

A Parallel Multiblock Black-Oil Model in Multimodel Implementation

Qin Lu, Landmark Graphics Corp., and Malgorzata Peszynska and Mary F. Wheeler, SPE, U. of Texas at Austin

Summary

In this paper we discuss the multiblock algorithm for an implicit black-oil model as implemented in the multiphase simulator framework of IPARS (Integrated Parallel Accurate Reservoir Simulator). The multiblock algorithm decomposes the simulation domain into multiple nonoverlapping subdomains, or blocks, according to the geometric, geological, and physical/chemical properties, and well distribution. Each block can have its own grid system, and the grids of the neighboring blocks can be nonmatching on the interface, which allows for local grid refinement, or discrete fault or fracture modeling. Adjacent blocks are coupled across the interface by a set of conditions imposing a continuity of both primary variables and component mass fluxes that is realized through the use of special interface mortar variables. The resulting system is solved by an interface Newton procedure. Regularization techniques and preconditioners are proposed to improve the performance of the solver. The multiblock technique is effective and scalable, as shown by our numerical experiments. In addition, we present how the multiblock black-oil model has been used in the coupling of different physical models.

Introduction

The main thrust of this paper is to investigate accurate and efficient numerical techniques for the simulation of flow and transport phenomena in porous media, which are of major importance in the environmental and petroleum industries. We propose to emphasize a novel numerical methodology called the multiblock algorithm. This algorithm decomposes the simulation domain into multiple subdomains (blocks) according to their geological, geometric, and physical/chemical properties. One then applies the most efficient grid, numerical scheme, and physical model in each subdomain so that the computational cost is reduced and accuracy is preserved.

Multiblock (also known as macro-hybrid) formulations^{1–8} provide numerical models consistent with the physical and engineering description of the underlying equations. That is, the equations hold with their usual meaning on the subdomains, and physically meaningful conditions are imposed on interfaces between the subdomains. In particular, it is possible both to couple different discretizations on nonmatching multiblock grids and to couple different physical models in different parts of the simulation domain. These two features make the multiblock approach one of great computational interest.

In many applications, the geometry and physical properties of the domain or the behavior of the solution may require the use of different grids in different parts of the domain that might not possibly match on the interface. For example, the geology of the subsurface may involve the modeling of faults, pinchouts, and other internal boundaries. In such cases, the discontinuities of coefficients (e.g., mobilities) reduce the accuracy of traditional single-block algorithm near-discontinuities. By splitting the domain into multiple subdomains along the boundaries of discontinuities, solutions in each subdomain may have smooth properties, and local convergence rates are regained. Furthermore, locally refined grids may be needed for the accurate approximation of local phenomena such as high gradients around wells.

More generally, multiblock decomposition can be induced by differences in the physical processes and mathematical models or by differences in the numerical discretization models applied to different parts of the simulation domain.^{9–11} The overall computational cost can be reduced by selecting the most appropriate model in a given part of the reservoir. For example, only a single- or two-phase model is needed for the aquifer part of the reservoir, whereas a black-oil or compositional model is necessary if the gas phase is present in a subdomain.

In this paper, we discuss the formulation and implementation of a multiblock algorithm for an implicit black-oil model. This work represents a nontrivial extension of the multiblock algorithm for a two-phase oil-water model¹ as, in particular, it needs to address numerical regularization issues arising at phase transitions. Next, we briefly describe how the multiblock black-oil model is used in the multiphysics coupling with the two-phase oil-water model. We also address the issues that arise during implementation in the IPARS framework. In particular, we discuss the parallelism between the multimodel problem with the MPI multicommutator and model-based load balancing strategies. In the end, we present numerical experiments that demonstrate the scalability of our approach.

Multiblock Black-Oil Model

The 3D reservoir domain Ω is divided into a series of n_b nonoverlapping subdomains (blocks) Ω_k , $k = 1, \dots, n_b$, owing to geological faults,¹² geometry irregularities, variations of rock properties, and physical/chemical properties of flow, well types, their distribution, etc. Each block has a smooth rectangular grid. The grids are constructed locally and may be nonmatching on the interfaces between neighboring blocks. **Fig. 1** illustrates a typical geometry for a 2D domain decomposition. Note that the interfaces between blocks are filled with “mortars.” These are elements of a finite element space called mortar space, which is constructed on the 2D interface.

Each block or subdomain Ω_k has an associated local physical model/solver that could be a single-phase model, a two-phase oil/water model, a black-oil model, a compositional model, etc. These models can be implicit, semi-implicit or explicit in time, and they can use different linear and nonlinear solvers. In this paper we focus on the use of an implicit black-oil model as a subdomain solver. See Ref. 9 for a discussion related to various oil/water models coupled in the multiblock formulation.

Black-Oil Subdomain Formulation. The black-oil model is a three-phase (water, oil, and gas) model describing the flow in a petroleum reservoir.^{13,14} It is assumed that no mass transfer occurs between the water phase and the other two phases. In the hydrocarbon (oil/gas) system, only two components are considered. The oil component (stock-tank oil) is the residual liquid at atmospheric pressure left after differential vaporization, leaving the gas component as the remaining fluid.

One should distinguish the difference between phase and component.^{13,15} For water, these two concepts are the same in the model because, by the above assumption, no mass transfer occurs between the water phase and the others. As in the above definition, oil and gas components refer to those hydrocarbon mixtures that would be in liquid (usually heavy hydrocarbons) and gaseous (usually light hydrocarbons) states, respectively, after surface separation. The corresponding pressure and temperature are normally called standard conditions, which are usually (but not always) 14.7

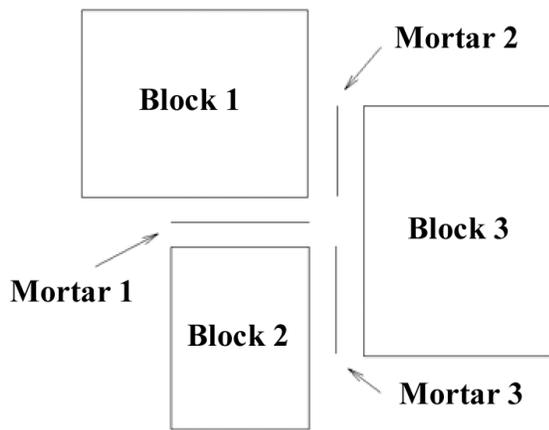


Fig. 1—2D multiblock domain decomposition.

psi and 60°F. The concept of phase would be meaningful only under a specific pressure and temperature.

To avoid confusion, we use capital letter subscripts to denote water, oil, and gas components (W , O , and G , respectively) and lower-case letter subscripts to denote water, oil, and gas phases (w , o , and g , respectively). Mass conservation for water, oil, and gas components are given by

$$\frac{\partial(\Phi N_w)}{\partial t} = -\nabla \cdot U_w + q_w, \dots \dots \dots (1)$$

$$\frac{\partial(\Phi N_o)}{\partial t} = -\nabla \cdot U_o + q_o, \dots \dots \dots (2)$$

$$\frac{\partial(\Phi N_g)}{\partial t} = -\nabla \cdot (U_g + R_o U_o) + q_g, \dots \dots \dots (3)$$

Note that the gas component flux consists of both the gas phase (free gas) and the gas dissolved in the oil phase with a gas/oil ratio denoted as R_o . At certain pressures (above bubblepoint) no free gas can exist, and only two-phase (water/oil) conditions exist. In such conditions, N_g is proportional to N_o through the gas/oil ratio for the given pressure. On the other hand, if N_g is bigger than $R_o N_o$ at a given pressure, then three phase conditions exist.

Darcy's law for multiphase flow is used to calculate the mass velocity of component M in its corresponding phase m

$$U_M = -\frac{Kk_{rm}}{B_m \mu_m} (\nabla P_m - \rho_m g \nabla D), \dots \dots \dots (4)$$

Consistent with volume balance, the saturations must satisfy the constraint

$$S_w + S_o + S_g = 1, \dots \dots \dots (5)$$

Capillary pressures are defined as functions of saturation in the following way:

$$P_{cow} = P_o - P_w, \dots \dots \dots (6)$$

$$P_{cgo} = P_g - P_o, \dots \dots \dots (7)$$

The above system of equations is discretized for a numerical solution; the subdomain is covered by a tensor product grid, and the equations are discretized using cell-centered differencing in space and backward Euler differencing in time with P_w , N_o , and N_g chosen as primary unknowns.¹¹ Upwinding is applied to cell edge values for numerical stability.^{13,16} We recall that the cell-centered differencing in space is equivalent to the expanded mixed-finite element method of the lowest order Raviart Thomas space (RT0) on a rectangular grid, provided certain quadrature rules are applied.^{3,4,17} The system of nonlinear algebraic equa-

tions is linearized by Newton's method. A GMRES solver with multilevel preconditioner^{18,19} or an LSOR solver is available to solve the linear system at every Newtonian step.

Interface Conditions for Black-Oil Model. The above subdomain black-oil model equations hold in the interior of each of the blocks. These equations need to be complemented by conditions ensuring mass and momentum conservation across the interface.

In this paper, for simplicity, we assume that the rock type is the same for all subdomains; therefore, the same capillary pressure data is used. As a consequence, the pressure of each phase is continuous. In the multiblock formulation, in order to impose the continuity of pressure and to ensure the mass conservation, we need to make the pressures and the mass fluxes corresponding to two adjacent subdomains match on the interface. Specifically, on each interface Γ_{kl} , $1 \leq k < l \leq n_{bl}$, we impose

$$P_m|_{\partial\Omega_k} = P_m|_{\partial\Omega_l}, \quad m = w, o, g, \dots \dots \dots (8)$$

$$[U_M \cdot \nu]_{kl} \equiv U_M|_{\partial\Omega_k} \cdot \nu_k + U_M|_{\partial\Omega_l} \cdot \nu_l = 0, \quad M = W, O, G, \dots \dots (9)$$

This approach is based on a domain decomposition algorithm for single-phase flow developed originally for conforming grids,² later generalized to nonmatching grids coupled with mortars,^{6,7} and used for two-phase oil/water models.^{1,9} Note that as a consequence of Eq. 8 and the assumed homogeneity of rock type, if the pressure of one phase is continuous, then the saturations and densities of all phases are continuous. Thus, in order to express momentum conservation, instead of pressures, as in Eq. 8, we can match a different set of variables, such as the values of the primary unknowns in the black-oil model:

$$P_w|_{\partial\Omega_k} = P_w|_{\partial\Omega_l}, \dots \dots \dots (10)$$

$$N_o|_{\partial\Omega_k} = N_o|_{\partial\Omega_l}, \dots \dots \dots (11)$$

$$N_g|_{\partial\Omega_k} = N_g|_{\partial\Omega_l}, \dots \dots \dots (12)$$

In fact, it is this set of conditions that is used in the discretization of Eqs. 8 and 9. We note, however, that the assumptions of Eqs. 11 and 12 are only used for simplicity and that the general case of different rock types is easily handled.

In discrete form, the conditions in Eq. 8 and 9 are imposed in the weak (average) sense using the mortar spaces as follows: let M_h denote the mortar space on the union of interfaces

$$\Gamma = \bigcup_{1 \leq k < l \leq n_{bl}} \Gamma_{kl},$$

and let $M_h = (M_h)^3$. The mortar space is constructed over a mortar grid using piecewise polynomial finite element shape functions. It was shown theoretically and numerically^{3,5,6} that, for matching grids, we may use the same space on the interface as the normal trace of the velocity space in subdomain; however, in the case of nonmatching grids, in order to preserve optimal convergence (and, in some cases, superconvergence), the mortar space must consist of piecewise polynomials of one degree higher than the normal trace of the velocity space in the subdomains. Because we use cell-centered finite differences in space in each subdomain and the normal trace of the velocity space is piecewise constant, we employ a piecewise linear space M_h on the interface.

The discrete equivalents of Eqs. 1 through 7, which are defined on unions of all blocks with Eqs. 9 through 12, imposed on the interface, are written as a nonlinear interface problem to be solved by iteration for the values of interface primary unknowns $\psi = (P_{h,w}, \bar{N}_{h,o}, \bar{N}_{h,g}) \in M_h$. The nonlinear equations evaluate the jump in fluxes $U_{h,M}$ corresponding to ψ as

$$B(\psi, \mu) = \sum_{1 \leq k < l \leq n_{bl}} \int_{\Gamma_{kl}} \left\{ \sum_M^{w, o, g} [U_{h,M}(\psi) \cdot \nu]_{kl} \mu \right\} d\sigma, \dots \dots (13)$$

To achieve mass conservation (or flux continuity), we need to determine ψ such that the jump in the fluxes is

$$B(\psi, \mu) = 0, \quad \forall \mu \in M_h, \dots \dots \dots (14)$$

We can also write $B(\psi, \mu) = (B(\psi), \mu)$, where the physical meaning of $B(\psi)$ is the square of the jump of the fluxes across all the interfaces corresponding to the given value of ψ , which provides a boundary condition for each phase/component. Note that a subdomain solver delivers the value of $U_h = (U_{h,w}, U_{h,o}, U_{h,g})$ at each block k .

The solution to the interface problem in Eq. 14 is equivalent to the solution on all subdomains with mass and momentum preserved (weakly) across the interface.

Interface Solver. The system of nonlinear equations on the interface in Eq. 14 is, in our implementation in IPARS, solved by an inexact Newton method.^{1,7,18} Each Newtonian step solution is computed by a forward difference GMRES iteration. Each GMRES iteration involves solving subdomain problems with the Dirichlet boundary condition provided by the interface code. The subdomain problems in turn deliver values of fluxes across each interface. Additionally, two inexpensive projection steps are necessary: the first one projects ψ from the interface mortar grid onto the local subdomain grids so that it may be used as the Dirichlet value, and the second one projects normal fluxes U_h computed by the subdomain solver from the local grids onto the mortar grid. Details of the algorithm can be found in Ref. 1, 5, 7, and 18.

The interface algorithm is an iterative procedure. It assumes an initial guess for the values of primary variables ψ on the interface to be used after projection as a Dirichlet boundary condition on subdomains. The subdomain problems are then solved, and the jump in the component fluxes $B(\psi)$ is calculated on the interface. If the jump is less than a given tolerance, the current timestep is completed; otherwise, the interface unknowns ψ are updated and the procedure is repeated until convergence.

To facilitate the above steps in implementation, additional memory needs to be allocated for each block to store boundary layer information such as the values of primary variables, transmissibilities, component mass fluxes, and so on. Analogous to the interior cells, upwinding is applied to the boundary values for flux calculations. One advantage of this approach is that the Dirichlet boundary conditions can be included in both the Jacobian matrix and the residuals of the block without modifying the subdomain code.¹¹

Regularization of Gas-Phase Relative Permeability on Interface.

One major issue that arises in modeling on the interface of the three-phase black-oil model is the treatment of the gas phase when it reaches the interface. The gas component consists of both free gas and gas dissolved in the oil phase. Either the gas phase can be immobile, when gas saturation is less than the residual gas saturation, or the gas phase can be mobile, when the gas saturation is greater than the residual value.^{15,20} The former situation may arise when the interface is near production wells. In this case, it is assumed that no gas phase exists initially on or near the interface. But the gas phase may be created later near the production wells because of a pressure drop. The gas phase zone then expands as a function of time, and eventually may reach the interface. Note that the gas phase is created because of the pressure drop and not as a result of transport.

Because of the nonzero residual gas saturation, when the front of the gas phase zone first reaches the interface, the physical problem becomes degenerate, and the Jacobian matrix on the interface is no longer diagonally dominant. Consequently, the interface solver may not converge to a physical solution, and the simulation may fail.

Fig. 2 is a typical gas relative permeability curve, where S_{gr} is the residual gas saturation. When the gas saturation S_g increases gradually from zero to the residual gas saturation S_{gr} , the concentration of the gas component increases on the interface, while the gas component mass flux (and therefore the jump in this flux across the interface) remains unchanged. Because neither the gas component mass flux with free gas (or gas phase) nor that with the dissolved gas changes (note that the amount of gas component dissolved in oil phase is unchanged since the pressure is un-

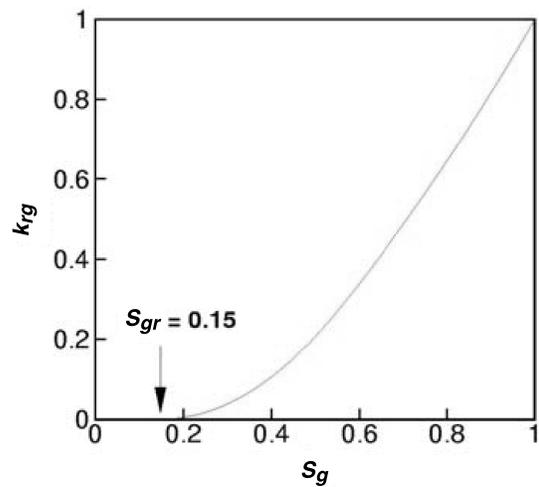


Fig. 2—A typical gas relative permeability curve.

changed), the change of the jump in the gas flux is zero on the interface. In this case, we have

$$\frac{\partial[U_G \cdot v]}{\partial N_G} = 0, \dots \dots \dots (15)$$

where $[U_G \cdot v]$ denotes the mass flux jump of the gas component across the interface. Numerically, this partial derivative is a diagonal term in the interface Jacobian matrix. The zero diagonal term makes the matrix no longer diagonally dominant.

Our solution to this problem is to regularize the gas relative permeability curve.¹¹ We enforce a small slope dk_{rg}/dS_g between $S_g = 0$ and $S_g = S_{gr}$, as shown in the right plot of Fig. 3, so that the derivative in Eq. 12 will be greater than zero. As a result, the interface Jacobian matrix will be better conditioned.

The admissible sizes of the slope are case-dependent. It is necessary to choose a slope sufficiently large for stabilization but not so large that the physics of the problem changes. Our experience indicates that the recommended range would be between 10^{-2} to 10^{-4} . Fig. 4 shows the comparison of the water pressure distribution of a 1D case when the problem is simulated with a single block (no interface) and with the multiblock (two blocks connected with an interface). Fig. 5 shows a comparison of gas saturation distributions. The dimensionless length of the case was 1, with grid size of $50 \times 1 \times 1$. The initial gas saturation was set to zero, and the gas relative permeability is shown in Fig. 2, with a residual gas saturation $S_{gr} = 0.15$. A production well was set at $x_d = 0.05$, an injection well at $x_d = 0.95$, and the interface at $x_d = 0.2$. The results were taken after 100 days of simulation. As we see, the slopes ranging from 10^{-2} to 10^{-4} work well. But with slopes beyond this range, the simulation crashed on the interface because of bad primary variable values.¹¹

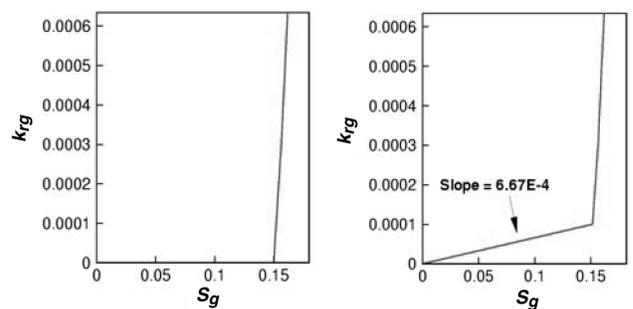


Fig. 3—Gas relative permeability curve near residual gas saturation. Left: initial curve. Right: regularized curve.

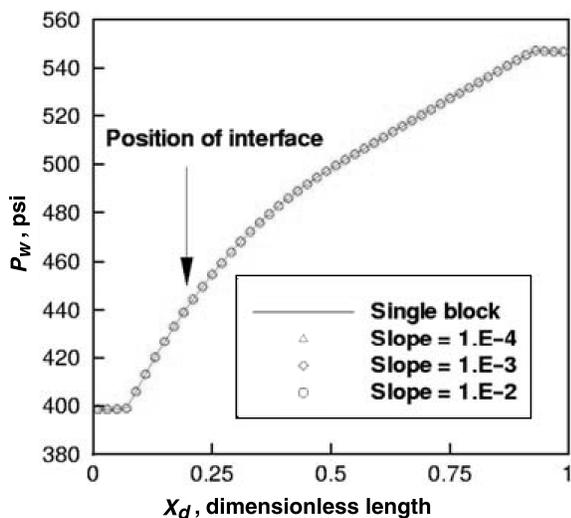


Fig. 4—Effect of gas relative permeability regularization on water pressure distribution of a 1D case with two blocks.

Note that the regularization is only necessary on the interface. In the subdomains, the original gas relative permeability is used.

Coupling Different Physical Models

In a given part of the reservoir, we may want to individually select the most appropriate physical model to be coupled to other parts and their assigned models through the multiblock algorithm. Each model should be as simple as possible, yet it should include necessary elements to describe the relevant physical and chemical phenomena of the flow and transport in the associated subdomain. This strategy may lead to a significant reduction in the overall computation, especially for multimillion grid size cases, though the accuracy of the simulation is still retained.^{10,11}

For example, in the aquifer part of the reservoir—or in the part with a water/oil two-phase system and no free gas created during the whole operation—a two-phase code is appropriate. In the part with gas cap, or with production wells around which free gas may be created because of low bottomhole pressure, a black-oil or a compositional code is necessary. Here we discuss the coupling of the implicit black-oil model and an implicit two-phase oil/water model. In this coupling it is essential that conservation quantities be preserved across the interfaces. With the black-oil model discussed above, we briefly review the oil/water model.

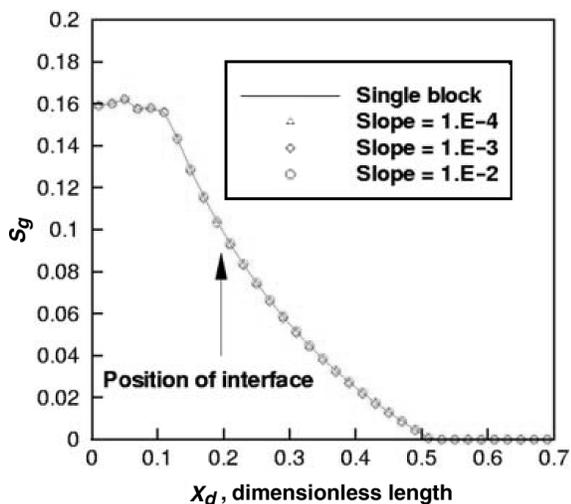


Fig. 5—Effect of gas relative permeability regularization on gas saturation distribution of a 1D case with two blocks.

Oil/Water Model. The immiscible two-phase oil/water model is, in some sense, a subset of the black-oil model in which the gas phase and component are absent. It is governed by the equations

$$\frac{\partial(\Phi N_w)}{\partial t} = -\nabla \cdot U_w + q_w, \dots \dots \dots (16)$$

$$\frac{\partial(\Phi N_o)}{\partial t} = -\nabla \cdot U_o + q_o, \dots \dots \dots (17)$$

Note that here we use purposely the phase subscripts instead of component subscripts for concentrations and fluxes. In the immiscible model, these two concepts are equivalent. This different notation allows us to easily differentiate between the black-oil model and the two-phase model.

Darcy's law for multiphase flow is used to calculate the mass velocity of phase m :

$$U_m = -\frac{Kk_{rm}}{\mu_m} \rho_m (\nabla P_m - \rho_m g \nabla D), \dots \dots \dots (18)$$

The saturations must satisfy the constraint

$$S_w + S_o = 1. \dots \dots \dots (19)$$

In this model, the phases are slightly compressible, so densities are assumed to depend on the pressure exponentially. Finally, capillary pressure is defined as function of saturation in the following way:

$$P_{cow} = P_o - P_w, \dots \dots \dots (20)$$

Oil-Phase Partitioning Across Interface. A natural question arises from the coupling of these two models: how should the gas component on the interface be dealt with now that on one side of the interface is a three-phase black-oil model and on the other side is a two-phase oil/water model that assumes no gas phase? The natural strategy is as follows: select subdomains appropriately and partition the oil phase on the interface into oil and gas components.¹¹

In the decomposition into subdomains, we assume that in the time period being simulated no gas phase (free gas) appears on any interface. This can be achieved by a careful decomposition of the domain. In other words, the interfaces between black-oil models and two-phase oil/water models are selected to be below the gas cap and a sufficient distance away from the production wells in which free gas may be created because of pressure drops. This means that the reservoir pressure is above the bubblepoint pressure¹⁵ on and near the interfaces. However, the gas component dissolved in the oil phase is still allowed to exist on interfaces. The selection of domains can of course be adaptively changed to meet these two criteria.

The oil/water model describes the flow of water and oil phases, but does not distinguish between the oil-phase and the oil component, while the black-oil model takes the oil component and the gas component in a liquid state as the oil-phase. Therefore, our second strategy involves the splitting of the oil phase on the interfaces into oil and gas components. This is achieved by constructing appropriate correlations relating component with phase. No difficulties arise in the aqueous phase, because the water phase consists only of the water component in both models.

We define the oil phase in the oil/water model as saturated oil with a constant nonzero gas/oil ratio R_o that can be chosen to be close to the solution gas/oil ratio R_{so} . Therefore, on the interface between these two models, we need to partition the oil phase concentration N_o of the oil/water model into the oil component concentration N_o and the gas component concentration N_g in order to match the values of the black-oil model, according to the mass fractions of these two components in the oil phase. Without considering unit conversions at this point, we define

$$N_{w|oD}^B = N_{w|oD}^H, \dots \dots \dots (21)$$

$$N_{O|\partial D}^B = N_{O|\partial D}^H \frac{\rho_{OS}}{\rho_{OS} + \rho_{GS}R_o}, \dots \dots \dots (22)$$

$$N_{G|\partial D}^B = N_{O|\partial D}^B R_o \dots \dots \dots (23)$$

Multimodel Interface Coupling. We may assume that the oil phase in the oil/water model is saturated, with a gas/oil ratio R_o that stays approximately constant during the simulation. Consequently, we may assume R_o to be constant on the interface between the oil/water model and the black-oil model. Therefore, the value of N_G on the interface can be calculated from N_O of the black-oil model as the gas/oil ratio R_o is assumed to be constant on the interface.

The primary variables of the subdomain models—the implicit oil/water model and the implicit black-oil model—as well as those on the interface are summarized in **Table 1**. The primary variables on the interface—water pressure P_w and oil-phase concentration N_o —are chosen to be something between the two subdomain models. We only solve for two primary variables, because there are only two phases existing on the interface and also because the gas component is assumed to be in a constant proportion to the oil component. Therefore, we only match the oil component and the water component fluxes across the interface. The gas component flux is matched automatically because the gas/oil ratio is constant on the interface.

The interface formulation can be summarized as follows: solve Eq. 14 where, for $\psi = (\bar{P}_w, \bar{N}_o) \in M_I$, the primary variables on the interface, we define

$$B(\psi, \mu) = \sum_{1 \leq k < l \leq n_{bl}} \int_{\Gamma_{kl}} \left\{ \sum_M^{w,o} [U_{h,M}(\psi) \cdot v]_{kl} \mu \right\} d\sigma. \dots \dots \dots (24)$$

The system of nonlinear equations on the interface is solved by an inexact Newton method as explained previously.

Similar to the splitting of oil phase concentration on the interface, we need to split the oil phase flux of the oil/water model into the oil component flux and the gas component flux to match the black-oil model, using the correlations analogous to Eqs. 21 through 23.

Implementation Issues

All implementation aspects and results presented in this paper are built upon the framework of IPARS (Integrated Parallel Accurate Reservoir Simulator),^{1-8,21} which has been developed at the Center for Subsurface Modeling at the U. of Texas at Austin. It is a new generation framework for developing and running parallel reservoir simulators, solving problems involving a million or more grid elements, and supporting a variety of physical and numerical models plus simplified well management. This framework provides an infrastructure that is common to most simulators, such as parallel message passing, memory management, grid generation, free-form keyword input, formatted output for visualization, user units specification, general purpose 2D function utilities, table lookup, and so on, so that the developer only needs to code the physical model of interest.

The MPI portable message passing interface²² is currently used to manage the message passing in parallel computations. The framework is portable, running on platforms such as IBM and SGI

TABLE 1—PRIMARY VARIABLES FOR THE SUBDOMAIN MODELS AND INTERFACE COUPLING IMPLICIT OIL/WATER MODEL AND BLACK-OIL MODEL		
Oil/Water Model	Black-Oil Model	Interface
P_O	P_w	\bar{P}_w
N_O	N_O	\bar{N}_O
	N_G	

workstations, Cray T3E, IBM SP, single PC under Linux, Windows and DOS, and PC clusters under Linux.

The framework can have multiple (fault) blocks (or subdomains), each of which may have its own physical model associated with it. The neighboring blocks are connected via an interface using either the mortar spaces approach presented here or the dual approach discussed in Ref. 8. Unit conversion between different models may be necessary in the coupling of different models.

The linear solvers for different physical models can be either different or the same. A parallel GMRES solver^{18,19} has been extended for solving multiple models simultaneously. The basic idea for the extension is to expand the workspace (e.g., the basis for the Krylov space used in solving a linear system¹⁸) from a scalar to an array, so that each model has its own entry in the workspace.

The parallelism is a delicate and interesting issue to tackle. Unlike the traditional single model simulator, the multimodel problem is actually an MIMD (multiple instruction multiple data) problem. We use multiple MPI communicators in the implementation.^{11,22} The processors are split into multiple groups (or communicators) so that each physical model has its own communicator. Within a communicator, the adjacent processors can exchange the boundary information that is necessary for parallel computation. Message passing between different communicators is also allowed.

Load balancing is important for parallel efficiency. The traditional (single-model) load-balancing strategy divides the grid cells evenly between processors. In multimodel cases, a processor may be shared by more than one model. This may cause a loss of parallel scalability because of a synchronization problem between different models. Therefore, we suggest using two criteria to improve load balancing (or model-based load balancing) in multimodel implementation.⁹⁻¹¹

First, because of synchronization issues existent between different communicators, if possible, one processor should never handle more than one model/code. Second, the number of processors assigned to each model should be weighted by the speed of the simulation, provided that the first criterion is satisfied. **Fig. 6** demonstrates a comparison of the speedup with these two strategies for a three-block case coupling the black-oil model and the oil/water model. With model-based load balancing strategies, ideal, or even superlinear speedup, can be obtained.

Computational Examples

Example 1: The First Multiblock Black-Oil Case. Our first example was the horizontal reservoir with geometry as shown in **Fig. 7**. One injection well and one production well were completed at the two ends of the reservoir with piecewise linearly changed bottomhole pressure with respect to the simulation time. The size

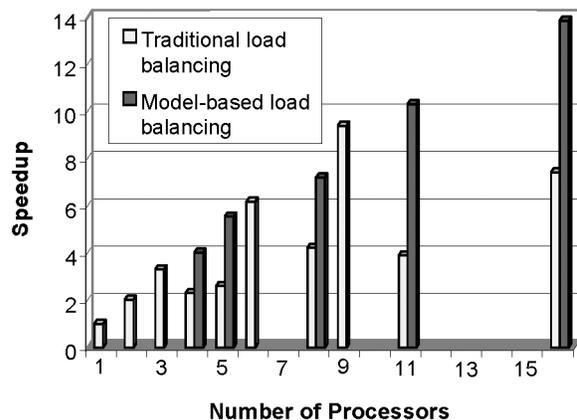


Fig. 6—Parallel multimodel speedup for coupling black-oil model and oil/water model.

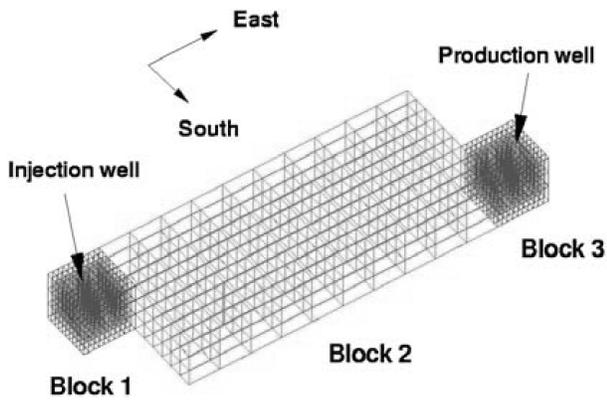


Fig. 7—The geometry and grid system of the reservoir used by Example 1.

of the reservoir was 1,600×500×15 ft, in two horizontal directions and one vertical direction, respectively (we will follow the same order in later discussions). The original water pressure was 2000 psi at the top of the reservoir. The permeability field was layered, with 100 md and 50 md in the horizontal and vertical directions, respectively, at the top and the bottom layers. The horizontal permeability was 50 md at the middle layer. The rock at the northeast and southwest corners was impermeable. The reservoir was split into three blocks according to geometry and well locations. At the blocks (Blocks 1 and 3) containing wells, which were at southeast and northwest corners, we applied a fine grid in each block, with grid size 10×10×3. The size of each grid cell was 20×20×5 ft. Block 2, the largest of the three blocks, contained no wells, and the flow in this block was smooth; therefore, we used a coarse grid of size 10×10×3. Here each grid cell was of size 160×50×5 ft.

Two interfaces were set between the neighboring blocks with the grid size of the mortar space as 1×3 (horizontal direction and vertical direction, respectively). A continuous piecewise linear basis was applied in the mortar space on the interface.

We ran this case with the multiblock black-oil code. It normally took two Newtonian steps on the interface and 18 subdomain evaluations, on average, per timestep. We observed that the jump in fluxes across the interface decreased quadratically with the Newtonian interface iterations, which is optimal for Newton's method.

For comparison purposes, we also ran this case using the single block black-oil model (no interface) with fine grid (20×20×5 ft) over the entire reservoir, with some of the elements keyed out at the corners to match the original geometry. Figs. 8 and 9 show the

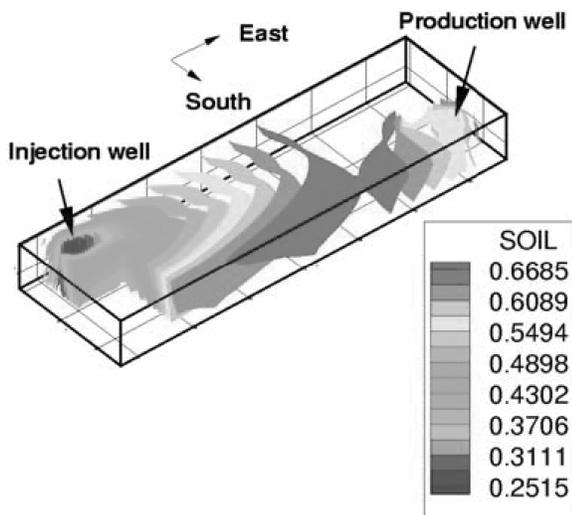


Fig. 9—Oil saturation contour for the horizontal reservoir at the 1,600th day given by single-block black-oil model.

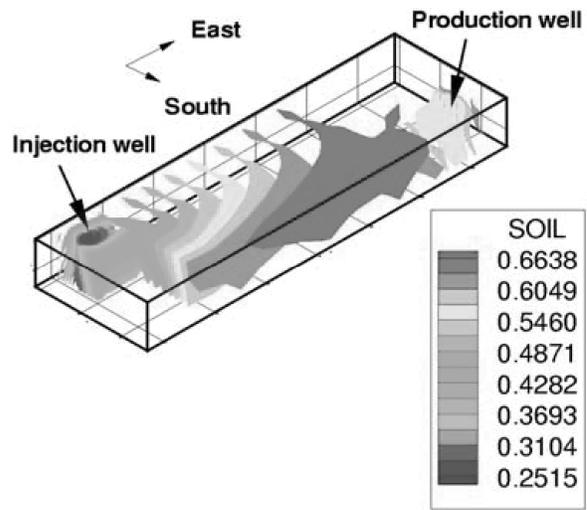


Fig. 8—Oil saturation contour for the horizontal reservoir at the 1,600th day given by multiblock black-oil model.

oil saturation contours obtained from these two approaches. Figs. 10 and 11 compare the gas saturation profiles between these two approaches.

We observed that both the oil and gas fronts cross the interfaces without any difficulties; gas relative permeability regularization is necessary here. The contours from these two approaches look very similar, though they are not exactly the same. The small difference arises from two sources. First, there is a small local discrepancy between the numerical results given by the interface algorithm and those given for the single block. This is because the flux and pressure continuity is imposed in a weak sense only and because the interface Newtonian iteration stops when some nonzero tolerance criterion is satisfied. Second, the graphic tool (Tecplot²³) renders the contours separately for each block; therefore, even continuous functions when shown on multiblock grid may appear to have discontinuous contours.

Another way to compare the results of different simulations is presented in Fig. 12, which shows the injection and production flow rates obtained from the multiblock and single-block simulators. The two simulators match each other perfectly.

Finally, we consider the computational efficiency of the multiblock approach. As we can see in Table 2, the multiblock simulation ran faster than the single-block black-oil model simulation

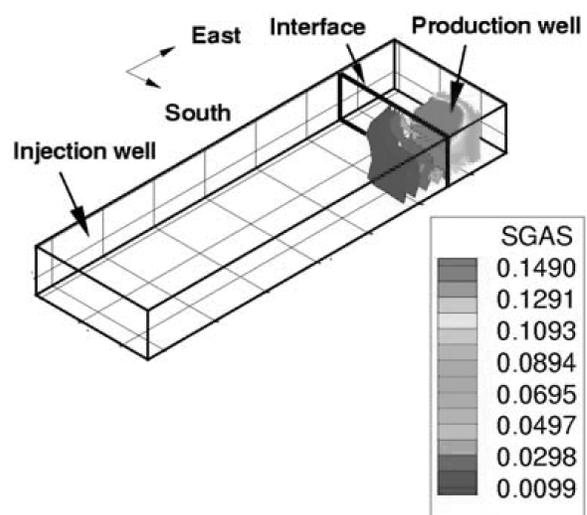


Fig. 10—Gas saturation contour for the horizontal reservoir at the 1,600th day given by multiblock black-oil model.

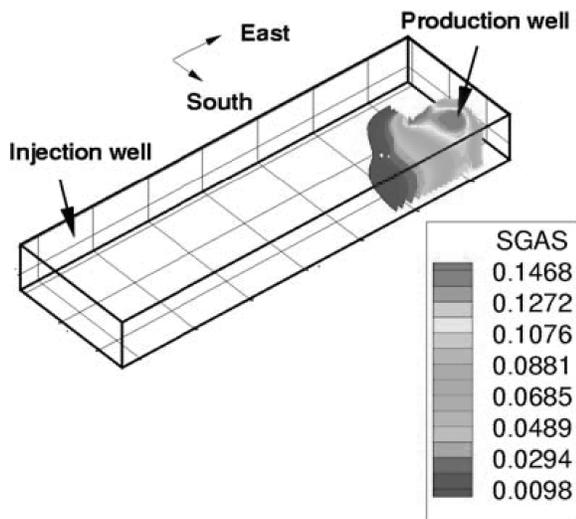


Fig. 11—Gas saturation contour for the horizontal reservoir at the 1,600th day given by single-block black-oil model.

by a factor of 1.43. Concluding this example, we note that by using the multiblock algorithm to handle local grid refinement and coarsening, we reduced the computational time spent without sacrificing accuracy.

Example 2: The Second Multiblock Black-Oil Case. This was a dipping rectangular reservoir, the geometry and permeability distributions for which are shown in Fig. 13.

The size of the reservoir is 880×880×3 ft. It has a dip angle $\alpha = 2.9^\circ$ from a horizontal direction. It is characterized by a very high permeability contrast; the permeability of most of the domain was 2 md in all directions. There was, however, a high permeability streak along the two sides of the reservoir with a permeability of 500 md, except for a layer at the middle of the streak, which had a permeability of 100 md. An injection well and a production well were located at the two opposite corners of the high-permeability streak. The initial water pressure and water saturation at the top of the reservoir were 2,300 psi and 0.22, respectively.

For this case, it was desirable to apply a fine grid in the regions with high permeability and wells and use a coarse grid in the remaining part of the reservoir. Therefore, we split the reservoir into three blocks with three interfaces set between neighboring blocks as in Fig. 13. The size of a fine-grid cell was 20×20×5 ft, and the size for a coarse-grid cell was 80×80×10 ft.

The mortar grids between Blocks 1 and 2, and between Blocks 1 and 3, were 3×1 (horizontal direction and vertical direction, respectively). The grid size of the mortar space between Blocks 2 and 3 was 1×1. The basis for the mortar spaces was taken to be continuous piecewise linear. The oil saturation profiles at the 1,000th day are shown in Fig. 14.

Again, we simulated the same problem with the single-block black-oil model for comparison. We had to use a fine grid all over the reservoir. As a result, the multiblock approach was almost twice as fast as the single-block approach, as shown in Table 3, while the well flow rates showed excellent matching as shown in Fig. 15.

Example 3: Multimodel Case. This is a reservoir with a geometry suitable for splitting into three blocks with two interfaces between

	Multiblock Black-Oil	Single-Block Black-Oil
Total CPU time (seconds)	2,438	3,486

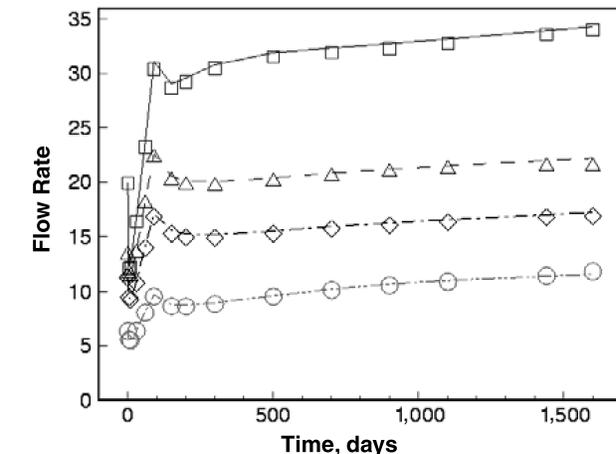
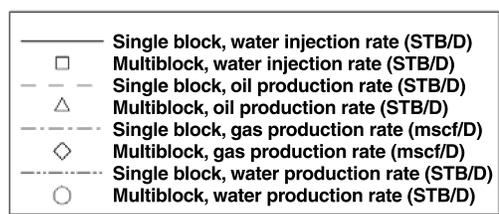


Fig. 12—Comparison of injection and production flow rates for Example 1 given by multiblock and single-block black-oil model.

neighboring blocks. The domain decomposition, gridding, permeability distribution, well locations, and assignment of physical models are shown in Fig. 16. The two interfaces were between Blocks 1 and 2 and between Blocks 1 and 3, respectively. The fault between Blocks 2 and 3 was impermeable. The size of Block 1 was 800×800×48 ft (two horizontal directions and one vertical direction, respectively); the size of Block 2 was 9,600×6,400×48 ft and 6,400×9,600×48 ft for Block 3. Grid sizes of the three blocks were 20×20×8, 18×13×4, and 13×19×4, respectively. The grid cell size in Block 1 was chosen as relatively small (40×40×8 ft per cell), because this block represented the production zone containing seven wells. In contrast, a coarse grid (160×160×16 ft per cell) was used in most of Blocks 2 and 3, except around four injection wells (80×80×16 ft per cell). The total number of grid cells was 5,124.

There were 11 wells in all: two oil-rate specified production wells, five bottomhole pressure-specified production wells, and four bottomhole pressure-specified water injection wells used to maintain the pressure. The permeability was layered as shown in Fig. 16 with light color denoting high permeabilities (200 md) and dark color marking low permeabilities (20 or 50 md). Block 1 had high initial oil saturations, ranging from 0.5 to 0.8, with a gas cap dominating the top 8 ft, while, for the most part, Blocks 2 and 3 were aquifers. The initial oil pressure at the top of the reservoir was 2,000 psi.

According to the geological and geometric characteristics of the reservoir and the well types and locations, we assigned the

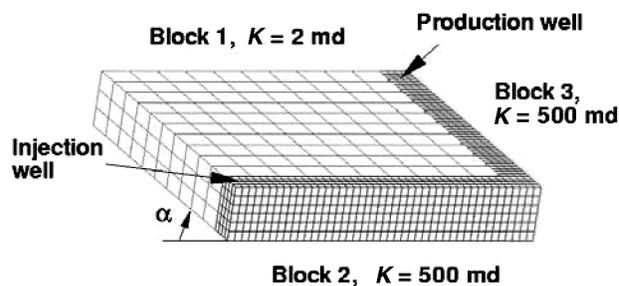


Fig. 13—Geometry, permeability distribution, wells location, multiblock decomposition, and gridding of the dipping reservoir used by Example 2.

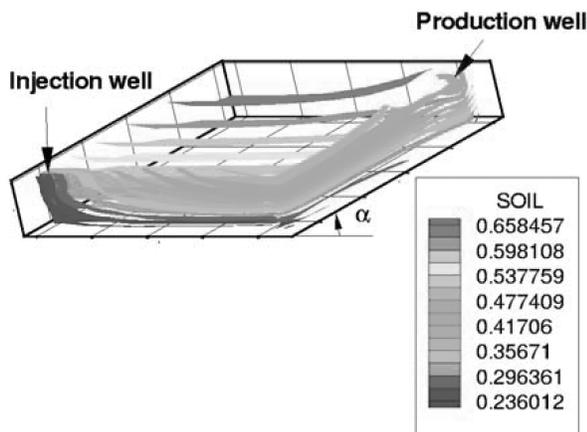


Fig. 14—Oil saturation profile at the 1,000th day given by multi-block black-oil model.

black-oil model in Block 1 and the oil/water model in Blocks 2 and 3.

We assumed that the oil phase in the oil/water model and on the interface was saturated with a constant gas/oil ratio, which was about 90% of the solution gas/oil ratio under the average pressure on interface.

We used the minimum mortar grid, which is 1x1. It usually took two Newtonian iterations on the interface to converge to the given tolerance. The jump in fluxes decreased quadratically. The optimal convergence rate for Newton's method when a good enough initial guess is given is quadratic as well. The initial timestep was 0.5 day, and it quickly increased to 20 days.

The oil component concentration profiles at the 500th day obtained from the multimodel are shown in Fig. 17. The effects of production and waterflooding are obvious in that the water front has passed through the interfaces, and the gas cap shrank back to the far corner of Block 1, away from the other two blocks.

For comparison purposes, we simulated the case of the single-block black-oil model with a uniform fine grid (40x40x8 ft for each cell) in the entire reservoir. The total number of grid cells was 41,600 in this case. The well flow rates are compared in Fig. 18. The total oil production rate and total gas production rate got excellent agreements between the multimodel and the single-block black-oil model. However, the total water production rate and total water injection rate had about 5% difference between the two approaches, which was expected, because the sizes of the grid cells around the injection wells were different between these two simulations. The fact that the injection wells were handled by the oil/water model in the multimodel approach, and that they were handled by the black-oil model in the single-block black-oil model approach, may have contributed to part of the differences.

As shown in Table 4, the multimodel simulation achieved a dramatic reduction in computational time by a factor of 7.07 compared to the single-block black-oil model. This is clearly an effect of using the faster model (oil/water model) in two larger blocks combined with grid coarsening there. Without the latter, the cost reduction could be less dramatic.

Conclusions

1. The multiblock algorithm is a promising method to couple appropriate grids and physical models based on decomposing the simulation domain into multiple subdomains (blocks) according

	Multiblock Black-Oil	Single-Block Black-Oil
Total CPU time (seconds)	2,773	5,467

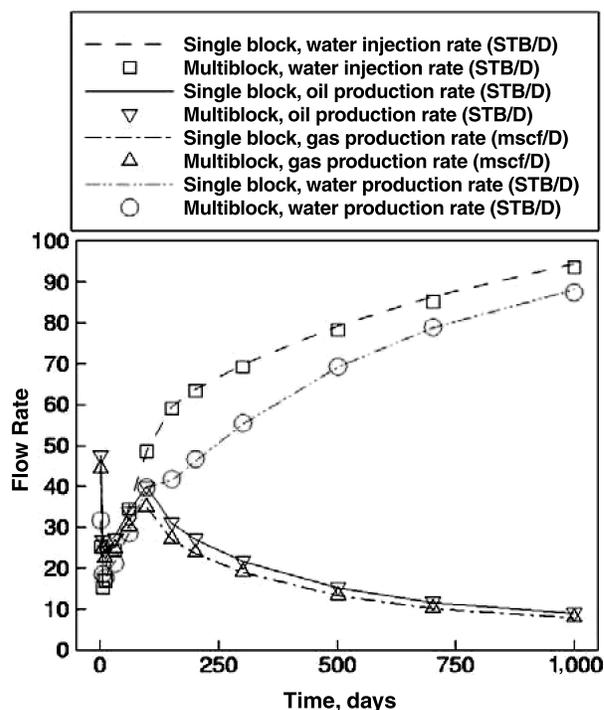


Fig. 15—Comparison of injection and production flow rates given by multiblock and single-block black-oil model for the case used by Example 2.

to the geological, geometric, and physical/chemical properties. The flux continuity across the interface between adjacent blocks is achieved accurately. Through a comparison study between the multiblock/multimodel and the traditional single-block/single-model, it has been demonstrated that the computational cost is reduced and accuracy is preserved when multiblock couplings are used.

2. In the multiblock black-oil model, the regularization of the gas relative permeability is applied on the interface to avoid the degeneration of the physical problem that may occur when the gas phase saturation on the interface is greater than zero but less than the residual value.
3. In the coupling of the implicit black-oil model and the implicit oil-water model, proper domain decomposition is needed to ensure that the gas phase never reaches the interface in the time period being simulated. Oil phase partitioning is applied to match the oil component and gas component, respectively, across the interface between the two models.
4. The MPI multicommutators have been used in parallel implementation of the multimodel problem. The processors are di-

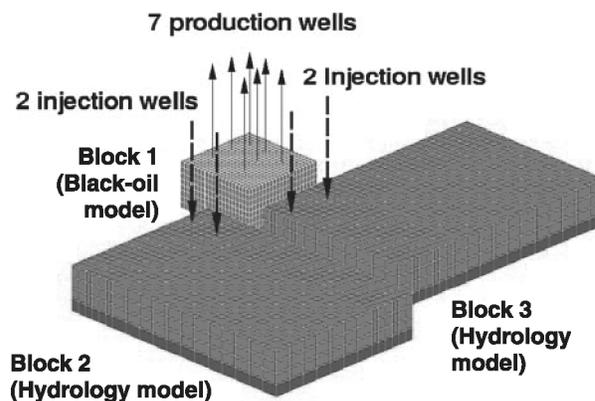


Fig. 16—Geometry, domain decomposition, gridding, permeability distribution, wells locations, and assignment of models of the reservoir used by Example 3.

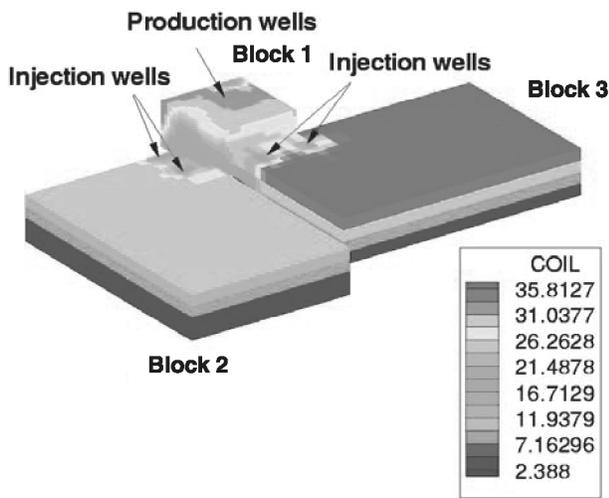


Fig. 17—Oil component concentration profile of the reservoir with three blocks at $t = 500$ days given by multimodel.

vided into multiple groups, or communicators, so that different groups can run different models simultaneously. This implementation allows an arbitrary number and combination of models, and it is independent of the interface code. Model-based load-balancing strategies have been applied to achieve ideal speedup; in some cases, even superlinear speedup can be obtained.

Nomenclature

B_m = formation volume factor of phase m , bbl/STB
 $B(\psi, \mu)$ = the jump in fluxes across the interface, L/t, ft/sec
 $d\sigma$ = differential of surface area, L^2 , ft^2
 D = depth, L, ft
 g = gravity magnitude, L/t^2 , psi-ft²/lb
 k_{rm} = relative permeability of phase m

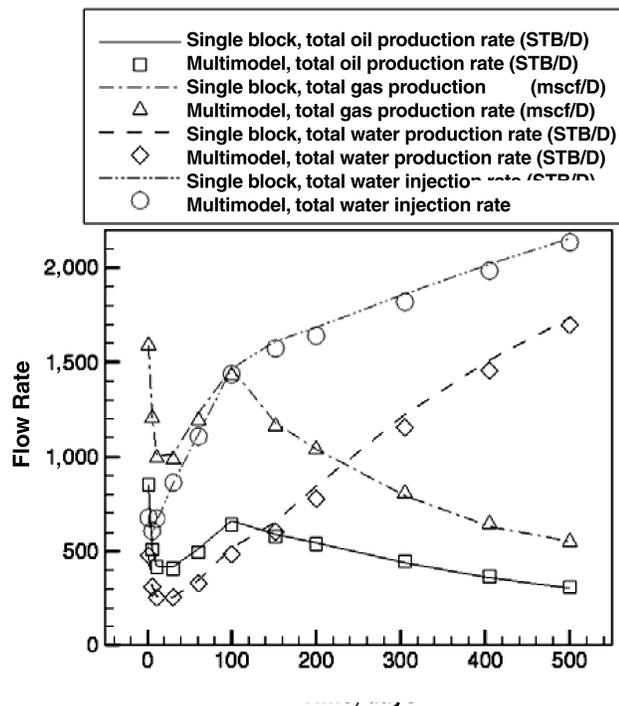


Fig. 18—Comparison of total well flow rates of the reservoir in Example 3 given by multimodel and single-block black-oil model.

TABLE 4—COMPARISON OF CPU TIME BETWEEN MULTIMODEL AND SINGLE-BLOCK BLACK-OIL MODEL FOR SIMULATION TIME OF 500 DAYS IN EXAMPLE 3

	Multimodel	Single-Block Black-Oil
Total CPU time (seconds)	4,053	28,639

K = absolute permeability, L^2 , md
 M_h = discrete space of mortar scalar functions on all interfaces
 n_{bl} = number of subdomains
 n_c = number of components
 N_m = concentration of phase m , m/L^3 , lb/ft³
 N_M = concentration of component M , or stock tank volume of component M per unit pore volume, m/L^3 , stb/bbl
 \bar{N} = interface (mortar space) component concentration, STB/bbl
 P_{cgo} = capillary pressure between gas and oil phases, m/Lt^2 , psi
 P_{cow} = capillary pressure between oil and water phases, m/Lt^2 , psi
 P_m = pressure of phase m , m/Lt^2 , psi
 \bar{P} = interface (mortar space) pressure, m/Lt^2 , psi
 q_M = mass rate of component M as source term, L^3/t , ft³/sec
 R_o = gas/oil ratio, mscf/scf
 R_{so} = solution gas/oil ratio, mscf/scf
 S_m = saturation of phase
 t = time, t, sec
 U_m = flux of phase m , m/t, lb/sec
 U_M = flux of component M , L/t, ft/sec
 α = angle, degree
 Γ_{kl} = the interface between the k th and the l th blocks
 μ = discrete mortar space test function
 μ_m = viscosity of phase m , m/Lt , cp
 ν = outer unit normal vector
 ν_l = outer unit normal vector on face of l th block
 ρ_m = density of phase m , m/L^3 , lb/ft³
 ρ_{GS} = gas phase density at standard condition, m/L^3 , lb/ft³
 ρ_{OS} = oil phase density at standard condition, m/L^3 , lb/ft³
 Φ = porosity
 ψ = primary unknowns on the interface
 Ω = the reservoir domain
 Ω_k = the k th block or subdomain
 ∂D = Dirichlet boundary of subdomain
 $\partial \Omega$ = boundary of Ω
 $[\cdot]_{kl}$ = jump (difference in values on two sides of an interface between the k th block and the l th block)

Subscripts

g = gas phase
 G = gas component
 h = approximated quantity
 m = phase subscript (w , o , or g)
 M = component subscript (W , O , or G)
 o = oil phase
 O = oil component
 S = standard condition
 w = water phase
 W = water component

Superscripts

B = properties of black-oil model
 H = properties of oil/water model

Acknowledgments

We would like to acknowledge several contributors to the multi-block IPARS project, in particular Ivan Yotov, Manish Parashar, and John Wheeler. The U.S. DOE, NSF (KDI grand DMS 9873326), NGOTP, and the Industrial Affiliates of the Center of Subsurface Modeling supported this work.

References

1. Wheeler, M.F. *et al.*: "A Parallel Multiblock/Multidomain Approach for Reservoir Simulation," paper SPE 51884 presented at the 1999 SPE Symposium on Reservoir Simulation, Houston, 14–17 February.
2. Glowinski, R. and Wheeler, M.F.: "Domain Decomposition and Mixed Finite Element Methods for Elliptic Problems," *Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia (1988) 144.
3. Arbogast, T., Wheeler, M.F., and Yotov, I.: "Logically Rectangular Mixed Methods for Flow in Irregular, Heterogeneous Domains," *Computational Methods in Water Resources XI*, SIAM, A.A. Aldama *et al.* (eds.), Computational Mech. Publ., Southampton (1996) 621.
4. Arbogast, T., Wheeler, M.F., and Yotov, I.: "Mixed Finite Elements for Elliptic Problems with Tensor Coefficients as Cell-Centered Finite Differences," *SIAM J. Numer. Anal.* (1997) **34**, 828.
5. Yotov, I.: "Mortar Mixed Finite Element Methods on Irregular Multiblock Domains," *Iterative Methods on Scientific Computation*, IMACS series *Comp. Appl. Math.*, J. Wang *et al.* (eds.), IMACS (1998) 239.
6. Yotov, I.: "Mixed Finite Element Methods for Flow in Porous Media," PhD dissertation, Rice U., Houston, Texas (1996).
7. Wang, P. *et al.*: "A New Generation EOS Compositional Reservoir Simulator: Part I—Formulation and Discretization," paper SPE 37979 presented at the 1997 SPE Reservoir Simulation Symposium, Dallas, 8–11 June.
8. Wheeler, M.F., Wheeler, J.A., and Peszynska, M.: *A Distributed Computing Portal for Coupling Multi-Physics and Multiple Domains in Porous Media*, Thirteenth International Conference on Computational Methods in Water Resources, Calgary, Alberta, Balkema (2000) 167.
9. Peszynska, M., Lu, Q., and Wheeler, M.F.: "Coupling Different Numerical Algorithms for Two-Phase Fluid Flow," *MAFELAP 1999* (Mathematics of Finite Elements and Applications), Elsevier (2000) 205.
10. Peszynska, M., Lu, Q., and Wheeler, M.F.: "Multiphysics Coupling of Codes," *Proc.*, Thirteenth International Conference on Computational Methods in Water Resources, Calgary, Alberta, Balkema (2000) 175.
11. Lu, Q.: "A Parallel Multi-Block/Multi-Physics Approach for Multiphase Flow in Porous Media," PhD dissertation, U. of Texas at Austin (2000).
12. Chapman, R.E.: *Petroleum Geology*, Elsevier Scientific Publishing Company, New York City (1983).
13. Peaceman, D.W.: *Fundamentals of Numerical Reservoir Simulation*, first edition, Elsevier Scientific Publishing Company, Amsterdam-Oxford-New York (1977).
14. Lake, L. W.: *Enhanced Oil Recovery*, Prentice Hall, Englewood Cliffs, New Jersey (1989).
15. Dake, L.P.: *Fundamentals of Reservoir Engineering*, Elsevier Scientific Publishing Company, Oxford-New York (1978).
16. Aziz, K. and Settari, A.: *Petroleum Reservoir Simulation*, Applied Science Publishers Ltd., London (1977).
17. Russell, T.F. and Wheeler, M.F.: "Finite Element and Finite Difference Methods for Continuous Flows in Porous Media," *The Mathematics of Reservoir Simulation*, R.E. Ewing (ed.), SIAM, Philadelphia (1983) 35.
18. Kelly, C.T.: *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia (1995).
19. Edwards, H.C.: "A Parallel Multilevel-Preconditioned GMRES Solver for Multiphase Flow Models in the Implicit Parallel Accurate Reservoir Simulator," *TICAM report*, Texas Inst. for Comp. and Appl. Math., Austin, Texas (February 1998).
20. Honarpour, H., Kedritz, L., and Harvey, A.H.: *Relative Permeability of Petroleum Reservoir*, CRC Press Inc., Boca Raton, Florida (1986).
21. Parashar, M. *et al.*: "A New Generation EOS Compositional Reservoir Simulator: Part II—Framework and Multiprocessing," paper SPE 37977 presented at the 1997 SPE Reservoir Simulation Symposium, Dallas, 8–11 June.
22. Snir, M.: *MPI: the Complete Reference*, The MIT Press, Cambridge, Massachusetts (1996).
23. Tecplot, User's Manual, version 7.5, Amtec Engineering Inc., Bellevue, Washington (1998).

SI Metric Conversion Factors

bbl × 1.589873	E-01 = m ³
cp × 1.0*	E-03 = Pa·s
°F (°F-32)/1.8	= °C
ft × 3.048*	E-01 = m
ft ² × 9.290304*	E-01 = m ²
ft ³ × 2.831685	E-02 = m ³
lbm × 4.435924	E-01 = kg
md × 9.869233	E-04 = μm ²
psi × 6.894757	E+00 = kPa

*Conversion factor is exact.

Qin Lu is a software developer at Landmark Graphics Corp. in Houston, where he currently works on reservoir production software. e-mail: qlu@lgc.com. Lu holds BS and MS degrees in modern mechanics from the U. of Science and Technology of China, and a PhD degree in petroleum engineering from the U. of Texas at Austin. **Malgorzata Peszynska** is a research associate and the Associate Director of the Center for Subsurface Modeling at the U. of Texas' Inst. for Computational and Applied Mathematics. e-mail: mpez@ticam.utexas.edu. She works on mathematical, numerical, and parallel implementation issues related to flow and transport in porous media. Peszynska holds an MS degree in applied mathematics from the Warsaw U. of Technology, Poland, and a PhD degree in mathematics from the U. of Augsburg, Germany. **Mary F. Wheeler** is a professor of aerospace engineering and engineering mechanics, as well as a professor of petroleum and geosystems engineering, at the U. of Texas, as well as the Director of the Center for Subsurface Modeling at that institution. e-mail: mfw@ticam.utexas.edu. Dr. Wheeler's research interests include numerical solution of partial differential systems with application to the modeling of subsurface and surface flows and parallel computation. Wheeler holds a BA degree and an MA degree, both in mathematics, from the U. of Texas at Austin and a PhD degree in mathematics from Rice U.