution times using the Lonestar cluster, and Fig. 18 shows the speedups for different numbers of processors. Again as expected, by increasing the number of processors, the execution time for running the case is decreased and the speedup is increased linearly. Figure 18 shows that the speedup is fairly linear up to 16 processors. However, we have a super linear speedup of 33.4 for 32 processors. One reason for this difference in speedup between the ideal line and 32 processors is probably because processors in the Lonestar cluster system are not homogeneous. As mentioned earlier, in the Lonestar cluster system, some of the processors have 3.06 GHz and some have 3.2 GHz speed, and we have no control on choosing the processors for the simulation runs. However, the chance for the single processor run to be directed to the processors with 3.06 GHz is more than three times than for the processors with 3.2 GHz, and this possibly could explain why for the 32 processors run, we exceed the ideal line. Figure 19 shows the breakdown of the CPU time for the two-processor run. The results of simulation runs show the same pie chart regardless of the number of processors used. In this case, for a two-processor simulation run, updating the Jacobian takes about 65% of the CPU time, while the parallel solver consumes about 15% of the computer time.



1 processor



16 processors

Fig. 17. Execution time breakdown for 2D waterflood (1 and 16 processors).

Table 5. Speedups and execution times for parallel simulations (3D waterflood)

No. of Processors	CPU Time (sec)	Speedup
2	74,471	2
4	36,911	4.03
8	18,236	8.17
16	9,323	15.98
32	4,456	33.42

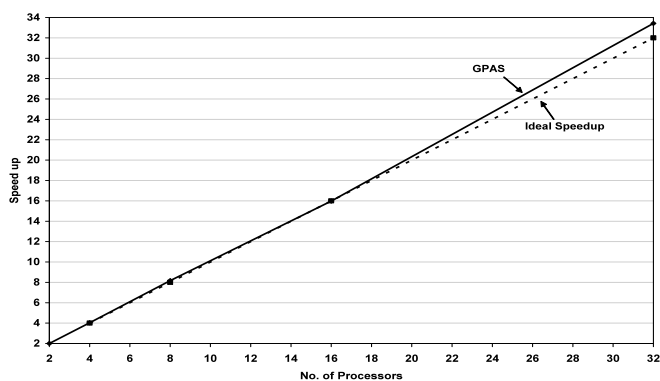


Fig. 18. Speedups for parallel simulations (3D waterflood).

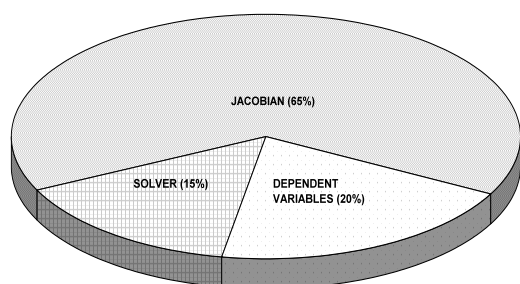


Fig. 19. Execution time breakdown for 3D waterflood (2 processors).

3D Gas Injection Case Study. The six-component, constant-rate, gas-injection process (a modified version of the SPE fifth comparative solution problem²³) was used to investigate the efficiency of the parallel processing simulation of the developed compositional dual porosity module of the GPAS simulator. The simulation domain is 1280x1280x100 ft³ discretized in 64x64x2 fracture gridblocks with the matrix blocks size of 10x10x10 ft³ discretized in 2x2 subgrids. The two layers each have thickness of 50 ft. In this case, the total number of gridblocks is 32,768 (64x64x2 fracture gridblocks and 2x2 matrix subgrids in each fracture gridblock). This leads to 458,752 unknowns in the fracture and matrix media solved at each timestep. The initial reservoir pressure is 1500 psia. There are four gas injectors located at corners of the reservoir model that inject at the constant rate of 1 Mscf/D each, and the only production well is located at the fracture gridblocks (32,32,1) to (32,32,2) and produces at constant bottomhole pressure of 1200 psia. The hydrocarbon phase behavior is obtained using a six-component application of the Peng-Robinson equation of state.

Once again, because the computational model is large for this case, there are no results available for the single processor. The simulation case was run on the Lonestar cluster system in parallel mode using 2, 4, 8, 16, and 32 processors. The results of simulation runs show the same oil recovery and oil and gas

production rates regardless of the number of processors used. Table 6 lists the speedups and execution times on the Lonestar cluster and Fig. 20 shows plot of the speedups for different number of processors and indicates a linear speedup for up to 32 processors.

Table 6. Speedups and execution times for parallel simulations (gas injection)

No. of Processors	CPU Time (sec)	Speedup
2	40,967	2
4	19,382	4.23
8	9,570	8.23
16	5,188	15.79
32	2,574	31.83

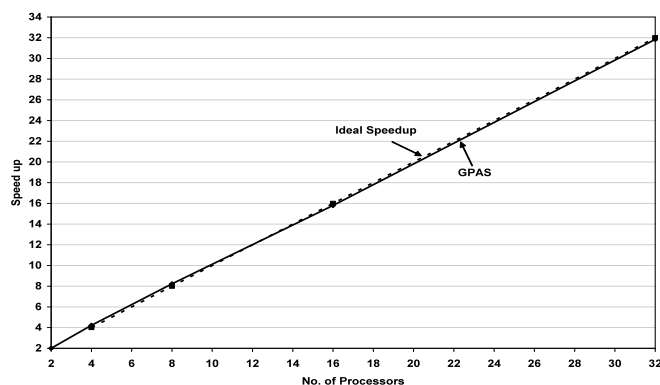


Fig. 20. Speedups for parallel simulations (gas injection).

In this large-scale simulation, the largest computational tasks were updating the Jacobian and the linear solver, which account for more than 90% of the total execution time, as seen in the pie chart for the 32-processors run (Fig. 21). Also, Uetani *et al.*²⁷ reported about 75% and Dogru *et al.*²⁸ reported about 76% of the CPU time is consumed in these two calculations for single porosity parallel simulators, which is very similar to the case reported in this study. However, there is a very important difference between this case and the other two. Based on Uetani's report, from this 75%, the parallel solver consumes more than 44% while updating Jacobian consumes about 31%. Also, based on Dogru's report, the parallel solver consumes 61% and updating Jacobian 15% of the computation time. Hence, the biggest chunk of CPU time consumption in single porosity parallel simulators is the parallel solver. By looking at Fig. 21, the parallel solver consumes only about 20% of the computer time while the updating Jacobian consumes more than 70%. The main reason for this difference is the nature of the solution approach in this developed dual porosity option of GPAS. As discussed in Solution Approach section of this paper, the entire calculation for the matrix unknowns is done locally within the processors while calculations for the fracture unknowns are done in the parallel solver. Hence, in our model, most of the calculations are done locally and we do not need to have a lot of inter-processor communications. This fact confirms the suitability of the developed dual porosity option of GPAS in parallel processing platforms. The other reason is that this is a compositional run with six components, and flash calculations, which include updating the Jacobian and residuals are

calculated locally, and are very time consuming. This explains the increase of almost 10% in updating Jacobian in this case relative to the 3D waterflood Case.

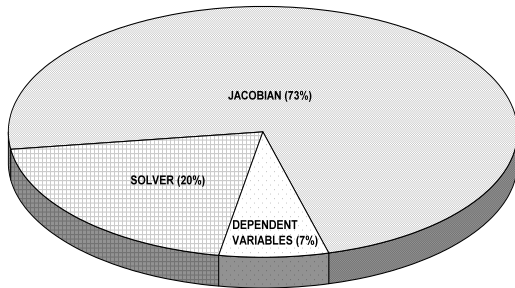


Fig. 21. Execution time breakdown for gas injection (32 processors).

Conclusions

The following conclusions are drawn from this study.

1. A new, parallel, equation-of-state compositional, fully implicit simulator to model naturally fractured oil reservoirs has been developed and verified successfully against the UTCHEM simulator as well as other commercial simulators. The methodology used to develop this reservoir simulator is based on a modified dual porosity model.
2. The matrix blocks were discretized in the horizontal direction using a modified multiple interacting continua (MINC) method concept and in the vertical direction using a stacked grids concept to provide an accurate calculation for matrix-fracture transfer function in transient flow. Based on our knowledge at this time, there is no compositional reservoir simulator that has the capability of subgridding in both horizontal and vertical directions. This subgrid scheme was used to model three-dimensional flow in the matrix rock with two-dimensional subgrids with good accuracy.
3. The efficiency of the parallel processing of the GPASv3.5 was verified using 2D and 3D waterflood as well as 3D compositional gas-injection processes. For the 2D waterflood process, a linear speedup up to eight processors was achieved. For the 3D waterflood process, a linear speedup was obtained using up to 32 processors. In the 3D compositional gas injection with six hydrocarbon components cases, because of the large size of the problem, excellent speedup was achieved. In this case, the linear solver and updating the Jacobian accounted for the majority of the CPU time consumption. Flash and phase behavior calculations in the fracture gridblocks and matrix subgrids, which are performed locally, account for this increase of CPU time consumption in updating the Jacobian.

4. The efficiency of the parallel processing of the developed dual porosity simulator was more satisfactory relative to the parallel efficiency of our single porosity simulator. The main reason is that based on the solution approach used in the developed dual porosity simulator, the unknowns in the matrix media are solved locally in each processor that contains the corresponding fracture gridblock. Hence, there would be much less time consumed in inter-processor communication that arises in the linear solver, thereby increasing the parallel efficiency of the developed dual porosity simulator.

Acknowledgement

The financial support of the Reservoir Simulation Joint Industry Project in the Center for Petroleum and Geosystems Engineering at The University of Texas is gratefully acknowledged.

Nomenclature

D = depth measured positive downward, L

f = fugacity

K_i = equilibrium ratio, dimensionless

L_j = mole fraction of phase j

NM = number of matrix blocks within a fracture gridblock

N_b = number of matrix subgrids

N_i = Moles of component i , per unit pore volume, mol/L³

n_c = number of hydrocarbon components

n_p = number of phases

P = Pressure, m/Lt²

q = flow rate, L³/t

t = time, t

V = volume, L³

\bar{v}_j = molar volume of phase j

x_{ij} = mole fraction of component i in phase j

ϕ = porosity, fraction

γ = fluid specific gravity m/L²t

τ = matrix-fracture transfer function, L³/t

λ = effective mobility, L³/mt³

ξ_j = molar density of phase j , mol/L³

Subscripts

b = bulk

f = fracture

g = gas

i = component index

j = phase index

m = matrix

o = oil

w = water

References

1. Barenblatt, G.E., I.P. Zheltov, and I.N. Kochina: "Basic Concepts in the Theory of Seepage of Homogeneous Liquids in Fissured Rocks," *J. Applied Mathematics (USSR)* (1960) **24**, No. 5, 1286-1303.
2. Warren, J.E. and P.J. Root: "The Behavior of Naturally Fractured Reservoirs," *Soc. Pet. Eng. J.*, (Sept. 1963) 245-255, *Trans.*, AIME, **228**.

3. Kazemi, H., L.S. Merrill, K.L. Porterfield, P.R. Zeman: "Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs," paper SPE 5719 presented at the SPE-AIME Fourth Symposium on Numerical Simulation of Reservoir Performance, Los Angeles, CA (Feb. 19-20, 1976).
4. Thomas, L.K., T.N. Dixon, and R.G. Pierson: "Fractured Reservoir Simulation," *Soc. Pet. Eng. J.* (Feb. 1983) 42-54.
5. Gilman, J.R. and H. Kazemi: "Improvements in Simulation of Naturally Fractured Reservoirs," *Soc. Pet. Eng. J.* (Aug. 1983).
6. Dean, R.H. and L.L. Lo: "Development of a Naturally Fractured Reservoir Simulator and Examples of Its Use," paper SPE 14110 presented at the 1986 Society of Petroleum Engineers International Meeting on Petroleum Engineering, Beijing, China, March 17-20.
7. Beckner, B.L., A. Firoozabadi, and K. Aziz: "Modeling Transverse Imbibition in Double Porosity Simulator," paper SPE 17414 presented at the 1987 California Regional Meeting, Long Beach, March 23-25.
8. Rossen, R.H. and E.I. Shen: "Simulation of Gas/Oil Drainage and Water/Oil Imbibition in Naturally Fractured Reservoirs," paper SPE 16982 presented at the 1987 Society of Petroleum Engineers Annual Technical Conference and Exhibition, Dallas, Sept. 27-30.
9. Saidi, M.A.: "Mathematical Simulation Model Describing Iranian Fractured Reservoirs and Its Application to Haft Kel Field," *Proceeding of the Ninth World Petroleum Congress, Tokyo (1975) 4*, PD 13(3), 209-219.
10. Gilman, J.R.: "An Efficient Finite-Difference Method for Simulating Phase Segregation in Matrix Blocks in Double Porosity Reservoirs," *Soc. Pet. Eng. Res. Eng.* (July 1986).
11. Pruess, K. and T.N. Narasimhan: "A Practical Method for Modeling Fluid and Heat Flow in Fractured Porous Media," paper SPE 10509 presented at the Sixth SPE Symposium on Reservoir Simulation, New Orleans, LA, February 1982, *Soc. Pet. Eng. J.*, Vol. 25, No. 1 (Feb. 1985) 14-26.
12. Wu, Y.S., K. Pruess: "A Multiple-Porosity Method for Simulation of Naturally Fractured Petroleum Reservoirs," paper SPE 15129 presented at the 56th California Regional Meeting of the SPE, Oakland, CA (April 2-4, 1986).
13. Chen, W.H. and R. E. Fitzmorris: "A Thermal Simulator for Naturally Fractured Reservoirs," paper SPE 16008 presented at the 1987 Society of Petroleum Engineers Symposium on Reservoir Simulation, San Antonio, Feb. 1-4.
14. Douglas, J., T. Arbogast, and P.J. Paes Leme: "Two Models for the Waterflooding of Naturally Fractured Reservoirs," paper SPE 18425 presented at the 1989 Society of Petroleum Engineers Symposium on Reservoir Simulation, Anaheim, CA, Feb. 6-8.
15. Beckner, B.L., H.M. Chan, A.E. McDonald, S.O. Wooten, and T.A. Jones: "Simulating Naturally Fractured Reservoirs Using a Subdomain Method," paper SPE 21241 presented at the 1991 Society of Petroleum Engineers Symposium on Reservoir Simulation, Anaheim, CA, Feb. 17-20.
16. Aldejain, Abdulaziz A.: "Implementation of a Dual Porosity in a Chemical Flooding Simulator," Ph.D. dissertation, The U. of Texas, Austin (1999).
17. Lake, L.W.: *Enhanced Oil Recovery*, Prentice Hall, Englewood Cliffs, NJ (1989).
18. Baker, L.E., A.C. Pierce, and K.D. Luks: "Gibbs Energy Analysis of Phase Equilibria," *Soc. Pet. Eng. J.*, 22, No. 5, 731-742 (1982).
19. Wang, P., S. Balay, K. Sepehrnoori, J. Wheeler, J. Abate, B. Smith, G.A. Pope: "A Fully Implicit Parallel EOS Compositional Simulator for Large Scale Reservoir Simulation," paper SPE 51885 presented at the SPE 15th Reservoir Simulation Symposium, Houston, TX (Feb. 14-17, 1999).
20. Chang, Yih-Bor: "Development and Application of an Equation of State Compositional Simulator," Ph.D. dissertation, The U. of Texas, Austin (1990).
21. Parashar, M., J.A. Wheeler, G. Pope, K. Wang, and P. Wang: "A New Generation EOS Compositional Reservoir Simulator: Part II – Framework and Multiprocessing," paper SPE 37977 presented at the SPE Symposium on Reservoir Simulation, Dallas, TX, 8-11 June 1997.
22. Wang, P., I. Yotov, M. Wheeler, T. Arbogast, C. Dawson, M. Parashar, and K. Sepehrnoori: "A New Generation EOS Compositional Reservoir Simulation: Part I. Formulation and Discretization," paper SPE 37979 presented at the 1997 SPE Reservoir Simulation Symposium, San Antonio, TX, June 1997.
23. Delshad, M., Pope, G.A., and Sepehrnoori, K.: "A Compositional Simulator for Modeling Surfactant Enhanced Aquifer Remediation, 1 Formulation," *J. Contamin. Hydrol.* (1996) 303.
24. Schlumberger Ltd., ECLIPSE Technical Description (2004).
25. Killough, J.E. and C.A. Kossack: "Fifth comparative simulation project: Evaluation of miscible flood simulations," paper SPE 16000 presented at the Ninth SPE Symposium on Reservoir Simulation, San Antonio, TX, February 1987.
26. Computer Modeling Group Ltd., GEM User's Guide (2004).
27. Uetani, T., B. Guler, and K. Sepehrnoori: "Reservoir Simulation on High Performance Clusters," presented at the Sixth World Multiconference on Systematics, Cybernetics, and Informatics, Orlando, FL, 14-18 July, 2002.
28. Dogru, A.H., K.G. Li, H.A. Sunaidi, W.A. Habiballah, L. Fung, N. Al-Zamil, D. Shin, A.E. McDonald, and N.K. Srivastava: "A Massively Parallel Reservoir Simulator for Large Scale Reservoir Simulation," paper SPE 51886 presented at the SPE Reservoir Simulation Symposium, Houston, TX, 14-17 Feb. 1999.