

Parallel Multigrid and Domain Decomposition Algorithms for Elliptic Equations

Lawrence C. Cowsar*
Alan Weiser*
Mary F. Wheeler*

Abstract. We discuss two algorithms for modeling elliptic equations on hypercubes: multigrid and domain decomposition. Both schemes have been designed for cell-centered finite differences. One of these approaches is a semicoarsening multigrid algorithm for anisotropic non-self-adjoint elliptic partial differential equations; this scheme is an extension [15] of earlier work of J. Dendy *et al.* [6] for symmetric problems. The second approach is a domain decomposition procedure first introduced by Glowinski and Wheeler [8] for mixed finite element methods and then later modified by Cowsar and Wheeler [3] to include multilevel acceleration.

Performance of the codes has been compared on the Intel i860 hypercube. Both procedures perform well for modeling the pressure or hydraulic head equation which arises in flow in porous media problems. In both procedures in the hypercube implementation the domain is decomposed into subdomains, with one subdomain being assigned to each processor. We discuss numerical experiments performed with these methods. In particular we emphasize scaling and parallel efficiencies.

1. Introduction. Let Ω be a rectangular or box-shaped domain in 2 or 3 dimensions with boundary Γ . The equation to be solved is the following:

$$(1) \quad u(x) = -A(x)\nabla p(x)$$

in Ω ,

$$(2) \quad \nabla \cdot u(x) = f(x)$$

in Ω , with boundary conditions

$$(3) \quad u \cdot n = 0$$

on Γ , with the additional condition that, say,

$$(4) \quad \int_{\Omega} p(x) dx = 0$$

* Department of Mathematical Sciences, Rice University

to fix p uniquely.

A physical interpretation of this equation is that of a material balance. The pressure or hydraulic head p induces a velocity or flux vector u with tensor coefficient matrix A , according to, say, Darcy's law. The divergence of the flux is equal by material balance to the sources (and sinks) $f(x)$, corresponding to well injection rates in reservoir simulation. The boundary conditions $u \cdot n = 0$ correspond to no flow into or out of the domain Ω .

We solve (1) using a mixed finite element method. We choose appropriate finite dimensional spaces V and W such that

$$(5) \quad V \subset H(\Omega, \text{div}) = \{v \in (L^2(\Omega))^D \mid \nabla \cdot v \in L^2(\Omega)\}$$

and

$$(6) \quad W \subset L^2(\Omega)$$

for a D -dimensional problem, where we require the inclusion

$$(7) \quad \text{div}(V) \subset W.$$

Then the mixed finite element method finds $U \in V$ and $P \in W$ such that

$$(8) \quad \int_{\Omega} A^{-1}U \cdot v dx = \int_{\Omega} P \nabla \cdot v dx - \int_{\Gamma} qv \cdot nds$$

for all $v \in V$ and

$$(9) \quad \int_{\Omega} \nabla \cdot U w dx = \int_{\Omega} f w dx$$

for all $w \in W$.

If we discretize by intervals in the coordinate directions in $x = (x_1, x_2)$ or (x_1, x_2, x_3) , choose W to be the space of piecewise constants on the resulting set of rectangular gridblocks, and choose each component of V to be the space of functions which are continuous piecewise linear in the given component direction and piecewise (discontinuous) constants in the other component directions, the lowest-order Raviart-Thomas mixed method [13] results. If we further use approximate quadrature rules corresponding to tensor product midpoint rules for all integrals except for the trapezoidal rule in the given component direction for the $\int_{\Omega} A^{-1}U \cdot v dx$ term, the standard block-centered finite difference or finite volume method is obtained [14,17]. The linear system is typically derived by pre-eliminating the velocities in terms of the pressures, resulting in a linear system for one unknown pressure per gridblock.

The target area we have in mind is petroleum reservoir simulation. The linear systems that arise in this area are characterized by several properties. They tend to be large (on order of tens of thousands of gridblocks) and a different system needs to be solved for each timestep (on order of thousands of timesteps). The coefficient matrices A tend to be anisotropic, with much stronger coupling in the vertical than horizontal directions because of the underlying "pancake-shaped" gridblock geometry, and strongly discontinuous functions of x because of the variations in the permeability of the reservoir rock. Various linear solution methods are in use in reservoir simulation, e.g., [1,7,16]. Among the methods proposed in recent reservoir simulation literature, semicoarsening multigrid [6] and domain decomposition (e.g., [12]) seem promising.

In Section 2, we describe the semicoarsening multigrid method. In Section 3, we describe the domain decomposition method. In Section 4, we compare the two methods on three test cases. In Section 5, we summarize our results.

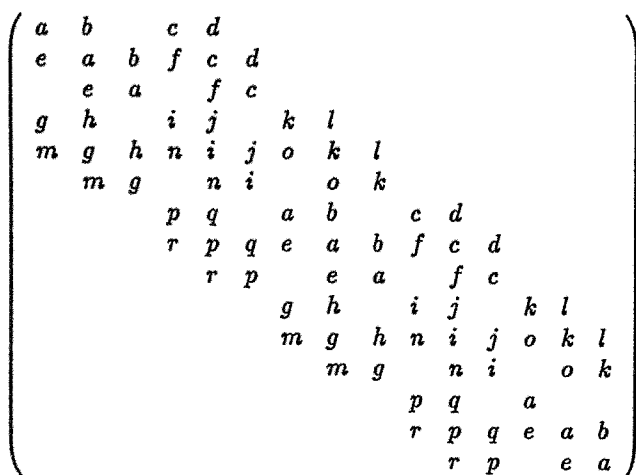


FIG. 1. 2-d matrix nonzero structure ($n_1=3, n_2=5$).

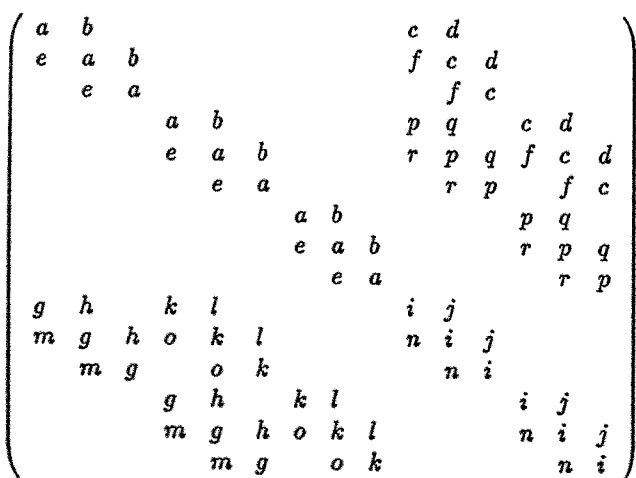


FIG. 2. 2-d red-black matrix nonzero structure.

2. The semicoarsening multigrid method. Consider a two-dimensional problem (1-4) with a discretization resulting in a nine-point operator. The resulting nine-diagonal matrix is depicted in Figure 1. Let P be the permutation matrix ordering lines of unknowns red-black. The resulting red-black matrix $P^T A P$ is depicted in Figure 2.

In block form, the red-black linear system is

$$(10) \quad Ax = b$$

or

$$(11) \quad \begin{pmatrix} A_{rr} & A_{rb} \\ A_{br} & A_{bb} \end{pmatrix} \begin{pmatrix} x_r \\ x_b \end{pmatrix} = \begin{pmatrix} b_r \\ b_b \end{pmatrix}.$$

The nonzeros in Figure 2 are labeled so that A_{rr} nonzeros are labeled e, a, b , A_{rb} nonzeros are labeled r, p, q, f, c, d , A_{br} nonzeros are labeled m, g, h, o, k, l , and A_{bb} nonzeros are

labeled n, i, j . One way to solve (11) is to form and solve the smaller Schur complement linear system

$$(12) \quad A_s x_b = b_s$$

where

$$(13) \quad A_s = A_{bb} - A_{br} A_{rr}^{-1} A_{rb}$$

and

$$(14) \quad b_s = b_b - A_{br} A_{rr}^{-1} b_r,$$

and then backsolve

$$(15) \quad x_r = A_{rr}^{-1} (b_r - A_{rb} x_b).$$

Now consider a two-level multigrid method for the original system, where the black unknowns are the coarse grid unknowns. The steps in such a method are:

- i) smooth on the fine grid
- ii) transfer the fine grid residual to the coarse grid and solve for the coarse grid correction
- iii) transfer the correction back to the fine grid and add it in to the current solution
- iv) smooth again on the fine grid.

In the black-box multigrid framework [5] it is customary to construct the coarse grid system $A_c x_c = b_c$ as follows:

$$(16) \quad \begin{pmatrix} T_{br} & I_{bb} \end{pmatrix} \begin{pmatrix} A_{rr} & A_{rb} \\ A_{br} & A_{bb} \end{pmatrix} \begin{pmatrix} T_{rb} \\ I_{bb} \end{pmatrix} \begin{pmatrix} x_c \end{pmatrix} = \begin{pmatrix} T_{br} & I_{bb} \end{pmatrix} \begin{pmatrix} b_r \\ b_b \end{pmatrix}$$

Here b_r and b_b denote the current residuals, and are only equal to the original right hand side if the current iterate is zero. I_{bb} denotes the coarse grid identity matrix: Pure injection is used for interpolating coarse grid unknowns to black unknowns on the fine grid.

If $T_{br} = -A_{br} A_{rr}^{-1}$ and $T_{rb} = -A_{rr}^{-1} A_{rb}$, then the coarse grid system for two-level multigrid is exactly the Schur complement system $A_s x_s = b_s$. However, the Schur complement matrix is much denser than the original matrix. Fill occurs for two reasons. First, A_{rr}^{-1} has dense diagonal blocks, so T_{br} and T_{rb} have dense diagonal blocks. Second, extra fill occurs in direction 1. For instance, the black unknowns corresponding to $(i1, i2) = (2, 1)$ and $(4, 3)$ are both connected to the red unknown corresponding to $(i1, i2) = (3, 2)$ via T_{rb} and T_{br} terms. Elimination of red terms results in direct connections between $(2, 1)$ and $(4, 3)$ in A_s even though their indices differ by more than 1 in direction 1. The only way to avoid this extra fill is to allow connections in T_{br} and T_{rb} in direction 2 only. The locations in Figure 2 corresponding to connections in direction 2 only are the locations for diagonals c, g, k , and p .

Some new notation must now be introduced to deal with connections in direction 2 only. Let an "l" (resp. "r") superscript appended to A_{rb} or A_{br} denote the result of zeroing out all entries except those corresponding to connection of a black unknown to a red unknown with a smaller (resp. larger) index in direction 2. Then $A_{rb} = A_{rb}^l + A_{rb}^r$ and $A_{br} = A_{br}^l + A_{br}^r$. For example, in Figure 2, A_{rb}^l may contain nonzeros on diagonals c, d , and f , A_{rb}^r may contain nonzeros on diagonals p, q , and r , A_{br}^l may contain nonzeros on diagonals g, h , and m , and A_{br}^r may contain nonzeros on diagonals k, l , and o . Similarly, let $diag_{rb}^l(v)$ (resp. $diag_{rb}^r(v)$, $diag_{br}^l(v)$, $diag_{br}^r(v)$) denote the matrix with the same nonzero structure as A_{rb} (resp. A_{rb}, A_{br}, A_{br}) with the only nonzero entries obtained from vector v and put on

the diagonal corresponding to connections of black unknowns to red unknowns with lower (resp. higher, lower, higher) index in direction 2 and with the same index in direction 1. For example, in Figure 2, $diag_{rb}^l(v)$ (resp. $diag_{rb}^r(v)$, $diag_{br}^l(v)$, $diag_{br}^r(v)$) puts nonzeros in the locations of diagonal c (resp. p , g , k).

With this notation, we define the following transfer operators due to Steve Schaffer at the New Mexico Institute of Mining and Technology [4]:

$$(17) \quad T_{rb} = -(diag_{rb}^l(A_{rr}^{-1}A_{rb}e) + diag_{rb}^r(A_{rr}^{-1}A_{rb}e))$$

$$(18) \quad T_{br} = -(diag_{br}^l(e^T A_{br}^l A_{rr}^{-1}) + diag_{br}^r(e^T A_{br}^r A_{rr}^{-1}))$$

where e is the vector of all ones.

We use (17) and (18) for transfer operators. We take the initial guess for the solution to (10) on a given grid to be

$$(19) \quad x_0 = \frac{b^T e}{e^T A e} e$$

rather than zero. For our smoother we use red-black line Gauss-Seidel with lines oriented in direction 1. This completes the specification of the basic two-level two-dimensional semicoarsening multigrid method. The adjective semicoarsening is used because the coarse grid is only coarsened in direction 2, while usual "full coarsening" multigrid involves coarsening in both directions 1 and 2 simultaneously.

The method works particularly well for strongly anisotropic problems [15].

Now consider a three-dimensional model problem with a 15-point operator (a 3 point operator in direction 3 tensored with a 5 point operator in directions 1 and 2). Let P be the permutation matrix ordering planes of unknowns red-black.

Again the same two considerations cause fill in the Schur complement matrix. Our notation to deal with connections in direction 3 is similar to the two-dimensional case.

We would again like to use transfer operators (17) and (18). However, now A_{rr} is a 5-point operator which is expensive to invert. Hence, following Schaffer [4], we use one cycle of our two-dimensional semicoarsening multigrid method to approximately solve the A_{rr} systems needed in constructing T_{rb} and T_{br} .

We use (19) for initial guesses and approximate red-black Gauss-Seidel plane relaxation for our fine-grid smoothing step, where the two-dimensional semicoarsening multigrid method approximately solves the plane Gauss-Seidel equations. This completes the specification of the basic two-level three-dimensional semicoarsening multigrid method.

Note that a fine grid three-dimensional 7-point operator forms 15-point operators on the coarser three-dimensional grids, and a fine grid two-dimensional 5-point operator forms 9-point operators on the coarser two-dimensional grids. Again this method works particularly well for strongly anisotropic problems [15].

We use two standard multigrid cycling approaches. The "V-cycle" (Figure 3) starts on the finest grid with a two-level method and solves the resulting coarser grid system recursively with another two level method, and so on until the coarsest grid system with one unknown is solved directly. The "FMV-cycle" or full multigrid V-cycle (Figure 4) generates an initial guess for the fine grid system by preceding the fine-grid V-cycle recursively with another V-cycle on the next-finest grid, and so on so that the very first calculation is a direct solve on the coarsest grid.

We parallelize the semicoarsening multigrid method on the Intel i860 hypercube using domain decomposition [15]. We assign a rectangular subdomain to each processor. Processors exchange boundary information with neighboring subdomains as needed. A special

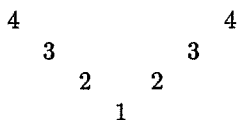


FIG. 3. *V-cycle*.

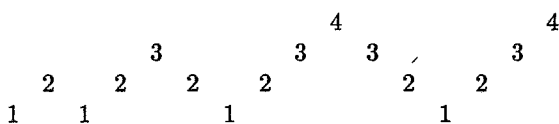


FIG. 4. *FMV-cycle*.

tridiagonal solver handles the line Gauss-Seidel solves — it consists of 2-level cyclic reduction coupled with burn-at-both-ends (BABE) elimination within each subdomain [11]. We use special techniques for grids “below C-level”, that is, grids that are so coarse they have no gridblocks for some processors (*cf.* [2,10]).

3. The domain decomposition method. We use a nonoverlapping Dirichlet domain decomposition method [3,8,9]. Assume the original linear system is symmetric and positive definite. To motivate the method, consider the grid depicted in Figure 5, with p 's representing pressures at the centers of gridblocks and λ 's representing pressures in thin interface blocks. Let the widths of the interface blocks shrink to zero. Then the Lagrange multipliers $\{\lambda\}$ enforce continuity of fluxes across interfaces, and the solution of the augmented system is the same as the solution of the original system without interfaces.

The resulting linear system for p 's and λ 's is of the form

$$(20) \quad \begin{pmatrix} M_{pp} & M_{p\lambda} \\ M_{\lambda p} & M_{\lambda\lambda} \end{pmatrix} \begin{pmatrix} p \\ \lambda \end{pmatrix} = \begin{pmatrix} b_p \\ b_\lambda \end{pmatrix}$$

and λ satisfies the Schur complement system

$$(21) \quad \bar{M}\lambda = \bar{b}$$

where

$$(22) \quad \bar{M} = M_{\lambda\lambda} - M_{\lambda p}M_{pp}^{-1}M_{p\lambda}$$

and

$$(23) \quad \bar{b} = b_\lambda - M_{\lambda p}M_{pp}^{-1}b_p.$$

\bar{M} is symmetric and positive definite since the original matrix is. Moreover, the condition number of \bar{M} is proportional to h^{-1} : there exist $C_0, C_1 > 0$ such that

$$(24) \quad C_0\|\lambda\|^2 \leq \lambda^T \bar{M}\lambda \leq \frac{C_1}{h}\|\lambda\|^2$$

where

$$(25) \quad \|\lambda\|^2 = \int_{\cup_i \Gamma_i} n^T A n \lambda^2 ds$$

and n is the outer normal to Γ_i [3,9].

p	p	λ	p	p
p	p	λ	p	p
λ	λ		λ	λ
p	p	λ	p	p
p	p	λ	p	p

FIG. 5. Pressures and Lagrange multipliers.

We solve the system $\bar{M}\lambda = \bar{b}$ by multigrid on interfaces. We never form \bar{M} explicitly, just apply it. Our current multigrid method obtains coarse grid systems by harmonic weighting of fine grid coefficients. The injection operator used is piecewise constant averaging, and the prolongation operator used is piecewise constant assignment. The smoother used is the conjugate residual method [7], and the subgrid solver is nested factorization with Orthomin acceleration [1]. The multigrid cycle procedure used is one FMV-cycle followed by subsequent V-cycles. On the Intel hypercube, two kinds of communication between processors take place – calculation of residuals on interfaces, and communication of dot products occur across processors during computation of smoothing on the interfaces.

4. Numerical results. We made runs with three test cases on the Intel iPSC/860 using the PGF77 compiler with the -O3 optimization option. The convergence tolerance was a reduction in the L^2 norm of the original residual by a factor of 10^6 .

Problem 1 is posed on the unit cube on a 48 by 48 by 8 grid, with $A = \text{diag}(1,1,100)$, sources $f(x) = \delta_{(1,1,1)} - \delta_{(0,0,0)}$ and boundary conditions $u \cdot n = 0$ on Γ . This problem represents a model with anisotropy, as would arise from a 10 to 1 ratio of horizontal to vertical grid block size in a reservoir simulation model.

Problem 2 is posed on the unit cube on a 24 by 24 by 24 grid, with $A = \text{diag}(a,a,a)$, $a=1$ except $a=.00001$ at simple flow barriers (Figure 6). The boundary conditions are $p(x_1, x_2, 1) = 1$, $p(x_1, x_2, 0) = 0$, and $u \cdot n = 0$ on the other four faces of Γ . This problem represents a model with discontinuous coefficients without a preferred direction of flow.

Problem 3 is posed on the unit square on an 80 by 80 grid, with $A = \text{diag}(a,a,a)$, a ranging from .6 to 12.7 (Figure 7). The boundary conditions are $p(0, x_2) = 1$, $p(1, x_2) = 0$, and $u \cdot n = 0$ at $x_2 = 0, 1$. This problem represents a model with more strongly discontinuous coefficients, as arises in reservoir simulation models of unfavorable mobility ratio displacement.

CPU time results are indicated in Tables 1-3. The numbers of processors in the three coordinate directions are indicated by $(nproc_1) \times (nproc_2) \times (nproc_3)$. MG, MGNC, and DD refer to the multigrid method, the multigrid method excluding CPU time for interprocessor communication, and the domain decomposition method, respectively.

The results indicate that the domain decomposition method scales much better than the multigrid method as the number of processors increases. This is because domain decomposition requires less interprocessor communication per iteration. Previous results with

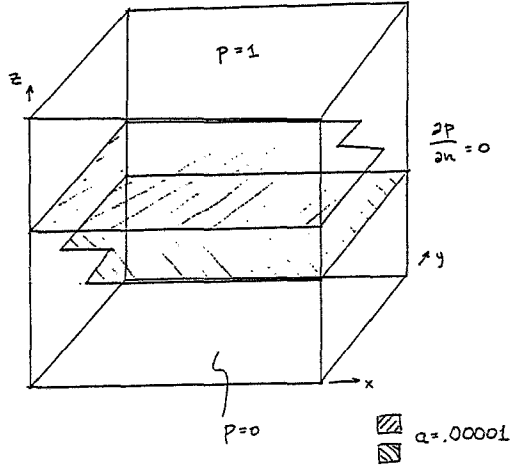


FIG. 6. Problem 2 specification

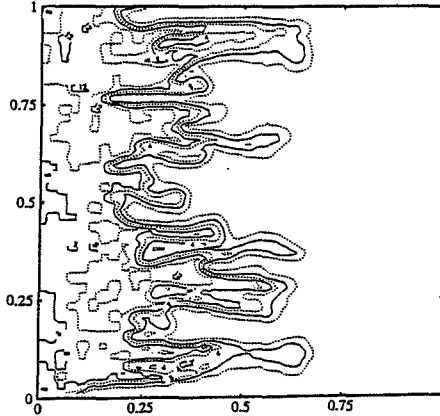


FIG. 7. Problem 3 coefficient a

TABLE 1
Problem 1 CPU times

	1x2x2	2x2x1	2x2x2	3x3x1	4x4x1
MG	12.4	20.7	18.1	18.2	19.2
MGNC	6.5	8.5	5.6	4.1	4.1
DD	.	18.6	10.2	9.2	4.8

TABLE 2
Problem 2 CPU times

	1x1x1	1x1x2	1x2x2	2x2x2	1x4x4	3x3x3
MG	13.9	10.8	11.6	16.9	11.3	17.5
MGNC	13.9	9.6	7.4	6.6	3.6	3.1
DD	.	34.6	14.0	7.3	3.9	3.2

TABLE 3
Problem 3 CPU times

	1x1	1x2	1x4	2x2	1x8	4x4	2x8
MG	1.1	1.2	1.1	1.8	0.9	1.9	1.5
MGNC	1.1	1.0	0.6	0.9	0.4	0.4	0.4
DD	.	13.1	5.3	7.3	3.8	2.3	.

the multigrid method on the Intel iPSC/2 indicated better speedups [15] - the slowdown on the i860 is due to the increased ratio of communication time to computation time. Relative results on the IPSC/2 would be closer to the MGNC rows than the MG rows. Multigrid is faster than domain decomposition with a small number of processors, because domain decomposition reverts to nested factorization on one processor, which requires more iterations than multigrid for a fine grid. Domain decomposition requires relatively more time for the problem with more strongly discontinuous coefficients. This is predominantly due to more iterations being required by the nested factorization subgrid solver. Both methods require few outer iterations to solve all three problems.

5. Summary. A semicoarsening multigrid method and a domain decomposition method for solving elliptic equations are presented. Both methods converge fast in terms of outer iterations for anisotropic models and models with strongly discontinuous coefficients. The advantages of the multigrid method are that it is defined for nonsymmetric models, has a systematic procedure for generating transfers and coarse grid operators, has an optimal convergence rate with one processor, and seems more robust than nested factorization as a subgrid solver for problems with discontinuous coefficients. The advantages of the domain decomposition method are that it requires much less interprocessor communication per iteration, much less storage, and requires no special tridiagonal solver.

Based on these relative advantages, we propose for future investigation a hybrid method in which semicoarsening multigrid is used to determine transfer operators, coarse grid linear systems, and subgrid solvers within the context of domain decomposition with multigrid on interfaces.

REFERENCES

- [1] Appleyard, J. R., and Cheshire, I. M., *Nested Factorization*, paper SPE 12264 presented at the Seventh Society of Petroleum Engineers Symposium on Reservoir Simulation, Nov. 16-18, 1983.
- [2] Briggs, B., Hart, L., McCormick, S., and Quinlan, D., *Multigrid Methods on a Hypercube*, in *Multigrid Methods: Theory, Applications, and Supercomputing*, S. F. McCormick, ed., Marcel Dekker, Inc., 1988.
- [3] Cowsar, L. C., and Wheeler, M. F., *A 3D Parallel Implementation of Domain Decomposition for Mixed Finite Elements*, to be published in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1991.
- [4] Dendy, J. E., Jr., private communication.
- [5] Dendy, J. E., Jr., *Black Box Multigrid*, *J. Comp. Phys.*, 48 (1982), 366-386.
- [6] Dendy, J. E., Jr., McCormick, S. F., Ruge, J. W., Russell, T. F., and Schaffer, S., *Multigrid Methods for Three-Dimensional Petroleum Reservoir Simulation*, paper SPE 18409 presented at the Society of Petroleum Engineers Symposium on Reservoir Simulation in Houston, TX, Feb. 6-8, 1989.
- [7] Eisenstat, S. C., Elman, H. C., and Schultz, M. H., *Block-Preconditioned Conjugate-*

- Gradient-Like Methods for Numerical Reservoir Simulation*, paper SPE 13534 presented at the Eighth Symposium on Reservoir Simulation, Dallas, Feb. 10-13, 1985.
- [8] Glowinski, R., and Wheeler, M. F., *Domain Decomposition and Mixed Finite Element Methods for Elliptic Problems*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. Meurant, J. Periaux eds., SIAM, Philadelphia, 1988, pp. 144-172.
- [9] Glowinski, R., Kinton, W., and Wheeler, M. F., *Acceleration of Domain Decomposition Algorithms for Mixed Finite Elements by Multi-Level Methods*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Periaux, O. B. Widlund eds., SIAM, Philadelphia, 1990, pp. 263-289.
- [10] Hempel, R., and Schuller, A., *Experiments With Parallel Multigrid Algorithms Using the SUPRENUM Communications Subroutine Library*, GMD Studie 141, St. Augustin, April 1988.
- [11] Johnsson, S. L., *Solving Tridiagonal Systems on Ensemble Architectures*, SIAM J. Sci. Stat. Comp., 8 (1987), 354-392.
- [12] Killough, J. E., and Wheeler, M. F., *Parallel Iterative Linear Equation Solvers: An Investigation of Domain Decomposition Algorithms for Reservoir Simulation*, paper SPE 16021 presented at the Ninth Symposium on Reservoir Simulation, San Antonio, Feb. 1-4, 1987.
- [13] Raviart, P. A., and Thomas, J. M., *A Mixed Finite Element Method for Second Order Elliptic Problems*, in Mathematical Aspects of the Finite Element Method, Lecture Notes in Mathematics, Springer-Verlag, Heidelberg, 1977.
- [14] Russell, T. F., and Wheeler, M. F., *Finite Element and Finite Difference Methods for Continuous Flow Problems*, in The Mathematics of Reservoir Simulation, R. E. Ewing, ed., Frontiers in Science, SIAM, Philadelphia, 1983.
- [15] Smith, R. A., and Weiser, A., *Semicoarsening Multigrid on a Hypercube*, Center for Research on Parallel Computation Technical Report 91112, Rice University, 1991.
- [16] Watts, J. W., *An Iterative Matrix Solution Method Suitable for Anisotropic Problems*, Soc. Pet. Eng. J., 11 (1971), 47-51.
- [17] Weiser, A., and Wheeler, M. F., *On Convergence of Block-Centered Finite Differences for Elliptic Problems*, SIAM J. Numer. Anal., 25 (1988), pp. 351-375.