Iterative Solution Methods for Modeling Multiphase Flow in Porous Media Fully Implicitly

Article *in* SIAM Journal on Scientific Computing · November 2003

DOI: 10.1137/S106482750240443X

4 authors, including:

Yuri Vassilevski
Russian Academy of Sciences
192 PUBLICATIONS **1,760** CITATIONS
SEE PROFILE

Mary F Wheeler
University of Texas at Austin
570 PUBLICATIONS **16,570** CITATIONS
SEE PROFILE

SIAM J. SCI. COMPUT.
Vol. 25, No. 3, pp. 905–926

ITERATIVE SOLUTION METHODS FOR MODELING MULTIPHASE FLOW IN POROUS MEDIA FULLY IMPLICITLY*

S. LACROIX†, YU. VASSILEVSKI‡, J. WHEELER§, AND M. WHEELER§

Abstract. We discuss several fully implicit techniques for solving the nonlinear algebraic system arising in an expanded mixed finite element or cell-centered finite difference discretization of two- and three-phase porous media flow. Every outer nonlinear Newton iteration requires solution of a nonsymmetric Jacobian linear system. Two major types of preconditioners, supercoarsening multigrid (SCMG) and two-stage, are developed for the GMRES iteration applied to the solution of the Jacobian system. The SCMG reduces the three-dimensional system to two dimensions using a vertical aggregation followed by a two-dimensional multigrid. The two-stage preconditioners are based on decoupling the system into a pressure and concentration equations. Several pressure preconditioners of different types are described. Extensive numerical results are presented using the integrated parallel reservoir simulator (IPARS) and indicate that these methods have low arithmetical complexity per iteration and good convergence rates.

Key words. multigrid and multilevel methods, decoupling techniques, reservoir simulation

AMS subject classifications. 65F10, 65N55, 65N06, 65N22

DOI. 10.1137/S106482750240443X

1. Introduction.

In this paper we address the iterative solution of systems arising in a fully implicit cell-centered finite difference or a lowest order Raviart–Thomas expanded mixed finite element discretization of multiphase flow in porous media. Fully implicit time-stepping schemes are the most robust, but expensive, in subsurface flow simulation since they involve large nonlinear systems to be solved at each time step. The algorithm presented here is two-level with the outer level an inexact Newton method and an inner iterative solution method for the corresponding Newton Jacobian. For multiphase flow, the latter matrix is large, sparse, nonsymmetric, and ill-conditioned. Thus, an iterative solver is a very important constituent of the simulator and is the subject of this work; namely, we address development of several preconditioners for the GMRES iteration applied to the solution of the Jacobian system.

Two major types of preconditioners are considered: supercoarsening multigrid (SCMG) and two-stage. The SCMG preconditioner reduces the three-dimensional system to two dimensions by applying a vertical aggregation followed by a two-dimensional multigrid. This approach is defined on rectangular grids and targets domains with larger horizontal than vertical dimensions. A correction based on a semicoarsened three-dimensional grid can be used to handle heterogeneities. This approach can also be used as a subdomain solver for multiblock applications.

A second family of preconditioners are called two-stage. They are based on decoupling the system into a pressure and concentrations equations and do not depend

*Received by the editors March 24, 2002; accepted for publication (in revised form) April 7, 2003; published electronically November 21, 2003.

http://www.siam.org/journals/sisc/25-3/40443.html

†IFP, 1 et 4 av. de Bois-Preau, BP 311, 92852 Rueil-Malmaison Cedex, France (sebastien.lacroix@ifp.fr).

‡INM, 8 ul.Gubkina, Moscow 117333, Russia. Current address: TICAM, The University of Texas at Austin, 211 E. 24th Street, Austin, TX 78712 (vasilevs@ticam.utexas.edu). The work of this author was supported by DOE grant DE-FG03-99ER25371.

§TICAM, The University of Texas at Austin, 211 E. 24th Street, Austin, TX 78712 (jwheeler@ticam.utexas.edu).

on the geometry of the grid. The two-stage preconditioner consists of a robust pressure preconditioner and a relaxed preconditioner for concentrations. Several pressure preconditioners of various types are described.

The above iteration solution methods have been implemented in the IPARS (integrated parallel reservoir simulator) framework [21, 33, 39, 40]. The latter has been developed at the Center for Subsurface Modeling at The University of Texas at Austin. This software has been extensively tested and has been shown to be comparable to existing industrial simulators. IPARS currently contains the following models: sequential and implicit (oil-water) two-phase; implicit black-oil (three-phase, three-component); two-phase (air-water) implicit; explicit and implicit single-phase; equation of state compositional; black-oil coupled to geomechanics; and reactive transport coupled to multiphase flow models. An attractive feature of IPARS is that it allows for the coupling of different models in different subdomains and supports message-passing parallel computations on structured multi-block meshes in two and three spatial dimensions.

In this paper we report on our experience with the SCMG and two-stage preconditioners in terms of comparative characteristics and numerical experiments. In particular, we demonstrate that these methods share a low arithmetical complexity per iteration (optimal or nearly optimal order with respect to the number of unknowns) and good convergence rates. Although the framework has multimodel and multi-block capabilities as well as flexibility with respect to the domain shape, we restrict our attention in this paper to single block and single model examples in a rectangular domain.

The contents of this paper are as follows. In sections 2 and 3, we discuss a general multiphase flow formulation, assuming isothermal and local equilibrium conditions, and its linearization. In section 4 we present the SCMG method applied to the GMRES solution of the Jacobian system. In section 5 we consider two-stage preconditioners for the GMRES iterations applied to the modified Jacobian system. In particular, we formulate physical assumptions that motivate the use of two-stage preconditioners, compare several types of preconditioners, and address the preconditioning of a decoupled pressure equation. In section 6 we discuss the parallel implementation of the solvers within IPARS. Conclusions are given in section 7.

2. Flow formulation. In this section we begin by describing a general multiphase formulation, assuming isothermal and local equilibrium conditions [7, 19], followed by examples.

Consider a porous medium of porosity ϕ and with a general anisotropic nonhomogeneous (intrinsic) permeability tensor $\boldsymbol{K}$. Let D denote depth and G the gravity constant. Multiple flowing phases are denoted by the subscript m, each with associated pressure P_m, density ρ_m, saturation S_m, relative permeability k_m, and viscosity μ_m. Here $\sum_m S_m = 1$. Dissolved and flowing multiple components are denoted with subscript M. Since a component can exist in more than one phase, we define n_{mM} as the mass fraction of component M in phase m and the total concentration of a component as $N_M = \sum_m \rho_m S_m n_{mM}$. Also, since one phase can consist of more than one component we have $\sum_M n_{mM} = 1$. The conservation of mass for each component is given by

$$\frac{\partial(\phi N_M)}{\partial t} + \nabla \cdot \boldsymbol{V}_M = q_M + \sum_m \phi S_m R_{mM}. \tag{1}$$

The source term q_M represents injection/production wells; the terms R_{mM} denote

chemical reactions. Flux $\boldsymbol{V}_M$ is the overall mass flux of component M,

$$\boldsymbol{V}_M = \sum_m \rho_m \left(n_{mM} \boldsymbol{U}_m - \phi S_m \boldsymbol{D}_{mM} \nabla n_{mM} \right), \tag{2}$$

where $\boldsymbol{U}_m$ is the velocity of phase m and $\boldsymbol{D}_{mM}$ represents a diffusion/dispersion tensor. The following definition of $\boldsymbol{U}_m$ is referred to as Darcy's law:

$$\boldsymbol{U}_m = -\boldsymbol{K} \frac{k_m}{\mu_m} (\nabla P_m - \rho_m G \nabla D). \tag{3}$$

The system is closed by adding capillary pressure relationships $P_{cm_1m_2} = P_{m_1} - P_{m_2}$ and relative permeabilities k_m, both typically fluid and rock-type specific functions of saturation. Generally, $k_m(S_m)$ is a nondecreasing function. Additionally, equations of state (constitutive laws) specify the dependence of density ρ_m and viscosity μ_m on pressure P_m and composition n_{mM}.

For a simple immiscible two-phase (oil-water, air-water, or oil-gas) flow model the reactive terms as well as dispersive terms are zero and "phase" is identified with "component" for both wetting fluid ($w \equiv W$) and nonwetting fluid ($n \equiv N$). Thus, $n_{nN} = 1, n_{wW} = 1, n_{nW} = 0, n_{wN} = 0$, and the mass conservation equations are given by

$$\frac{\partial(\phi N_W)}{\partial t} + \nabla \cdot (\rho_w \boldsymbol{U}_w) = q_W, \tag{4}$$

$$\frac{\partial(\phi N_N)}{\partial t} + \nabla \cdot (\rho_n \boldsymbol{U}_n) = q_N. \tag{5}$$

The three-phase black-oil model accounts for the flow of three distinct phases: water, liquid (mainly hydrocarbon components), and gaseous (vapor) air or hydrocarbon. Correspondingly, a system of three equations similar to (4)–(5) can be defined [19, 22].

3. Linearization of general model equations. A multiphase flow model consists of $n + m$ equations associated with each grid cell [7]. For the sake of brevity of presentation, we neglect the capillary pressure and gravity terms. The first n equations are those for conservation of n species M_i:

$$\Delta_t M_i = Q_i \Delta t, \quad i = 1, \ldots, n. \tag{6}$$

Here, Q_i represents intercell flow and well terms:

$$Q_i = \sum_\kappa T_{i\kappa}(p_\kappa - p) - q_i, \tag{7}$$

$\sum_\kappa$ denotes the summation over all neighbor grid cells κ; p and p_κ stand for a grid cell and a neighbor cell pressure; q_i denotes the production rate of species i; and $T_{i\kappa}$ is a transmissibility for flow of species i between a grid cell and its neighbor κ. The equations (6)–(7) are obtained by applying the expanded mixed finite element (the lowest order Raviart–Thomas) method with a trapezoidal quadrature to each of the mass balance equations [38, 37]. This discretization is equivalent to the cell-centered finite difference discretization.

In the case of fully implicit schemes, both $\Delta_t M_i = M_i^{k+1} - M_i^k$ and $Q_i = Q_i^{k+1}$ are unknown and computed by the Newton method. Let $M_i^{k,l+1}$, $Q_i^{k,l+1}$ be the new iterates approximating M_i^{k+1}, Q_i^{k+1}, respectively. Then (6) may be rewritten as

$$M_i^{k,l+1} - M_i^{k,l} + M_i^{k,l} - M_i^k - Q_i^{k,l} \Delta t = (Q_i^{k,l+1} - Q_i^{k,l}) \Delta t. \tag{8}$$

Since $M_i^{k+1} - M_i^k = Q_i^{k+1}\Delta t$, the residual of Newton iteration is

$$r_i = M_i^{k,l} - M_i^k - Q_i^{k,l}\Delta t,$$

and (8) may be written in the form of increments,

$$\delta M_i + r_i = \delta Q_i \Delta t, \quad i = 1, \ldots, n. \tag{9}$$

Given a set of n species, there always exists a set of $n + m$ variables $\{Y_j\}$, $j = 1, \ldots, n + m$, such that each M_i is a unique function of $\{Y_j\}$. The first n variables from $\{Y_j\}$ are called primary, and the remaining variables are referred to as secondary. In order to close the system (9), we need additional m constraint equations. They may express phase equilibrium, saturation constraint, and other model constraints. A general form of the additional equations is

$$\delta L_i + r_i = 0, \quad i = n + 1, \ldots, n + m. \tag{10}$$

Linearization of (9)–(10) yields linear equations

$$\sum_{j=1}^{n+m} g_{ij}^M \delta Y_j + r_i = \sum_{j=1}^{n+m} g_{ij}^Q \delta Y_j, \quad i = 1, \ldots, n, \tag{11}$$

$$\sum_{j=1}^{n+m} g_{ij}^L \delta Y_j + r_i = 0, \quad i = n + 1, \ldots, n + m. \tag{12}$$

where g_{ij}^M, g_{ij}^Q, and g_{ij}^L are the entries of the accumulation, transport and well, and constraint Jacobian terms, respectively.

The system (11), (12) may be written in a block matrix form, based on a primary and a secondary variable decomposition, as

$$\begin{pmatrix} B & C \\ D & E \end{pmatrix} \begin{pmatrix} \delta Y_I \\ \delta Y_{II} \end{pmatrix} = \begin{pmatrix} \delta Z_I \\ \delta Z_{II} \end{pmatrix}. \tag{13}$$

The dependence of the secondary variables is eliminated by the reduction to the Schur complement counterpart of the system (13),

$$A := B - CE^{-1}D, Z := Z_I - CE^{-1}Z_{II}, Y := \delta Y_I,$$

$$AY = Z. \tag{14}$$

The system (14) is obtained by the reduction of linearized equations to the primary variables. These equations are the linearization of the residual formulation for the system of conservation equations. Since Y stands for the vector of primary variables, (14) may not be reduced to a smaller system. It is to be solved by an iterative technique. Although (14) is a Schur complement reduction of the Jacobian system (13), for the sake of brevity we shall refer to it as the Jacobian system. Since the iterative technique used in our numerical experiments is GMRES [16, 25], iterations are controlled by the Euclidean norm of the residual. In the case of different scales of (11), the Euclidean norm may be an irrelevant measure. In order to guarantee the reduction of the Euclidean residual norm for each component equation in (11), we take the initial residual norm to be equal to the minimal component contribution to the global Euclidean norm. We notice that there exist, other than the Euclidean norm, physically motivated and more relaxed stopping criteria for the iterations which reduce the total iteration count.

4. SCMG preconditioner for the coupled system. To define any multigrid method, several components are required. Using superscripts to indicate level number, where 1 denotes the finest level so that $A^1 = A$ and the finest grid (set of degrees of freedom Ω^1) coincides with that of A, the components that a conventional multigrid needs are as follows:

1. "Grids" $\Omega^M \subset \cdots \subset \Omega^2 \subset \Omega^1$.
2. Grid operators $A^M, \ldots, A^2, A^1$.
3. Grid transfer operators:
   "Interpolation," the coarse-to-fine operator, $I_{k+1}^k, k = 1, 2, \ldots, M-1$.
   "Restriction," the fine-to-coarse operator, $I_k^{k+1}, k = 1, 2, \ldots, M-1$.
4. Relaxation scheme for each level.

The recursively defined $(1,1)$ V-cycle uses the above components as follows.

ALGORITHM $MV^k(u^k, f^k)$.

(15) If $k = M$, set $u^M = (A^M)^{-1} f^M$.
Otherwise:
Relax on $A^k u^k = f^k$.
Perform coarse grid correction:
Set $u^{k+1} = 0, f^{k+1} = I_k^{k+1}(f^k - A^k u^k)$.
"Solve" on level $k+1$ with $MV^{k+1}(u^{k+1}, f^{k+1})$.
Correct the solution by $u^k := u^k + I_{k+1}^k u^{k+1}$.
Relax on $A^k u^k = f^k$.

The constituents of the SCMG are as follows. The fine three-dimensional grid Ω^1 is followed by the two-dimensional grid Ω^2 ("two by two" coarsening of an areal cross section of Ω^1). The remaining grids Ω^i, $i = 3, \ldots, M$, are obtained by "two by two" coarsening of the parent grid Ω^{i-1} (see Figure 1). The restriction operator I_1^2 aggregates (sums up) the values (and equations) in vertical cell columns with cell column indexes $2j-1, 2j$, $j = 1, \ldots, N_y/2$, $2k-1, 2k$, $k = 1, \ldots, N_z/2$, where N_y and N_z stand for the number of cells in the y and the z direction, respectively. The interpolation operator $I_2^1 \equiv (I_1^2)^T$ broadcasts an Ω^2 cell value to the respective vertical cell columns. On other levels, the interpolation operators I_{k+1}^k, $k = 2, \ldots, M-1$, are defined to be transposed to the restriction operators I_k^{k+1}. The restriction operator I_k^{k+1} sums up the values and equations for cells belonging to a cell in the coarser mesh Ω^{k+1}. The grid operator A^k, $k = 2, \ldots, M$, is defined as the Galerkin projection of A^{k-1} by $I_{k-1}^k A^{k-1} I_k^{k-1}$. The stencil of the grid operators A^k, $k = 2, \ldots, M$, is the finite difference two-dimensional stencil provided that A^1 has a finite difference three-dimensional stencil. The relaxation scheme on the finest level is a one-line SOR iteration with vertical blocking of unknowns and clustering of the primary variables within grid cells. The vertical cell columns are ordered in the red-black fashion. Hence, the one-color blocks of A^1 are block diagonal matrices, their blocks being associated with vertical cell columns. In the case of LSOR relaxation on the finest grid ($k = 1$) the subblocks are banded matrices with the bandwidth $2n-1$ and the order $N_x n$, where N_x stands for the number of cells in the vertical grid column. The relaxation scheme on the coarser grids ($k = 2, \ldots, M-1$) is the point SOR iteration. In this case the diagonal subblocks are dense matrices of order n. In both cases, the systems with one-color blocks are easy to solve and the (line) SOR iteration is very efficient. For the SOR parameter, we use a heuristic formula. For the finest level relaxation,

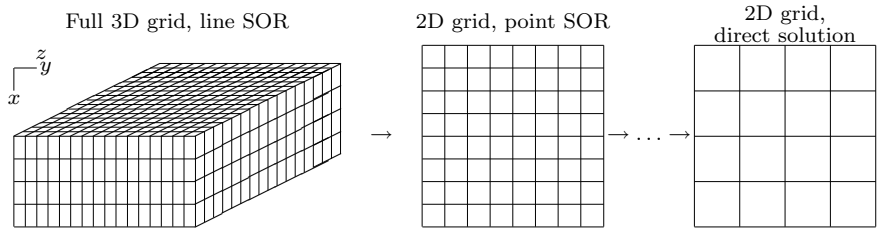

FIG. 1. *SCMG preconditioner.*

we choose an SOR parameter equal to 1 for the sake of the strongest damping of the high frequency components of the error. The most expensive component of SCMG is the relaxation scheme on the finest level and we omit the postsmoothing step. In contrast to the finest level, we can afford a relatively large (5–10) number of point SOR pre- and postsmoothing sweeps within the two-dimensional cycle due to the severe reduction in the number of unknowns.

We now briefly discuss our motivation for SCMG. In standard multigrid schemes a sequence of gradually coarsened grids is used in order to define the grid operators A^k and the grid transfer operators I_{k+1}^k, I_k^{k+1}. The grid transfer operators are chosen on the basis of the mesh hierarchy and the grid operators A^k (in the case of discontinuous coefficients) [9]. In addition, the operators I_{k+1}^k, I_k^{k+1} may be affected by transport phenomena, in particular in the vicinity of wells. The grid operators A^k are chosen as Galerkin coarsening $I_{k-1}^k A^{k-1} I_k^{k-1}$. If the grid transfer operators are not chosen properly, the multigrid convergence deteriorates significantly because of the multiplicative accumulation of the error in (15). Therefore, the use of simple, problem independent, grid transfer operators for general problems is confined to a two-level setting. The relaxation scheme is usually chosen to be either a Jacobi or a Gauss–Seidel smoother in order to damp the high frequency components of the error. In the multicomponent case, the linear system comprises equations with different scales associated with different phenomena; different blocks of the system matrix may have different mathematical properties [14]. Therefore, clustering equations within any grid cell is an essential requirement for the relaxation scheme. Additional clustering along vertical grid columns does not increase the arithmetical complexity of the relaxation and damps the high frequency components and the vertical constituent of the low frequency components. The line SOR iteration with vertical blocking of unknowns makes the relaxation scheme more flexible. However, the LSOR sweep does not efficiently damp all the low frequency components. In order to damp the dominant low frequency components, an LSOR sweep with one-dimensional correction, LSORC, was suggested for two-dimensional problems [34]. The constrained residual technique [32] is a two-dimensional generalization of LSORC. It requires entering a subspace generated by a certain two-dimensional aggregation and iterating in the subspace of residuals. The general theory of subspace iteration was developed in [15]. Maintaining the subspace is provided by a constrained correction of a conventional preconditioner (e.g., one LSOR sweep for A^1 represented by the operator $\tilde{A}^1$) by means of solving two-dimensional aggregated problems with the matrix A^2. The matrix form of the constrained residual preconditioner

$$\tilde{A}^1 + I_2^1 (A^2)^{-1} I_1^2 (I - A^1 \tilde{A}^1)$$

guarantees that all GMRES residuals belong to the subspace $\{I_1^2 v = 0\}$ provided the

initial guess is chosen to be $I_2^1(A^2)^{-1}I_1^2Z$. Therefore, the eigenvectors of $A^1\tilde{A}^1$ with zero vertical mean values will not affect the convergence of GMRES iterations with the constrained residual preconditioner. Numerical evidence [13, 41] has shown that both entering the subspace and computing the exact solution of the aggregated problems are not necessary ingredients of the technique, though a more accurate aggregated solution is preferable. Of additional consideration is the general "pancake" reservoir topology in applications. It results in a larger number of grid cells in horizontal directions compared to the vertical direction. Intuition and analysis [35] suggest the aggregation in the vertical direction in this case.

The approximate solution of the aggregated two-dimensional problem is obtained in SCMG by the V-cycle (15) with strong pre- and postsmoothings and the simplest intergrid transfer operators. This results in the accurate solution of the two-dimensional aggregated system in the case of smooth coefficients. In the case of rough coefficients, the two-dimensional solution is not as accurate. However, it is capable of damping the low frequency aggregated components, which is the purpose of the two-dimensional correction. We notice that the constrained residual framework does not assume expensive postsmoothing LSOR sweeps on the finest grid Ω^1. That is why we omit the postsmoothing LSOR sweep. On other grids, we apply both pre- and postsmoothing SOR sweeps in order to make the two-dimensional correction as accurate as possible.

It is pertinent to note that in spite of the structural similarity between SCMG and the conventional geometric three-dimensional multigrid (MG), they rely on different iterative mechanisms. The three-dimensional MG exploits the consequent residual corrections on coarser levels, whereas SCMG is the constrained residual technique, where the strengthened two-dimensional MG is used for the solution of the aggregated problem. We believe that the latter framework allows us to use simple problem independent grid transfer operators for applications with strongly variable coefficients.

The low frequency correction of the three-dimensional operator by the two-dimensional solution may cause a loss of efficiency: the convergence of the SCMG is sensitive to the grid size in horizontal directions (horizontal constituents of all low frequency components are not damped) and heterogeneity of the permeability coefficients (recursive usage of inappropriate interpolation operators). In order to minimize the above effects, we augment the SCMG with an additional correction. We want to minimize the use of simple interpolation operators and to damp the horizontal constituents of the low frequency components. On the other hand, the correction has to be relatively cheap compared to the SCMG cycle. We found a compromise in a "four by four" aggregation in the horizontal directions and the approximate solution of the resulted system by eight LSOR iterations with an appropriate parameter and vertical clustering of the unknowns. The cost of the additional correction does not exceed one-half of the finest level relaxation of SCMG. The most important feature of the correction is that the coarsening direction is orthogonal to the relaxation direction. Benefits of this approach have been discussed in [8]. In the implementation of the additional correction, the restriction operator I_1^c sums up the values and equations for cells in the same horizontal layer belonging to a coarse grid column associated with the coarse grid Ω^3 (see Figure 2). The coarse grid operator is the Galerkin projection of A^1, $I_1^cA^1(I_1^c)^T$.

In Tables 1 and 2 we show the performance of the SCMG with and without the additional correction. Two models are considered: the oil-water [22, 42] ($n = 2$), where the primary variables are oil pressure and oil concentration, and the black-oil [29, 42] ($n = 3$), with oil pressure and oil and gas concentrations as primary variables. An

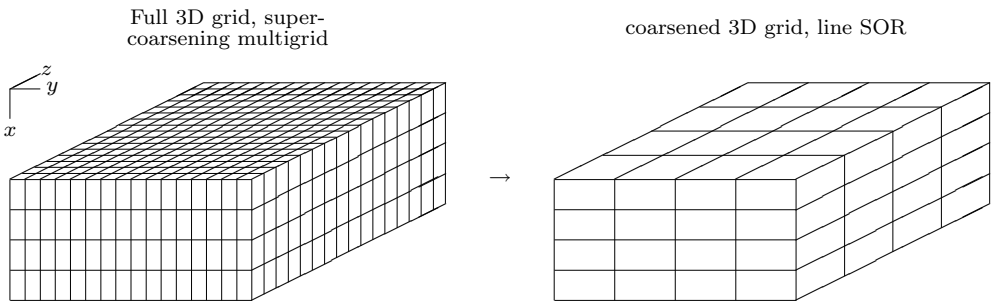

FIG. 2. *Additional correction for the SCMG preconditioner.*

18-day simulation takes 10 time steps. In the tables we examine several test cases: two wells with specified "bottom hole pressure" in a homogeneous permeability field with a narrow horizontal layer with 4-fold jump in x-permeability (Cases 1–4), and 26 wells with specified "bottom hole pressure" in a heterogeneous permeability field (SPE9 data [12]) with 10^4-fold variations in permeability (Case 5). The black-oil simulation uses zero initial gas concentration. The $N_x \times N_y \times N_z$ grids are $10 \times 80 \times 80$ (Case 1), $10 \times 20 \times 20$ (Case 2), $20 \times 40 \times 40$ (Case 3), $40 \times 80 \times 80$ (Case 4), and $15 \times 24 \times 25$ (Case 5). The stopping criterion for the GMRES iterations is the relative reduction of the Euclidean residual norm for each component equation in (11) by a factor of 100. In all experiments throughout this paper the mass balance was maintained to at least six digits. In the tables we present the accumulated number of GMRES iterations (#GMRES), the CPU time of the linear solver, the average number of GMRES iterations per Newton step (#GMRES / #Newt), and the CPU time per iteration per unknown (CPU/(#GMRES #dof)) scaled by 10^{-6}. The measurements were performed on Pentium II (400 MHz).

The data show that the CPU time per iteration is proportional to the number of unknowns (for a given model); the convergence rate of the GMRES iterations does not depend on the number of cells in the vertical direction (Cases 1 and 4). However, for the SCMG preconditioner, the number of iterations per Newton step depends on the square root of the mesh size in the areal direction, and it is sensitive to the heterogeneity of the permeability field. For SCMG with the additional correction, the convergence rate is almost insensitive to mesh size and is less sensitive to the heterogeneity of the permeability field. This results in the reduction of CPU time on large grids.

5. Two-stage preconditioning the coupled system. An alternative approach to the iterative solution of (14) exploits physical rather than geometrical properties of matrix A. Since the blocks of the system matrix have different physics, the sensible approach to the construction of a preconditioner is to precondition different blocks separately, taking advantage of their properties. This yields two potential advantages. First, instead of the construction of a preconditioner for the whole matrix, one can focus on preconditioning the stiffest block of the matrix and use a relaxed preconditioner for the remaining part. A second advantage of this approach is its independence of the underlying mesh geometry since the two preconditioners may depend only on the algebraic matrix structure. The blocks of the entire matrix are coupled through nontrivial off-diagonal blocks, and thus the issue of decoupling the blocks must be considered.

TABLE 1
Oil-water simulation with the SCMG preconditioner.

	Preconditioner	Case 1	Case 2	Case 3	Case 4	Case 5
#GMRES		415	104	169	422	378
CPU time	SCMG	426	5.5	74	1570	53
#GMRES / #Newt		12.6	5.2	7.7	12.8	18.9
CPU/(#GMRES #dof)		7.7	6.6	6.8	7.2	7.7
#GMRES		210	102	131	210	225
CPU time	SCMG	275	7.5	79	1037	41
#GMRES / #Newt	with	6.3	5.1	5.7	6.3	11.2
CPU/(#GMRES #dof)	corr.	9.9	9.2	9.4	9.6	10

TABLE 2
Black-oil simulation with the SCMG preconditioner.

	Preconditioner	Case 1	Case 2	Case 3
#GMRES		311	111	162
CPU time	SCMG	621	12.3	139
#GMRES / #Newt		9.4	5.0	6.5
CPU/(#GMRES #dof)		10	9.2	9
#GMRES		183	102	126
CPU time	SCMG	528	16.7	164
#GMRES / #Newt	with	5.5	4.6	5
CPU/(#GMRES #dof)	corr.	15	13.5	13.5

5.1. Decoupling pressure equation: Basic assumptions. Properties of the Jacobian system depend on the nature of (6) and (10). Since we do not specify particular characteristics of (6) and (10), we make *assumptions* on the reduced Jacobian matrix A, in order to motivate further constructions.

Although numerous sets of primary variables are available [23], we restrict our attention to a very particular set.

ASSUMPTION 1. *We assume that Y_1 is the grid cell pressure and $\{Y_j\}$, $j = 2, \ldots, n$, are the grid cell concentrations (or saturations).*

We remark that no special phase pressure has been chosen. However, the optimal choice of the component turns out to be very important in computational practice.

Additional constraint equations (12) used to close (11) may be chosen to possess local properties. Thus, we may assume that the constraint equations (12) state relationships among variables in each grid cell independently of other grid cells. On the other hand, the physical properties of all three terms of the Jacobian, accumulation $\Delta_t M_i$, transport $\sum_\kappa T_{i\kappa}(p_\kappa - p)$, and well terms q_i indicate that *the interaction among variables is chiefly local*. In algebraic terms, this allows us to make the following assumption.

ASSUMPTION 2. *Consider the block representation of matrix A associated with grid cell blocks. The off-diagonal block entries responsible for intercell interaction among different variables are small compared to the respective entries of the diagonal block (interactions within a cell).*

According to Assumption 1, our formulation is presented in terms of pressure and concentrations. At least for the black-oil isothermal models, the studies [2, 5, 22] demonstrate that the pressure equation is essentially parabolic or elliptic and the saturation equations are hyperbolic or transport dominated parabolic. These features are expected to be inherited by compositional models as well [36]. A well-known consequence is that the pressure equation must be treated implicitly and the saturation equations *may* be treated explicitly (Impes models) [23]. Applicability of

the Impes models is a starting point for our considerations. We note that implicit pressure and explicit saturation advancing in time approximates the original parabolic equations. It implies that the solutions due to Impes and fully implicit time stepping are close to each other. Therefore, the respective time step nonlinear operators are close in a sense, and their linearizations (Jacobian) are expected to possess similar properties as well. Thus, given a meaningful guess for the pressure variable, we hope that an explicit update of the concentrations yields a meaningful guess for the concentration variables. It means that an explicit concentration calculation based on physically reasonable pressure computation results in a meaningful approximation for the inversion of the fully implicit Jacobian.

ASSUMPTION 3. *Consider a reduced system with the fully implicit Jacobian* (14). *Let the matrix A and the vectors Y, Z be split into pressure and concentration blocks,*

$$A = \begin{pmatrix} A_p & A_{ps} \\ A_{sp} & A_s \end{pmatrix}, \quad Y = \begin{pmatrix} Y_p \\ Y_s \end{pmatrix}, \quad Z = \begin{pmatrix} Z_p \\ Z_s \end{pmatrix},$$

and let a meaningful approximation of $\tilde{Y}_p$ to Y_p and an easy-to-invert approximation of $\tilde{A}_s$ to A_s be known. Then $(\tilde{Y}_p, \tilde{A}_s^{-1}(Z_s - A_{sp}\tilde{Y}_p))^T$ is a meaningful approximation to $(Y_p, Y_s)^T$.

The choice $\tilde{A}_s = A_s$ implies the solution of a concentration system. Lesser stiffness of A_s allows us to replace A_s by a simple approximation (cell block Jacobi). We note, however, that Assumption 3 is not applicable to the solution of (14) directly, since a meaningful guess of $\tilde{Y}_p$ is not known. Computation of such a guess is the main target of decoupling techniques.

5.2. Decoupling pressure equation: Combinative technique versus block Gauss–Seidel update. The pressure equation in (14) can be expressed as

$$A_p Y_p + A_{ps} Y_s = Z_p.$$

Here the pressure variable is coupled to the concentration variables by matrix A_{ps}. This coupling is chiefly local (Assumption 2), which means that the entries of matrix A_{ps} not belonging to the diagonal cell blocks $\{A\}_{ii}$ of A may be neglected in the construction of a preconditioner. Therefore, any transformation of system (14) that makes the diagonal cell blocks $\{A_{ps}\}_{ii}$ of A_{ps} to be zero essentially decouples pressure from concentration and allows us to find $\tilde{Y}_p$. Hereinafter, we denote by $\{A\}_{ii}$ the diagonal blocks of a matrix A, which is decomposed into grid cell blocks. Within this notation we consider a transformation of (14) such that $\{A_{ps}\}_{ii} = 0$.

Actually, the simplest transformation is the block diagonal matrix, where each block is just $\{A\}_{ii}^{-1}$. The drawback of this transformation is that it does not preserve the Euclidean norm of the right-hand side, and hence requires a much more accurate solution to the transformed system (14) than is required by the inexact Newton method. Other transformations that preserve, or nearly preserve, the Euclidean norm are CPR [14, 32], Householder [10], and quasi-Impes [17, 18]. They provide preconditioners that yield the same convergence of an iterative procedure [17, 18]. Here we consider the Householder transformation since it preserves the Euclidean norm exactly, and its implementation is more efficient.

Let G_{ii} be a product of $n-1$ Householder matrices,

$$G_{ii} = P_{1,ii} \cdot P_{2,ii} \dots P_{n-1,ii}. \tag{16}$$

Multiplication of a matrix by $P_{k,ii}$ results in a zero kth row of the upper triangular part of $\{A\}_{ii}$. Hence,

$$G_{ii}^H\{A\}_{ii} = \left\{ \begin{array}{cc} A_p^H & O \\ A_{sp}^H & A_s^H \end{array} \right\}_{ii}, \tag{17}$$

where A_s^H is a lower triangular matrix. This implies not only decoupling pressure from concentrations but also a virtual factorization of the concentration block A_s^H within a cell block. Multiplication of (14) by the block diagonal matrix

$$G = blockdiag\{G_{ii}\}$$

yields the transformed system

$$A^H = GA, \quad A^H Y = GZ. \tag{18}$$

Block representation of A^H and its block Gauss–Seidel preconditioner $\tilde{A}^H$ related to primary variables are

$$A^H = \left(\begin{array}{cc} A_p^H & A_{ps}^H \\ A_{sp}^H & A_s^H \end{array} \right), \quad \tilde{A}^H = \left(\begin{array}{cc} A_p^H & O \\ A_{sp}^H & \tilde{A}_s^H \end{array} \right). \tag{19}$$

Here $\tilde{A}_s^H$ denotes the cell block Jacobi approximation to A_s^H. The solution procedure for a system $\tilde{A}^H x = r$ is as follows. The pressure equation, $A_p^H x_p = r_p$, is solved; the concentration residual is computed and preconditioned, $x_s = (\tilde{A}_s^H)^{-1}(r_s - A_{sp}^H x_p)$. We notice that the last step is computationally inexpensive because of the virtual factorization of matrices $\{A_s^H\}_{ii}$. In addition, according to Assumption 3, the exact solution of the pressure equation may be replaced by an approximate one. In algebraic terms, one could use a preconditioner $\tilde{A}_p^H$ for A_p^H in the pressure equation.

The assumption that pressure "governs" concentrations but is not "governed" by concentrations may be excessive. The preconditioner providing feedback for the pressure-concentration interaction is likely to converge faster. An example of such a preconditioner is the combinative two-stage preconditioner [4, 14]. The action of the two-stage combinative preconditioner for matrix A^H, $Y = (\tilde{A}_2^H)^{-1}Z$, is as follows:

1. Solve the pressure equation $A_p^H Y_p = Z_p$.
2. Compute the total residual

$$\left(\begin{array}{c} R_p \\ R_s \end{array} \right) = \left(\begin{array}{c} Z_p \\ Z_s \end{array} \right) - \left(\begin{array}{c} A_p^H \\ A_{sp}^H \end{array} \right) Y_p.$$

3. Precondition the total residual and update

$$\left(\begin{array}{c} Y_p \\ Y_s \end{array} \right) := (\hat{A}^H)^{-1} \left(\begin{array}{c} R_p \\ R_s \end{array} \right) + \left(\begin{array}{c} Y_p \\ O \end{array} \right).$$

Here, $\hat{A}^H$ stands for a preconditioner to A^H providing a pressure dependence on concentrations. The difference between the combinative $\tilde{A}_2^H$ and block Gauss–Seidel preconditioner $\tilde{A}^H$ is in computing and preconditioning the residual, as well as in the presence of the feedback update of the pressure. By analogy with the block Gauss–Seidel preconditioner, the block $(A_p^H)^{-1}$ may be replaced by any preconditioner. We notice that the preconditioner $\hat{A}^H$ to the whole matrix may be chosen to

TABLE 3
Different types of coupling preconditioners within the combinative technique.

	Oil-water		Black-oil	
method	#GMRES	CPU time	#GMRES	CPU time
$\tilde{A}^H_{2,L}$	74	60	293	319
$\tilde{A}^H_{2,I}$	87	74	381	372
$\tilde{A}^H_{2,J}$	201	91	787	505
$\tilde{A}^H$	203	85	701	425

be rather weak, since its goal is to provide a pressure-concentration feedback. Possible candidates are $ILU(1)$ [32], $ILU(0)$, one LSOR iteration, and the block Jacobi preconditioner [18].

In Table 3 we compare the combinative preconditioner with the block Gauss–Seidel preconditioner (19). The preconditioner $\tilde{A}^H$ uses the cell block Jacobi approximation $\tilde{A}^H_s$ of A^H_s. The global preconditioner $\hat{A}^H$ in the combinative method $\tilde{A}^H_2$ is the cell block Jacobi preconditioner ($\tilde{A}^H_{2,J}$), or one LSOR iteration with blocks associated with vertical grid lines ($\tilde{A}^H_{2,L}$), or an ILU(0) preconditioner for A^H ($\tilde{A}^H_{2,I}$). Two models are considered: the oil-water [22, 42] and the black-oil [29, 42], with the primary variables from the example of section 4. We use the $20 \times 40 \times 40$ mesh and a heterogeneous permeability field with 10-fold variations. The 18-day simulation with two "bottom hole pressure" wells took 10 time steps. The black-oil simulation involves no initial gas concentration. In Table 3 we present the accumulated number of GMRES iterations and the CPU time of the iterations. The stopping criterion for the linear solver is the relative reduction of the Euclidean residual norm for each component equation in (11) by a factor of 100. The pressure block preconditioner is chosen to be the algebraic multigrid (AMG) (see the next section). The data show that the most efficient method is the combinative preconditioner with one LSOR sweep for the coupled system.

5.3. Preconditioning pressure equation. The modified pressure block $\bar{A} \equiv A^H_p$ of the Jacobian matrix is sparse and nonsymmetric. Moreover, it is stiff due to its elliptic nature. In this section, we consider several preconditioners for the modified pressure block that have low arithmetical cost proportional to the order of the pressure block. Before we describe the preconditioners, we discuss the structure of the pressure block. Since the underlying meshes in IPARS are three-dimensional rectangular, a preconditioner to the pressure block may be constructed for the finite difference 7-point stencil. Therefore, if necessary , one could consider the pressure block as a block matrix with a two-dimensional (5-point) areal stencil. Each diagonal block is a tridiagonal matrix associated with couplings within a vertical cell column. If we order the vertical cell columns in the red-black areal fashion, we obtain the following block representation:

$$\bar{A} = \begin{pmatrix} \bar{A}_r & \bar{A}_{rb} \\ \bar{A}_{br} & \bar{A}_b \end{pmatrix}. \tag{20}$$

Here, $\bar{A}_r$ and $\bar{A}_b$ are the block diagonal matrices. Each block, being a tridiagonal matrix, represents the interaction of a column of grid cells with itself (a one-dimensional problem). Therefore, evaluation of $\bar{A}_r^{-1}$, $\bar{A}_b^{-1}$ is computationally cheap.

The line SOR is obtained by applying extrapolation to the block Gauss–Seidel method. This extrapolation takes the form of a weighted average between the previous

iterate x^{k-1} and the computed block Gauss–Seidel iterate $\bar{x}^k$ as follows:

$$x^k = \omega \bar{x}^k + (1-\omega)x^{k-1}, \quad k = 1, \ldots, L_{SOR}. \tag{21}$$

The optimal value of ω is computed according to heuristic formulae. The larger the number of LSOR iterations L_{SOR} is, the better the LSOR operator $\bar{A}_{LSOR}$ approximates $\bar{A}$.

The separable preconditioner is based upon an inversion of the following discrete operator [1, 15, 24, 30]:

$$\bar{A}_{Sp} = A_1 \otimes M_2 \otimes M_3 + M_1 \otimes A_2 \otimes M_3 + M_1 \otimes M_2 \otimes A_3.$$

Here, A_i stands for a symmetric tridiagonal matrix and M_i stands for a diagonal matrix, $i = 1, 2, 3$. The method features insensitivity to an anisotropy of the grid, arbitrary dimensions of matrices A_i, M_i, and a suboptimal arithmetical complexity ($O(N \log^2 N)$). Matrices M_i are taken to be identities, and matrices A_i are chosen to be submatrices of the pressure block of the transport term of the unmodified Jacobian A. The submatrices are associated with given "tracing grid lines" in directions x, y, z.

Finally, we consider two versions of an algebraic multilevel method, the AMG [28] and the multilevel incomplete LU (MLILU) [3]. Their general components and V-cycle structure are the same as in (15). They are different only in the setup phase and grid transfer operators. Operators on coarse grids are defined via Galerkin projection, i.e.,

$$A^{k+1} = I_k^{k+1} A^k I_{k+1}^k.$$

We note that, in the case of the AMG, the interpolation I_{k+1}^k and restriction I_k^{k+1} operators are transposed, $I_k^{k+1} = (I_{k+1}^k)^T$. For the MLILU method, this is true only for symmetric matrices. The choice of components in the algebraic multilevel methods is done in a separate preprocessing step as follows:

Setup phase.

1. Set $k = 1$.
2. Partition Ω^k into disjoint sets C^k and F^k:
   (a) Set $\Omega^{k+1} = C^k$.
   (b) Define "interpolation" I_{k+1}^k and "restriction" I_k^{k+1}.
3. Set $A^{k+1} = I_k^{k+1} A^k I_{k+1}^k$.
4. If Ω^{k+1} is small enough, set $M = k+1$ and stop. Otherwise, set $k = k+1$ and go to step 2.

The goals of the setup phase are to choose the set C^k of coarse grid points, to choose for each fine grid point $i \in F^k \equiv \Omega^k \setminus C^k$ a small set $C_i^k \subset C^k$ of interpolating points, and to choose for each coarse grid point $j \in C^k \equiv \Omega^{k+1}$, a small set $F_j^k \subset F^k$ of restriction points. Interpolation and restriction are then defined in terms of weights by

$$(I_{k+1}^k u^{k+1})_i = \begin{cases} u_i^{k+1} & \text{if } i \in C^k, \\ \sum\limits_{j \in C_i^k} \omega_{ij} u_j^{k+1} & \text{if } i \in F^k, \end{cases} \qquad (I_k^{k+1} u^k)_j = u_j^k + \sum_{i \in F_j^k} \bar{\omega}_{ji} u_i^k.$$

We recall that in case of AMG, $\bar{\omega}_{ji} = \omega_{ij}$.

Thus, the general structure of AMG and MLILU is the same. However, the techniques for generation of the intergrid transfer operators and separation into fine and coarse nodes are totally different. First, it is important to note that AMG is

applied to matrices with positive diagonal entries, whereas MLILU is appropriate for matrices with negative diagonal entries. As a result, the definitions of these sets of influencing points differ as follows:

$$S_i = \left\{ j \neq i \,|\, -a_{ij} \geq \sigma \max_{k \neq i}(-a_{ik}) \text{ (AMG)} \right\},$$

$$S_i = \left\{ j \neq i \,|\, a_{ij} \geq \sigma \max_{k \neq i}(a_{ik}) \text{ (MLILU)} \right\}.$$

The use of these sets is important for the computation of the weights ω_{ij}, $\bar{\omega}_{ji}$.

The weight evaluation is based on different ideas for AMG and MLILU. The AMG algorithm is focused on errors e satisfying $Ae \approx 0$ [28]. The MLILU method is designed to approximate the Schur complement K, associated with a Gauss elimination, by a matrix $\bar{K}$ so that a filtering condition is satisfied $(K - \overline{K})t = 0$ for a given test vector t [3].

The other major difference is also in the separation step. The main tool of the separation is an oriented adjacency matrix graph whose edges are described by the formula

$$e_{ij}^{(0)} = \begin{cases} 1 & \text{if } j \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$

The coarse nodes are defined to be the nodes influencing the maximum number of fine nodes. By the influence of a node i we understand the summation of all the edges coming from other nodes, i.e., $\lambda_i = \sum_{j \in (S_i)^T} e_{ji}$, where $S_i^T = \{k, i \in S_k\}$. In the MLILU method, the sets C^k and F^k are constructed to reduce a fill-in. Namely, for each node i, the MLILU separation step looks forward to finding two weakly connected nodes (parent nodes), which provide a minimum fill-in in the next graph after elimination of the node i. Then, the connections for the parent nodes are reinforced compared to other nodes linked to the node i, i.e.,

$$e_{ij}^{(1)} = \begin{cases} e_{ij}^{(0)} & \text{if } j \text{ is not a parent node,} \\ \beta_{ij}(\nu_i, nb_i) & \text{otherwise,} \end{cases}$$

where ν_{ij} is the number of new edges in the next graph and nb_i is the number of neighbors of the node i. The couplings are updated in MLILU, whereas they remain the same in AMG. In other words, the separation step for AMG exploits all the nodes from the influencing sets while, for MLILU, it selects just subsets of the influencing sets. After this step the separation is as follows:

1. Set $C^k = \{\}$, $F^k = \{\}$, $U^k = \Omega^k$.
   (a) Until U^k is empty select i with maximum λ_i.
      i. If $\lambda_i > 0$, set $C^k = C^k \cup \{i\}$ and $U^k = U^k \setminus \{i\}$.
      ii. For all nodes j depending on i, i.e., $j \in S_i^T$:
      – If (AMG), select all j as fine nodes $F^k = F^k \cup \{j\}$.
      – If (MLILU), select j as a fine node only if two of his parent nodes are coarse nodes or if it is linked to a unique coarse node.
      – Update connections of the nodes k influencing all the new fine nodes and the coarse node regarding:
      – – If (AMG), initial sets S_k^T.
      – – If (MLILU), updated sets $\overline{S_k^T}$ corresponding to the new graph.

iii. If $\lambda_i = 0$:
- If (AMG), always select this node as a coarse node, $C^k = C^k \cup \{i\}$.
- If (MLILU), impose weaker F node condition and check if i could be a fine node, otherwise select i as a coarse node.

2. If (AMG), check all F nodes to ensure there is no connection between them or dependency toward interpolating nodes.
   If (MLILU), improve interpolation and restriction by possible extensions of the sets of parent nodes.

The last step of MLILU does not change the separation into subsets F^k and C^k; it slightly modifies the interpolation and restriction patterns in terms of F_j^k, C_i^k. In contrast to MLILU, the last step of AMG modifies the separation essentially.

It is important to notice the difference in the node labeling for both methods. The MLILU separation seems to be more advanced compared to the AMG counterpart, since it provides a better approximation for a lower complexity. The compensation for better labeling is a higher initialization price (10 to 40 MLILU V-cycles versus 5 to 10 AMG V-cycles). For other details we refer to [6, 3], respectively.

The major computational cost in our nonlinear solution is the initialization of the pressure multilevel preconditioner. In order to balance the initialization and the iterative solution, we take advantage of a two-stage method. We recall that the LSOR iteration (with vertical blocking) is a very good smoother in the vertical direction. Therefore, we can replace the preconditioning of the whole pressure block by preconditioning its Galerkin projection, $\bar{A}_c = I_x^T \bar{A} I_x$, where I_x stands for an interpolation operator in the vertical direction. We define the interpolation operator as a block diagonal matrix, where each block is associated with a vertical grid line. In order to define the matrix blocks, we group the unknowns within a block by triples, except those associated with the top and the bottom of the grid line: $\{1\}, \{2, 3, 4\}, \{5, 6, 7\}, \ldots, \{N_x\}$. The block of I_x contains ones in each column associated with each subset, i.e.,

$$I_x = \begin{pmatrix} 1 & 0 & 0 & \ldots \\ 0 & 1 & 0 & \ldots \\ 0 & 1 & 0 & \ldots \\ 0 & 1 & 0 & \ldots \\ 0 & 0 & 1 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

The restriction operator, I_x^T, is simply the averaging operator. The 7-point stencil of the Galerkin projection $\bar{A}_c$ remains the same, and the order of matrix $\bar{A}_c$ is almost as much as three times smaller than that of matrix $\bar{A}$. If we denote the multilevel preconditioner (AMG or MLILU) for $\bar{A}_c$ by B_c, the two-stage preconditioners are

$$\bar{A}_{AMG,2} = I_x B_{c,AMG} I_x^T + \bar{A}_{LSOR(1)}(I - \bar{A} I_x B_{c,AMG} I_x^T),$$

$$\bar{A}_{MLILU,2} = I_x B_{c,MLILU} I_x^T + \bar{A}_{LSOR(1)}(I - \bar{A} I_x B_{c,MLILU} I_x^T).$$

We notice that the introduction of the *geometric* interpolation operator may result in a deterioration of the convergence if the interpolation operator is not stable in an appropriate norm. This may happen in the case of severe heterogeneity of coefficients.

In Tables 4 and 5 we show the performance of the combinative technique with different pressure block preconditioners. As in section 3, two models are considered: the oil-water [22, 42] and the black-oil [29, 42], with the primary variables from the example of section 4. The black-oil simulation has no initial gas concentration. The

TABLE 4
Oil-water simulation with the decoupling preconditioner.

	Pressure block prec.	Case 1	Case 2	Case 3	Case 4	Case 5
#GMRES	$\bar{A}_{LSOR(5)}$	1296	163	307	1471	740
CPU		1515	11.6	195	6561	117
#GMRES / #Newt		41.8	8.1	14.6	46	37
CPU/(#GMRES #dof)		8.8	8.9	9.9	8.7	8.8
#GMRES	$\bar{A}_{Sp}$	206	86	110	221	-
CPU		375	7.1	75	1411	-
#GMRES / #Newt		6.2	4.3	5.2	6.7	-
CPU/(#GMRES #dof)		13.8	10.3	10.7	12.4	-
#GMRES	$\bar{A}_{AMG}$/ $\bar{A}_{AMG,2}$	139/154	56/56	81/81	204/186	173/435
CPU		401/338	11.9/6.6	166/73	3374/1295	51/77
#GMRES / #Newt		4.2/4.8	2.8/2.8	3.9/3.9	6.4/5.8	8.6/21.7
CPU/(#GMRES #dof)		22/16	26/15	32/14	32/14	16/10
#GMRES	$\bar{A}_{MLILU}$/ $\bar{A}_{MLILU,2}$	136/153	55/56	77/88	-/184	220/448
CPU		1166/627	16.2/8.8	289/128	-/3692	91/92.4
#GMRES / #Newt		4.1/4.8	2.8/2.8	3.6/4.4	-/5.8	11/22.4
CPU/(#GMRES #dof)		67/32	37/20	59/23	-/39	23/11

TABLE 5
Black-oil simulation with the decoupling preconditioner.

	Pressure block prec.	Case 2	Case 3
#GMRES	$\bar{A}_{LSOR(5)}$	110	253
CPU		15.2	286
#GMRES / #Newt		4.4	7.9
CPU/(#GMRES #dof)		11	11
#GMRES	$\bar{A}_{AMG,2}$	149	355
CPU		20.8	392
#GMRES / #Newt		5.7	11.1
CPU/(#GMRES #dof)		11	11

18-day simulation takes 10 time steps. The same five test cases are examined (cf. section 3). The stopping criterion for the GMRES iterations is chosen as the 100-fold relative reduction of the Euclidean residual norm for each component equation in (11); see [10]. In the tables we present the accumulated number of GMRES iterations (#GMRES), the CPU time of the linear solver, the average number of GMRES iterations per Newton step (#GMRES / #Newt), and the CPU time per GMRES iteration per degree of freedom (CPU/(#GMRES #dof)) scaled by 10^{-6}.

Based on the data, we conclude that the CPU time per GMRES iteration is proportional (LSOR), or almost proportional (AMG, MLILU,[1] separable), to the number of unknowns (for a given model); the convergence rate of all the methods except $\bar{A}_{LSOR}$ very slightly depends on the grid (Cases 1–4, oil-water model). In the case of the black-oil model, the convergence is more sensitive to the mesh size, and $\bar{A}_{LSOR}$ appears to be very efficient (at least on moderately sized grids). Two methods, the AMG and the MLILU, are *robust* with respect to heterogeneity of the permeability field, while the convergence of their projected counterparts, $\bar{A}_{AMG,2}$ and $\bar{A}_{MLILU,2}$, as well as the LSOR inner iterations, is sensitive to heterogeneity. Another feature of the heterogeneous case is that the arithmetical complexity of both MLILU and AMG

[1] An increase in the CPU time per iteration per unknown for the MLILU preconditioners is attributable to memory reallocation effects rather than to computational work.

solvers is smaller than in the homogeneous cases. The separable preconditioner is not applicable in the heterogeneous case since there is no combination of one-dimensional operators approximating a three-dimensional heterogeneous operator. Also, it is not relevant to the black-oil model.

6. Parallelization of the algorithms. The underlying meshes in IPARS are three-dimensional rectangular, and the grid cells (and degrees of freedom) are grouped in clusters of vertical columns. The vertical blocking of unknowns and equations is used in the solution techniques discussed in the previous sections. This implies that degrees of freedom within each vertical cell column are associated with one processor. Therefore, grid data are allocated among processors in the areal fashion. The main purpose of decomposition is to equidistribute the arithmetical load over the processors.

The GMRES iterative solution of linear problems requires the evaluation of the matrix-vector product. In IPARS, parallelization of the matrix-vector product is implemented using the "ghost cell" technique. Assume matrix A, consisting of P block rows, is to be multiplied by vector x, partitioned into P blocks, where both A and x are allocated among P processors, so that the ith block row of A and the ith block of x are known to processor i as follows:

$$A = \begin{pmatrix} A_{11} & \dots & A_{1P} \\ A_{21} & \dots & A_{2P} \\ \vdots & \ddots & \vdots \\ A_{P1} & \dots & A_{PP} \end{pmatrix}.$$

Then, in order to obtain the ith block of vector Ax on processor i, the following sum is computed:

$$A_{ii}x_i + \sum_{j \neq i}^{P} A_{ij}x_j.$$

This means that, in addition to having x_i allocated to the ith processor, the parts of x_j, $j \neq i$, contributing to $A_{ij}x_j$ must be available to the ith processor. If the matrix A comes from a finite element or finite difference approximation, the contributing parts of x_j live on neighbor-to-processor i cells. These cells are referred to as "ghost cells" for the processor i since they do not constitute the set of degrees of freedom on the processor i but contribute to the matrix-vector multiplication. Before each evaluation, the contributing part of x_j assigned to the processor i must be updated.

Other parallel ingredients of the parallel GMRES implementation include the scalar product and the preconditioner-vector multiplication. The scalar product is computed using the summation of the local scalar products associated with different processors. The parallel version of the SCMG preconditioners relies on parallel constituents at each level k (15): residual computation, evaluation of restriction and interpolation operators, and the relaxation scheme. The residual computation involves a parallel evaluation of the grid operator A^k, which is implemented via the "ghost cell" technique, provided that all operators A^k, $k = 1, \dots, M-1$, and associated "ghost cells" are known. Parallel implementation of the restriction and interpolation operators is straightforward, since they are defined on the basis of underlying meshes and are known explicitly. The relaxation schemes, LSOR and point SOR sweeps, are parallelized as follows. Since the (L)SOR iteration is an extrapolated (block) Gauss–Seidel preconditioner as defined in (21), its parallelization reduces to parallelization

of the Gauss–Seidel method. We recall that the vertical cell columns are ordered in the red-black fashion. Evaluation of the red-black block Gauss–Seidel preconditioner implies the inversion of the diagonal blocks and the vector-matrix multiplication for the off-diagonal block. The diagonal blocks are block diagonal matrices, their blocks being associated with vertical cell columns. Systems with the diagonal blocks may be solved very efficiently and in parallel. The vector-matrix multiplication for the off-diagonal block is parallelized by the "ghost cell" technique, since the matrix is a part of the reduced Jacobian A and is known explicitly.

Within the IPARS, the alternative to the SCMG preconditioner is the two-stage combinative preconditioner [4, 14, 17]. The preconditioner is applied to the modified (decoupled) Jacobian system. At the first stage the pressure block $\bar{A}$ is preconditioned; at the second stage the resulting residual for pressure and concentrations is preconditioned by $(\hat{A}^H)^{-1}$ and added to the preconditioned pressure. The preconditioner for the pressure-concentration residual is one LSOR iteration. Its parallelization was discussed above. Almost all implemented preconditioning techniques for the pressure block deal with parts of the system matrix, and the "ghost cell" technique is sufficient for their parallel implementation. The remaining method (separable preconditioner) is based on its private multilevel communication scheme and predefined slicewise decomposition of data. In this case, the solution technique requires a consistency with the data allocation among processors. We have chosen the parallel and single processor versions of a separable solver discussed in [24]. The communication pattern of this method is complicated and is hidden from the user of the parallel solver.

Parallelization of the algebraic multilevel methods is done in the domain decomposition framework [31]. We use a two-level hybrid II Schwarz preconditioner [20, 27] defined via its inverse

$$B = B_{AS} + B_0(I - \bar{A}B_{AS}). \tag{22}$$

Here, B_{AS} stands for the additive Schwarz preconditioner with the minimal overlap and B_0 denotes a coarse space preconditioner.

The construction of B_{AS} is based on the algebraic multilevel preconditioners. Let the matrix $\bar{A}$ be split into blocks according to data allocation among P processors as follows:

$$\bar{A} = \begin{pmatrix} \bar{A}_{11} & \dots & \bar{A}_{1P} \\ \bar{A}_{21} & \dots & \bar{A}_{2P} \\ \vdots & \ddots & \vdots \\ \bar{A}_{P1} & \dots & \bar{A}_{PP} \end{pmatrix},$$

and let B_{AS} be a block diagonal matrix whose blocks are algebraic multilevel preconditioners of diagonal blocks of $\bar{A}$, $B_{AS} = blockdiag\{B_{11}, \dots, B_{PP}\}$. Since the matrices $\bar{A}_{ij}$, $i \neq j$, have nonzero values only in the close-to-interface cells, and $B_{ii}A_{ii} \simeq I$, the preconditioner B_{AS} damps the error in inner cells, but, in close-to-interface cells, the remaining error is large. The coarse space preconditioner is designed to damp this error and to coordinate mean subdomain values in order to damp the error propagation in the global scale. Let the restriction operator R be defined componentwise,

$$\begin{aligned} (Ru)_i &= \sum_{j \in \mathcal{N}_i} u_j, \ i = 1, \dots, P, \\ (Ru)_{j+P} &= u_j, \quad j \in \Gamma_b, \end{aligned}$$

TABLE 6
Oil-water simulation on IBM SP.

	Pressure block prec.	8 procs.	16 procs.	32 procs.
#GMRES	$\bar{A}_{LSOR(6)}$	1652	1628	1679
CPU		7894	3760	1992
#GMRES / #Newt		41	40.7	42
speed-up(8)		1.0	2.0	4.0
#GMRES	$\bar{A}_{Sp}$	259	258	
CPU		1148	617	
#GMRES / #Newt		6.5	6.5	
speed-up(8)		1.0	1.85	
#GMRES	$\bar{A}_{AMG,2}$	397	422	498
CPU		5091	1064	625
#GMRES / #Newt		10.7	11.4	13.4
speed-up(16)		-	1.0	2.0

where Γ_b stands for the set of close-to-interface cells and $\mathcal{N}_i$ stands for the set of interior cells at processor i. If we denote by B_c an approximate inverse (preconditioner) of the coarse space matrix $R\bar{A}R^T$, the coarse space preconditioner is defined by $B_0 = R^T B_c R$ and the overall preconditioner (22) is then

$$B = B_{AS} + R^T B_c R(I - \bar{A}B_{AS}). \tag{23}$$

The approximate inversion of $R\bar{A}R^T$ is implemented in terms of several LSOR sweeps. The arithmetical price of one LSOR sweep is proportional to the number of close-to-interface cells, which is much less than the number of all cells (all interior cells in each subdomain are represented by one degree of freedom). Thus, even a large number of LSOR sweeps is affordable for the coarse space system. Actually, due to aggregation, the coarse space matrix $R\bar{A}R^T$ is much better conditioned than its full space counterpart $\bar{A}$. In practice, we need very few LSOR iterations (six iterations are used in experiments) to provide a good approximate inverse B_c to $R\bar{A}R^T$. Parallelization of the LSOR method was discussed earlier; parallelization of B_{AS} is trivial since it is the block diagonal matrix.

We notice that the coarse space preconditioner is unusual. The number of its degrees of freedom is much larger than that for conventional methods [11, 20], since its purpose is not only to "coordinate" the additive Schwarz subspaces, but also to damp the error on the interfaces. From this standpoint it may be considered as a variation of a two-stage method [43]. We remark that the method is insensitive to an irregular shape of the interface and is affected by the number of subdomains very weakly (cf. Table 6).

In Table 6 we show the parallel properties of the combinative technique with different pressure block preconditioners. Here, we consider only the oil-water model, the simulation case similar to Cases 1–4, on the mesh $40 \times 200 \times 125$ (1,000,000 grid cells). The stopping criterion for the GMRES iterations is the relative reduction of the Euclidean residual norm for each component equation in (11) by a factor of 20. We present the accumulated number of linear iterations (#GMRES, #MG), the execution time of the linear solver (CPU), the average number of linear iterations per Newton step (#GMRES / #Newt, #MG / #Newt), and the speed-up of one linear iteration with respect to execution on t processors (speed-up(t)). The measurements were obtained on 8, 16, and 32 processors of an IBM SP (160 MHz). In the case of the AMG, we attribute the speed-up to 16-processor execution because of memory effects

in the 8-processor execution. All the methods provide a good speed-up. Another peculiarity of the AMG case is that the number of iterations grows slightly with the increase in the number of processors. We explain this by the domain decomposition effect: the method is not mathematically the same on different sets of processors.

7. Conclusions. We considered several issues related to the solution of systems of nonlinear partial differential equations arising in the fully implicit simulation of multiphase flow in porous media. Two models have been examined, the oil-water and the black-oil. In this paper, we addressed the performance of several linear solvers. The solvers use the preconditioned GMRES method applied to the Jacobian system. The SCMG preconditions the coupled system of Jacobian equations. It is based on the rectangular topological structure of the mesh and logically fits the classical multigrid scheme. Its particular features are the LSOR presmoother in the vertical direction and the very fast reduction of the problem dimension due to the aggregation in the vertical direction at the first level. On other levels, the two-dimensional V-cycle is equipped with several pre- and postsmoothing SOR sweeps. As a result, the multigrid cycle is very cheap computationally, and the convergence rate does not depend on the vertical mesh but depends on the areal mesh. In addition, the method is sensitive to strong heterogeneity of the permeability field. In order to reduce the above dependencies, we augment the SCMG with an additional correction, which is a few LSOR sweeps for the aggregated Jacobian system. The aggregated system is obtained by "four by four" coarsening in horizontal planes. The additional correction results in a lesser number of iterations but a higher complexity of iterations. Advantages of the additional correction are seen on large meshes.

The alternative to the SCMG is the two-stage preconditioner. It is applied to the modified Jacobian system. The goal of the modification is to decouple a pressure equation from concentration equations. After modification, certain matrix entries are neglected so that the pressure diagonal block may be preconditioned independently of others. The goal of the first stage is to precondition the pressure equation, and the goal of the second stage is to construct the global preconditioner for the modified system. The global preconditioner is the combinative one, since it uses a strong (aggressive) preconditioner for the pressure and a weak preconditioner for the pressure-concentration feedback. We motivated and examined several types of global preconditioners. For the pressure block preconditioning, we tested several types of contemporary methods, such as AMG, MLILU, direct solvers for the separable operators, and LSOR inner iterations. The first two methods do not rely on the underlying mesh, while the others use the rectangular structure of the mesh. An additional advantage of AMG and MLILU is that they are robust with respect to heterogeneity of the permeability field. The drawback is a high initialization expense. In order to compensate the initialization with fast iterations, we apply one-level geometric projection with LSOR smoothing and reduce the order of the pressure block. The reduction turns out to be very efficient. The most attractive features of the combinative technique are a potential independence of the underlying grid, the extraction of the pressure block to be preconditioned thoroughly, and robustness to a permeability heterogeneity.

Comparing the SCMG and the fast two-stage preconditioners, we notice they have approximately the same efficiency for the oil-water model. For the black-oil model, the SCMG preconditioner seems to be more appealing. From a parallelization standpoint, the best speed-up is attributable to methods based on inner LSOR iterations. Development of a parallel version of the SCMG preconditioner is in progress.

Acknowledgments. This work is based on the IPARS solution routines of J. Wheeler and H. C. Edwards. J. Eaton and P. Quandalle contributed the LSOR preconditioner for the pressure block. The authors are very grateful to R. Dean, M. Peszynska, and G. Xiuli for fruitful discussions and to K. Stuben, T. Rossi, and J. Toivanen for permission to use their software.

REFERENCES

[1] A. Abakumov, A. Yeremin, and Yu. Kuznetsov, *Efficient fast direct method of solving Poisson's equation on a parallelepiped and its implementation in an array processor*, Soviet J. Numer. Anal. Math. Modelling, 3 (1988), pp. 1–20.

[2] K. Aziz and A. Settari, *Petroleum Reservoir Simulation*, Applied Science Publishers Ltd., London, 1979.

[3] R. Bank and C. Wagner , *Multilevel ILU decomposition*, Numer. Math., 82 (1999), pp. 543–576.

[4] G. Behie and P. Vinsome, *Block iterative methods for fully implicit reservoir simulation*, Society of Petroleum Engineering Journal, 22 (1982), pp. 658–668.

[5] J. B. Bell, J. A. Trangenstein, and G. R. Shubin, *Conservation laws of mixed type describing three-phase flow in porous media*, SIAM J. Appl. Math., 46 (1986), pp. 1000–1017.

[6] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, and J. W. Ruge, *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1886–1908.

[7] K. H. Coats, *A note on Impes and some Impes-based simulation models*, SPE 49774, in Proceedings of the SPE 1999 Resrv. Simul. Symp., Houston, Texas, February 14–17, 1999, pp. 21–39.

[8] J. Dendy, S. McCormick, J. Ruge, T. Russel, and S. Schaffer, *Multigrid methods for three-dimensional petroleum reservoir simulation*, SPE 18409, in Proceedings of the SPE 1989 Resrv. Simul. Symp., Houston, Texas, February 6–8, 1989, pp. 19–25.

[9] J. Dendy, *Multigrid methods for petroleum reservoir simulation on SIMD machines,* SPE 25243, in Proceedings of the SPE 1993 Resrv. Simul. Symp., New Orleans, LA, February 28–March 3, 1993, pp. 97–104.

[10] H. C. Edwards, *A Parallel Multilevel-Preconditioned GMRES Solver for Multiphase Flow Models in the Implicit Parallel Accurate Reservoir Simulator,* TICAM report 98-04, The University of Texas at Austin, 1998. Available online at http://www.ticam.utexas.edu/reports/1998/index.html

[11] E. W. Jenkins, C. E. Kees, C. T. Kelley, and C. T. Miller, *An aggregation-based domain decomposition preconditioner for groundwater flow,* SIAM J. Sci. Comput., 23 (2001), pp. 430–441.

[12] J. E. Killough, *Ninth SPE comparative solution project: A reexamination of black-oil simulation,* SPE 29110, in Proceedings of the 13th SPE Resrv. Simul. Symp., San Antonio, TX, 1995, pp. 135–147.

[13] J. Killough and M. Wheeler, *Parallel iterative linear equation solvers: An investigation of domain decomposition algorithms for reservoir simulation,* SPE 16021, presented at the SPE 1987 Resrv. Simul. Symp., San Antonio, Texas, February 1–4, 1987, pp. 293–312.

[14] H. Klie, *Krylov-Secant Methods for Solving Large Scale Systems of Coupled Nonlinear Parabolic Equations*, Ph.D. Thesis, Rice University, Houston, TX, 1996.

[15] Yu. Kuznetsov, *Matrix computational processes in subspaces*, Comput. Math. Appl. Sci. Engrg., 6 (1984), pp. 15–31.

[16] Yu. Kuznetsov and G. Marchuk, *Iterative methods and quadratic functionals,* Nauka, Novosibirsk, 1972 (in Russian). French translation: *Methodes iteratives et fonctionnelles quadratiques*, in Methodes de L'Informatique-4, J. L. Lions and G. I. Marchouk, eds., Dunod, Paris, 1974, pp. 3–132.

[17] S. La Croix, Y. Vassilevski, and M. Wheeler, *Iterative Solvers of the Implicit Parallel Accurate Reservoir Simulator (IPARS),* I*: Single Processor Case*, TICAM report 00-28, The University of Texas at Austin, Austin, TX, 2000. Available online at http://www.ticam.utexas.edu/reports/2000/index.html.

[18] S. La Croix, Y. Vassilevski, and M. Wheeler, *Decoupling preconditioners in the implicit parallel accurate reservoir simulator (IPARS)*, Numer. Linear Algebra Appl., 8 (2001), pp. 537–549.

[19] L. W. Lake, *Enhanced Oil Recovery,* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[20] J. Mandel, *Hybrid domain decomposition with unstructured subdomains*, Contemp. Math., 157 (1994), pp. 103–112.
[21] M. Parashar, J. A. Wheeler, G. Pope, K. Wang, and P. Wang, *A new generation EOS compositional reservoir simulator. Part* II: *Framework and multiprocessing,* in Proceedings of the Fourteenth SPE Symposium on Reservoir Simulation, Dallas, Texas, Society of Petroleum Engineers, Richardson, TX, 1997, pp. 31–38.
[22] D. Peaceman, *Fundamentals of Numerical Reservoir Simulation*, Elsevier, Amsterdam, 1977.
[23] *Proceedings of the* 15*th Symposium on Reservoir Simulation*, Society of Petroleum Engineers Inc., Richardson, TX, 1999.
[24] T. Rossi and J. Toivanen, *A parallel fast direct solver for block tridiagonal systems with separable matrices of arbitrary dimension*, SIAM J. Sci. Comput., 20 (1999), pp. 1778–1796.
[25] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[26] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[27] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.
[28] K. Stuben, *Algebraic multigrid (AMG): Experiences and comparisons*, Appl. Math. Comput., 13 (1983), pp. 419–452.
[29] J. A. Trangenstein and J. B. Bell, *Mathematical structure of the black-oil model for petroleum reservoir simulation*, SIAM J. Appl. Math., 49 (1989), pp. 749–783.
[30] P. Vassilevski, *Fast algorithm for solving a linear algebraic problem with separable variables*, C.R. Acad. Bulgare Sci., 37 (1984), pp. 305–308.
[31] Yu. Vassilevski, *Iterative Solvers for the Implicit Parallel Accurate Reservoir Simulator (IPARS),* II: *Parallelization Issues*, TICAM report 00-33, The University of Texas at Austin, Austin, TX, 2000. Available online at http://www.ticam.utexas.edu/reports/2000/index.html.
[32] J. R. Wallis, R. P. Kendall, and T. E. Little, *Constrained residual acceleration of conjugate residual methods,* SPE 13536, in Proceedings of the SPE 1985 Resrv. Simul. Symp., Dallas, Texas, February 10-13, 1985, pp. 415–428.
[33] P. Wang, I. Yotov, M. F. Wheeler, T. Arbogast, C. N. Dawson, M. Parashar, and K. Sepehrnoori, *A new generation EOS compositional reservoir simulator. Part* I*: formulation and discretization,* in Proceedings of the Fourteenth SPE Symposium on Reservoir Simulation, Dallas, Texas, Society of Petroleum Engineers, Richardson, TX, 1997, pp. 55–64.
[34] J. W. Watts, *An iterative matrix solution method suitable for anisotropic problems*, Society of Petroleum Engineering Journal, 11 (1971), pp. 47–51.
[35] J. W. Watts, *A method for improving line successive overrelaxation in anisotropic problems – a theoretical analysis*, Society of Petroleum Engineering Journal, 255 (1973), pp. 105–118.
[36] J. W. Watts, *A compositional formulation of the pressure and saturation equations*, SPE Reservoir Engineering, 1 (1986), pp. 243–252.
[37] A. Weiser and M. F. Wheeler, *On convergence of block-centered finite differences for elliptic problems*, SIAM J. Numer. Anal., 25 (1988), pp. 351–375.
[38] T. F. Russell and M. F. Wheeler, *Finite element and finite difference methods for continuous flows in porous media*, in The Mathematics of Reservoir Simulation, R. E. Ewing, ed., SIAM, Philadelphia, 1983, pp. 35–106.
[39] M. F. Wheeler and M. Peszynska, *Computational engineering and science methodologies for modeling and simulation of subsurface applications*, Advances in Water Resources, 25 (2002), pp. 1147–1173.
[40] M. F. Wheeler, J. A. Wheeler, and M. Peszynska, *A distributed computing portal for coupling multi-physics and multiple domains in porous media,* in Computational Methods in Water Resources, L. R. Bentley, J. F. Sykes, C. A. Brebbia, W. G. Gray, and G. F. Pinder, eds., A.A. Balkema, The Netherlands, 2000, pp. 167–174.
[41] J. Wheeler and R. Smith, *Reservoir simulation on a hypercube,* SPE 19804, in Proceedings of the SPE 1989 Ann. Tech. Conf., San Antonio, Texas, October 8-11, 1989.
[42] J. Wheeler, *IPARS User's Manual*, Tech. report CSM, TICAM, The University of Texas at Austin, Austin, TX, 2000.
[43] J. Xu, *A new class of iterative methods for nonselfadjoint or indefinite problems*, SIAM J. Numer. Anal., 29 (1992), pp. 303–319.