CHAPTER 17

A Comparison of Some Domain Decomposition Algorithms for Nonsymmetric Elliptic Problems

Xiao-Chuan Cai*
William D. Gropp†
David E. Keyes‡

Abstract. In recent years, competitive domain-decomposed preconditioned iterative techniques have been developed for nonsymmetric elliptic problems. In these techniques, a large problem is divided into many smaller problems whose requirements for coordination can be controlled to allow effective solution on parallel machines. Central questions are how to choose these small problems and how to arrange the order of their solution. Different specifications of decomposition and solution order lead to a plethora of algorithms possessing complementary advantages and disadvantages. In this report we compare several methods, including the additive Schwarz algorithm [3, 7, 10], the multiplicative Schwarz algorithm [2, 8], the tile algorithm [13], the CGK [5] and CSPD [19] algorithms, and the popular global ILU-family of preconditioners, on some nonsymmetric and/or indefinite elliptic model problems discretized by finite difference methods. The preconditioned problems are solved by the unrestarted GMRES method.

1. Introduction. The focus of this paper is domain decomposition methods for the solution of large linear systems of nonsymmetric and/or indefinite elliptic finite difference equations. In the past five years, there has been gratifying progress in the development of domain decomposition algorithms for symmetric elliptic problems, and a number of rapidly converging methods have been designed for which the condition number of the iteration matrix is uniformly bounded or grows only in proportion to a power of $(1+\ln(H/h))$, where H is the diameter of a typical subdomain and h is the diameter of a typical element into which the subdomains are divided. Such algorithms are often called "optimal" or "nearly optimal"

* Department of Mathematics, University of Kentucky, Lexington, KY 40506. *cai@ms.uky.edu.* The work of this author was supported in part by the Center for Computational Science at the University of Kentucky.

† Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, *gropp@mcs.anl.gov.* The work of this author was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

‡ Dept. of Mechanical Engineering, Yale University, New Haven, CT 06520, *keyes@cs.yale.edu.* The work of this author was supported in part by the NSF under contract ECS-8957475, by the National Aeronautics and Space Administration under NASA Contract NAS1-18605 while in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center Hampton, VA 23665, and by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, under Contract W-31-109-Eng-38 while in residence at Argonne National Laboratory, Argonne, IL 60439.

algorithms, respectively, though we note that these adjectives pertain to the convergence rate only, and not to the overall computational complexity. The computational complexity of "nearly optimal" algorithms may still retain terms that are polynomial in $1/H$ or in H/h, depending upon how the component problems are solved. For nonsymmetric and/or indefinite problems, the theory to date is far less satisfactory. Yet, the solution of such problems is an important computational kernel in implicit methods (for instance, Newton-like methods) used in the solution of nonlinear partial differential equations such as arise in computational fluid dynamics. Such a kernel is often CPU-bound or memory-bound, or both on the fastest and largest computers available. Furthermore, it may often be the only computationally intensive part of production finite difference codes whose efficient parallelization is not straightforward, particularly when the distribution of data throughout the computer's memory hierarchy cannot be dictated exclusively by linear algebra considerations. If the parallel solution of nonsymmetric and indefinite problems were truly routine, many applications now customarily solved by various types of operator splitting could be handled fully implicitly.

An efficient iterative algorithm for elliptic equations requires a discretization scheme, a basic iterative method, and a preconditioning strategy. There is a significant difference between symmetric and nonsymmetric problems, the latter being considerably harder to deal with both theoretically and algorithmically. The main reasons are the lack of a generally applicable discretization technique for the general nonsymmetric elliptic operator, the lack of "good" algebraic iterative methods (such as CG for symmetric, positive definite problems), and the incompleteness of the mathematical theory for the performance of the algebraic iterative methods that do exist (such as GMRES [18]). By a "good" method, we mean a method that is provably convergent within memory requirements proportional to a small multiple of the number of degrees of freedom in the system, independent of the operator. Though GMRES seems as popular as any nonsymmetric solver, one has to assume that the symmetric part is positive definite and be able to afford amounts of memory roughly in proportion to the number of iterations, in order to obtain rapid convergence. The task of finding a good preconditioner for nonsymmetric or indefinite problems is more important than for symmetric, positive definite problems, since, first, the preconditioner can force the symmetric part of the preconditioned system to be positive definite, and second, a better-conditioned system implies both more rapid convergence and smaller memory requirements.

Domain decomposition methods are commonly classified according to a few orthogonal criteria. "Overlapping" and "nonoverlapping" methods are differentiated by the decomposition into territories on which the elemental subproblems are defined. Overlapping methods generally permit simple (Dirichlet) updating of the boundary data of the subregions at the expense of extra arithmetic complexity per iteration from the redundantly defined degrees of freedom. "Additive" (Jacobi-like) or "multiplicative" (Gauss-Seidel-like) methods are differentiated by the interdependence of the subregions within each iteration. For the same number of subregions, additive methods are intrinsically more parallelizable. Classified according to convergence rate, there are "optimal" algorithms, for which the rate is independent of the number of unknowns as well as the number of subregions; "nearly optimal" algorithms, for which the rate depends on the number of unknowns and subregions through a power of logarithm at worst; and "nonoptimal" algorithms. Compared in this paper are optimal overlapping algorithms, both additive and multiplicative; a nearly optimal nonoverlapping algorithm, partly additive-partly multiplicative; and a nonoptimal nonoverlapping multiplicative algorithm.

Most of the theory concerning the convergence rate of domain decomposition methods is in the framework of the Galerkin finite element method. In some cases the Galerkin results transfer immediately to finite difference discretizations, though from our experimental experience this is less true for nonsymmetric problems than for symmetric. Whereas experimental papers for symmetric problems, such as [12] and [15], predominantly played the role of *verifying* theory, in this paper we hope to *stimulate* it.

The outline of this paper is as follows. In Section 2, we describe five domain decomposition methods. Because of space limitations we are unnaturally brief, relegating detailed algorithmic descriptions and existing theory to the references and to a subsequent expanded version. Section 3 contains the numerical results for four different test problems, followed by some brief conclusions in Section 4.

2. Description of Algorithms.

In this section, we briefly describe all the algorithms under consideration. We give only the formulation used in our experiments but note here that each is representative of a class. For theoretical purposes most of these algorithms are best formulated in terms of the subspace projections defined by the elliptic bilinear forms. Since we use only finite difference discretization here, matrix notation is more convenient.

2.1. A Two-Level Discretization and Notation.

Let Ω be a two-dimensional polygonal region with boundary $\partial\Omega$, and $\mathcal{L}$ a second-order linear elliptic operator corresponding to a homogeneous Dirichlet boundary value problem on Ω. The finite difference approximation of this Dirichlet problem is denoted by

$$B_h \mathbf{u}_h = (A_h + N_h)\mathbf{u}_h = f_h, \tag{1}$$

where B_h, A_h, and N_h are $n \times n$ matrices and h characterizes the mesh interval of the grid, which will be referred to as the h-level or fine grid. Here A_h represents the discretization of the symmetric, positive definite part of the operator $\mathcal{L}$, and N_h represents the remainder. The total number of interior nodes of the h-level grid of Ω is denoted as n. Two finite difference discretizations are employed alternately, namely, the central and upwind discretizations. In practice, multiple discretizations can be usefully combined in the same iterative process; see, for instance, [16].

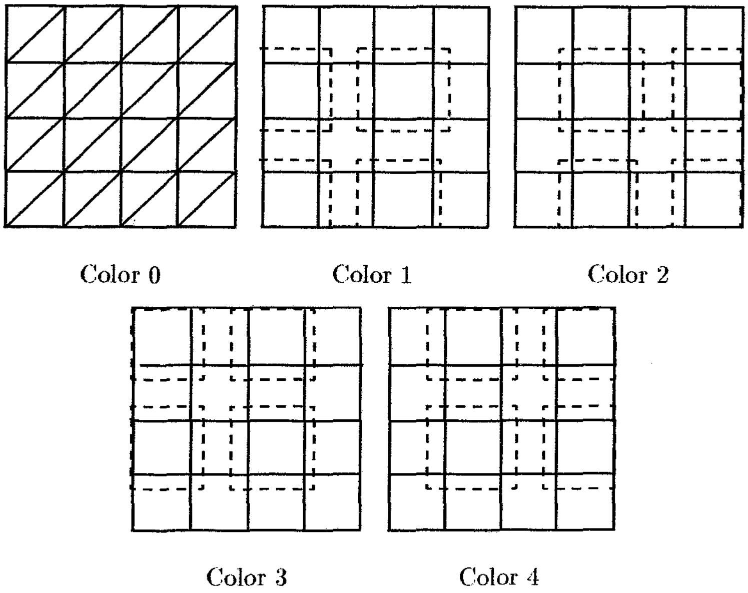
Our methods require a coarse grid over Ω, referred to as the H-level grid and containing n_0 interior nodes. $B_{h,0}$ is an $n_0 \times n_0$ matrix representing the finite difference discretization of $\mathcal{L}$ on the H-level grid. Let $\Omega_i, i = 1, \cdots, N$, be nonoverlapping subregions of Ω with diameters of order H, such that $\bigcup \bar{\Omega}_i = \bar{\Omega}$ and the nonboundary vertices of any Ω_i coincide with the H-level nodes. We refer to $\{\Omega_i, i = 1, \cdots, N\}$ as a nonoverlapping decomposition or substructuring of Ω.

When the unknowns are ordered with respect to the substructuring of the region, the stiffness matrix B_h can be written in the block form

$$B_h = \begin{pmatrix} B_{II} & B_{IE} & B_{IC} \\ B_{EI} & B_{EE} & B_{EC} \\ B_{CI} & B_{CE} & B_{CC} \end{pmatrix}, \tag{2}$$

where B_{II} is a block diagonal matrix representing the discretization of the independent subregion interior problems, B_{EE} corresponds to the problems on the edges (also called interfaces) excluding crosspoints, and B_{CC} corresponds to the crosspoints. The block matrices with differing subscripts contain the h-scale coupling of the original discretization between points in the different sets.

Following [7, 10], we can obtain an overlapping decomposition of Ω, denoted as $\{\Omega_i', i = 1, \cdots, N\}$, by extending each Ω_i to a larger region Ω_i', which is, however, cut off at the physical boundary of Ω. Let n_i be the total number of h-level interior nodes in Ω_i', and let



Color 0 Color 1 Color 2

Color 3 Color 4

FIG. 1. *The coloring pattern of 16 fine grid overlapped subregions and a coarse grid region. Color "0" is for the global coarse grid. The extended subregions of the other colors are indicated by the dotted boundaries.*

$B'_{h,i}$ denote the $n'_i \times n'_i$ stiffness matrix corresponding to the finite difference discretization of $\mathcal{L}$ on the fine grid in Ω'_i. The size of the matrix $B'_{h,i}$ depends not only on the size of the substructure Ω_i but also on the degree of overlap. We reserve the subscript "o" for the global coarse grid and note that $\Omega'_0 = \Omega$. Let R'_i be an $n'_i \times n$ matrix representing the algebraic restriction of an n-vector on Ω to the n'_i-vector on Ω'_i. Thus, if v_h is a vector corresponding to the h-level interior nodes in Ω, then $R'_i v_h$ is a vector corresponding to the h-level interior nodes in Ω'_i. The transpose $(R'_i)^t$ is an extension-by-zero matrix, which extends a length n_i vector to a length n vector by padding with zero. $R'_0(\equiv R_0)$, an $n_0 \times n$ matrix, is somewhat special. It is the fine-to-coarse grid restriction operator that is needed in any multigrid method.

2.2. Multiplicative Schwarz Methods (MSM). The original Schwarz alternating method is a purely sequential algorithm. To obtain parallelism, one needs a good subdomain coloring strategy, so that a set of independent subproblems can be introduced within each sequential step and the total number of sequential steps can be minimized. A detailed description of the algorithm can be found in [2, 8, 17].

The coloring is realized as follows. We define an undirected graph, associated with the decomposition $\{\Omega'_j\}$, in which nodes represent the extended subregions and the edges intersections of the extended subregions. This graph can be colored with colors $0, \cdots, J$, such that no connected nodes have the same color. For example, a five-color strategy ($J = 4$) is used as shown in Figure 1. This is optimal for this special case, in which the total number of subregions (including the coarse grid on the global region) is $N + 1 = 17$. It is obvious that the colorings are not unique. Numerical experiments confirm intuition that minimizing the number of colors enhances convergence.

Letting $B'_{h,0} = B_{h,0}$ and $R'_0 = R_0$, we describe the algorithm in terms of a subspace correction process. If u_h^n is the current approximate solution, then u_h^{n+1} is computed as follows. For $j = 0, 1, \cdots, J$:

(i) Compute the residual in subregions with the j^{th} color:

$$r_h^{n+\frac{j}{J+1}} = f_h - B_h u_h^{n+\frac{j}{J+1}}$$

(ii) Solve for the subspace correction in all $\Omega_i^{'}$s that share the j^{th} color:

$$B_{h,i}^{'} e_h^{n+\frac{j}{J+1}} = R_i^{'} r_h^{n+\frac{j}{J+1}}$$

(iii) Update the approximate solution in all $\Omega_i^{'}$s that share the j^{th} color:

$$u_h^{n+\frac{j+1}{J+1}} = u_h^{n+\frac{j}{J+1}} + (R_i^{'})^t e_h^{n+\frac{j}{J+1}}$$

At each iteration, every subproblem is solved once. For $j \neq 0$, applications of operators $R_j^{'}$ and $(R_j^{'})^t$ do not involve any arithmetic operations. For $j \neq 0$, within each (i)-(iii), the operations in regions sharing the same color can be done in parallel.

This algorithm can be employed in the stationary, Richardson sense or as a preconditioner for another algebraic iterative process. Along with the other algorithms to be described below, we shall normally employ it as a preconditioner for GMRES, but because of its historical importance we also include the Richardson version in our tests. In this paper, we shall use the abbreviation MSM for the multiplicative Schwarz-preconditioned GMRES method, and MSR for the simple Richardson process.

2.3. Additive Schwarz Method (ASM). Following [7] and using the notation of the preceding subsection, we can define the inverse of the matrix M_h, the additive Schwarz preconditioner, as

$$M_h^{-1} = (R_0)^t(B_{h,0})^{-1}R_0 + (R_1^{'})^t(B_{h,1}^{'})^{-1}R_1^{'} + \cdots + (R_N^{'})^t(B_{h,N}^{'})^{-1}R_N^{'}. \tag{3}$$

The key ingredients for the success of the ASM are the use of overlapping subregions and the incorporation of a coarse-grid solver. At each iteration, all subproblems are solved once. It is obvious that all subproblems are independent of each other and can therefore be solved in parallel.

2.4. Coarse Grid Plus SPD Preconditioning (CSPD). For a symmetric, positive definite elliptic problem, many good preconditioners are available. Supplemented by an additional coarse-mesh preconditioner, they may become good, sometimes optimal, preconditioners for nonsymmetric problems, as shown in [19]. More precisely, let $(\tilde{A}_h)^{-1}$ be a spectrally equivalent symmetric, positive definite preconditioner for A_h, which is in turn the symmetric, positive definite part of B_h. Then the new preconditioner can be written as

$$M_h^{-1} = \omega(R_0)^t(B_{h,0})^{-1}R_0 + (\tilde{A}_h)^{-1}, \tag{4}$$

where $\omega > 0$ is a balancing parameter. In this paper, the symmetric, positive definite preconditioner $(\tilde{A}_h)^{-1}$ is taken as the symmetric, positive definite additive Schwarz preconditioner. For $i = 0, \cdots, N$, we denote by $A_{h,i}^{'}$ an $n_i^{'} \times n_i^{'}$ matrix that corresponding to the discretization of the second-order terms of $\mathcal{L}$ in $\Omega_i^{'}$ with zero boundary condition. Then, we have

$$(\tilde{A}_h)^{-1} = (R_0)^t(A_{h,0})^{-1}R_0 + (R_1^{'})^t(A_{h,1}^{'})^{-1}R_1^{'} + \cdots + (R_N^{'})^t(A_{h,N}^{'})^{-1}R_N^{'}. \tag{5}$$

To obtain the optimal ω, one needs to know, in some sense, how good the preconditioner $(\tilde{A}_h)^{-1}$ is. In our numerical experiments, using (5) in (4) as a preconditioner for A_h, the choice of $\omega = 1.0$ is quite acceptable in comparison with other methods. The issue of finding the optimal ω in the general case is not yet understood.

2.5. Tile Algorithms (GK90/GK91). The "tile" preconditioner, introduced in [13], is a nonoverlapping multiplicative-type preconditioner formed as follows:

$$(6) \qquad M_h^{-1} = \begin{pmatrix} B_{II} & B_{IE} & B_{IC} \\ & T_{EE} & B_{EC} \\ & & B_{h,0} \end{pmatrix}^{-1} \begin{pmatrix} I & & \\ & I & \\ & Q & L \end{pmatrix},$$

where the matrix T_{EE} is the so-called tangential interface preconditioner, a block diagonal matrix in which each block corresponds to the interface between a pair of neighboring subdomains. The coefficients of each block can be obtained by the usual three-point discretization of the "tangential part" of the underlying differential operator, that is, the set of terms that remain when the operator is expressed in terms of local tangential and normal derivatives and all normal derivatives are then set to zero. The matrices Q and L define a so-called ramp-weighted averaging method used to calculate the right-hand side of the coarse-mesh problem. L is diagonal with all positive elements, all elements of Q are nonnegative, and the row sums of L and Q together give unity. A fuller description can be found in [13].

To perform the matrix-vector multiply, three sequential steps are needed: solution of a coarse-mesh problem with a locally-averaged right-hand side; solution of the interface problems with right-hand sides updated by the boundary values provided by the coarse-grid solution; and solution of the interior problems with right-hand sides updated by the boundary values provided by the coarse-grid and interface solutions. Note that the second and third steps are composed of completely independent subtasks on each interface and subdomain.

A more recent tile algorithm (GK91) incorporates two refinements. The right-hand side of each interface problems is modified prior to their solution using T_{EE} to include an approximation to the non-tangential terms of B_{EE}. These terms are formed from bivariate interpolation of the coarse-mesh solution, quadratic normal to the interface and linear tangential to it. The other modification is a fine-mesh correction to the crosspoint values, which occurs as the last step of the preconditioner. This correction is based on the original bottom block row of the stiffness matrix (2). A detailed matrix interpretation is furnished in [6]. As is apparent from the tables, these additional sequential stages can have a substantial impact on the convergence rate of the tile algorithm without requiring more than one set of subdomain solves per iteration.

2.6. Substructuring Algorithm (CGK). A nearly optimal nonoverlapping-partly additive-partly multiplicative method, proposed in [5], can be formulated as

$$(7)\ M_h^{-1} \;=\; (R_0)^t B_{h,0}^{-1} R_0 + \begin{pmatrix} B_{II}^{-1} + B_{II}^{-1} B_{IE} K_{EE}^{-1} B_{EI} B_{II}^{-1} & -B_{II}^{-1} B_{IE} K_{EE}^{-1} & 0 \\ -K_{EE}^{-1} B_{EI} B_{II}^{-1} & K_{EE}^{-1} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where K_{EE} is a block diagonal matrix, each block corresponding to an interface. Each block in K_{EE} has the form of the square root of the one-dimensional Laplacian along the interface, with size equal to the number of interior interface nodes. For selfadjoint problems, this historically important interface preconditioner is spectrally equivalent to the Schur complement of the interface degrees of freedom in the submatrix corresponding to each pair of adjacent subregions; see the survey paper [15].

To form the action of $M_h^{-1} u_h$, one needs to solve a coarse-grid problem and, at the same time, solve sequentially three sets of subproblems: a first set of independent interior problems; a set of independent interface problems; and another set of interior problems.

3. Numerical Experiments. In this section, we present some numerical results obtained by applying the aforementioned algorithms to

$$\left\{ \begin{array}{rcll} \mathcal{L}\mathbf{u} & = & f & \text{in} \quad \Omega \\ \mathbf{u} & = & 0 & \text{on} \quad \partial\Omega, \end{array} \right. \tag{8}$$

where different elliptic operators $\mathcal{L}$ will be specified and $\Omega = [0,1] \times [0,1]$. In all cases, the exact solution $u = \epsilon^{xy} \sin(\pi x) \sin(\pi y)$, and f can thus be set accordingly.

The unit square is subdivided into two-level uniform meshes, with h and H representing the fine- and coarse-mesh sizes. The elliptic operator is discretized by the usual five-point central- or upwind-difference methods over both meshes. The full GMRES method, without restarting, is used for all of the left-preconditioned linear systems (except for MSR) in the usual Euclidean norm, and the stopping criterion is the reduction of the initial (preconditioned) residual by five orders of magnitude in this norm. Double precision is used in the entire computation.

3.1. The Poisson Equation. Our first test problem is the Poisson equation, $\mathcal{L}u = -\triangle u$. Although this is a symmetric problem, we still use GMRES as the outer iterative method. For symmetric, positive definite problems, the iteration matrices of ASM and CGK are symmetric and positive; and therefore, with a suitable inner product, CG is more efficient. The iteration counts are given in Table 1. (Entries that would have required overlap greater than the coarse mesh size H are omitted.) Among all algorithms, the MSM with a reasonable overlapping size takes the least number of iterations. Since MSR does not depend on an outer algebraic iterative method, it takes the least amount of computer memory.

With the minimal overlapping assumption (one fine-grid cell in each direction), the convergence of ASM and MSM are seen to be almost independent of h especially when H is fine enough. In contrast, h-independent convergence of MSR requires that the overlap grow in discrete size so that the same physical overlap is maintained. Thus for MSR, the number of iterations is constant along the diagonals of the blocks labeled $H = 1/4$, $H = 1/8$, and $H = 1/16$, as h and the discrete overlap successively double, whereas for MSM and ASM the iteration counts are nearly constant even along the rows of the $H = 1/8$ and $H = 1/16$ entries. This fact suggests that the "sufficiently large" overlapping hypothesis used in the Schwarz theory [7, 8, 10] can be weakened, at least for symmetric definite problems.

Because of additivity, which means less dependence between the subproblems of the coarse-mesh solve and the interface/interior solves, the CGK preconditioner is presumably weaker than that of [1]. However, the numerical results, at least for the present test problem, do not reveal any weakness. In fact, the additivity offers more parallelism.

3.2. A Nonsymmetric Problem. Our second test problem is nonsymmetric with constant coefficients, $\mathcal{L}u = -\triangle u + \delta u_x + \delta u_y$. We specify the constant $\delta > 0$ in the Table 2. The elliptic operator is discretized by two schemes, namely, the central-difference method for relatively small δ and the upwind-difference method for relatively large δ.

When using the central-difference method, for a fixed fine-mesh size h, we observe that as δ is increased beyond a certain number (near 10), all methods, except MSM with sufficient overlap, show a sharp upturn in the number of iterations. MSR loses its convergence if δ is larger than this transitional δ for essentially all overlapping sizes. All GMRES-stabilized methods continue to converge but at a reduced rate, especially the nonoverlapping methods. The nonoverlapping methods have difficulty handling large convection terms. In comparison, MSM converges in a surprisingly small number of steps that, with generous overlap, is almost independent of δ.

Central-Difference Method												
$\delta =$	1	5	10	50	100	150	1	5	10	50	100	150
Methods	$H = 1/4$						$H = 1/8$					
MSR(ovlp=h)	19	18	15	∞	∞	∞	10	10	10	13	∞	∞
MSR(ovlp=$2h$)	12	11	9	21	∞	∞	7	7	7	14	∞	∞
MSR(ovlp=$4h$)	7	8	8	22	∞	∞	6	6	6	10	35	∞
MSR(ovlp=$8h$)	6	7	8	24	∞	∞	5	5	5	7	10	21
MSM(ovlp=h)	7	7	7	10	10	9	5	5	5	8	10	12
MSM(ovlp=$2h$)	6	6	6	8	8	8	4	4	4	7	8	11
MSM(ovlp=$4h$)	5	5	6	7	7	7	4	4	4	5	7	9
MSM(ovlp=$8h$)	5	5	5	6	6	6	4	4	4	4	5	7
ASM(ovlp=h)	15	17	18	22	22	21	11	12	12	20	26	32
ASM(ovlp=$2h$)	13	15	15	20	20	21	10	10	11	18	23	27
ASM(ovlp=$4h$)	12	13	13	18	19	20	10	11	11	15	20	23
ASM(ovlp=$8h$)	11	12	12	16	17	17	10	11	12	14	16	19
GK90	25	25	26	35	39	42	19	21	22	34	47	58
GK91	20	23	25	26	22	18	14	16	19	39	42	45
CGK	13	14	16	28	35	47	11	12	13	26	36	50
CSPD($\omega = 1.0$)	11	13	15	37	57	73	10	12	13	26	42	55
	$H = 1$											
ILU(0)	85	88	84	59	41	28						
ILU(1)	36	32	27	15	8	5						
ILU(2)	56	56	52	35	23	15						

Upwind-Difference Method												
$\delta =$	10	50	100	500	1000	10000	10	50	100	500	1000	10000
Methods	$H = 1/4$						$H = 1/8$					
MSR(ovlp=h)	18	14	13	18	18	18	10	13	14	21	23	23
MSR(ovlp=$2h$)	14	14	15	16	16	16	10	13	16	17	16	15
MSR(ovlp=$4h$)	12	13	14	13	12	12	9	12	12	12	12	12
MSR(ovlp=$8h$)	10	10	10	11	11	11	8	9	9	8	8	8
MSM(ovlp=h)	9	9	8	7	7	7	7	9	9	10	11	11
MSM(ovlp=$2h$)	8	8	7	7	7	7	7	8	8	9	9	9
MSM(ovlp=$4h$)	7	7	6	6	6	6	7	7	6	6	6	6
MSM(ovlp=$8h$)	5	5	5	5	5	5	5	5	5	5	5	6
ASM(ovlp=h)	19	20	19	18	17	17	14	19	21	22	22	23
ASM(ovlp=$2h$)	17	18	16	16	17	17	14	17	19	19	20	19
ASM(ovlp=$4h$)	15	16	16	16	16	16	14	15	16	17	17	18
ASM(ovlp=$8h$)	13	14	14	14	14	14	13	14	15	15	16	16
GK90	28	31	33	40	42	45	22	27	30	37	39	42
GK91	28	23	19	12	10	8	22	24	20	16	15	15
CGK	17	22	25	41	47	49	16	23	38	41	50	60
	$H = 1$											
ILU(0)	85	62	51	23	16	6						
ILU(1)	27	17	11	5	4	2						
ILU(2)	52	37	29	13	9	4						

TABLE 2

Iteration counts for solving the nonsymmetric model equation. The fine mesh size is uniformly $1/h = 128$.

$h =$	1/32	1/64	1/128	1/32	1/64	1/128	1/64	1/128
Methods	$H = 1/4$			$H = 1/8$			$H = 1/16$	
MSR(ovlp=h)	7	11	19	6	7	10	5	6
MSR(ovlp=$2h$)	6	7	11	5	6	7	4	5
MSR(ovlp=$4h$)	5	6	7		5	6		4
MSR(ovlp=$8h$)		5	6			5		
MSM(ovlp=h)	5	6	7	4	4	5	3	3
MSM(ovlp=$2h$)	5	5	6	4	4	4	3	3
MSM(ovlp=$4h$)	4	5	5		4	4		3
MSM(ovlp=$8h$)		4	5			4		
ASM(ovlp=h)	11	13	15	10	10	11	9	8
ASM(ovlp=$2h$)	11	11	13	10	10	10	8	8
ASM(ovlp=$4h$)	10	11	11		10	10		8
ASM(ovlp=$8h$)		10	11			10		
CGK	12	13	13	12	11	11	10	10

TABLE 1
Iteration counts for solving the Poisson equation.

The complexion changes when we switch to the upwind-difference method. The iteration counts for the overlapping methods remain nearly constant even for large δ. It is seen that with modest overlap, two fine-mesh widths in the test problem, the iteration counts are independent of δ for MSR, MSM and ASM. For the nonoverlapping methods, the iteration counts continue to grow significantly as the constant δ increases.

From these results, there is a strong connection between the stability of the discretization scheme and the convergence rate of the domain decomposition methods. The current Galerkin finite element-based domain decomposition theory for nonsymmetric problems predicts very well the behavior of algorithms with central-difference discretizations; for example, a finer coarse mesh leads to more rapid convergence. However, the situation is different with upwind differencing; refining the coarse mesh may not always reduce the number of iterations.

For the range of problems considered, MSM and ASM are the most robust schemes and behave well in all cases. Unaccelerated MSR is too sensitive to the stability of the discretization. However, if used with a stable discretization scheme, it not only provides a competitive convergence rate, but does so while occupying the least computer memory.

The nonoverlapping methods do not behave well if the constant δ is large with either discretization scheme. We believe that this behavior is traceable to the interface preconditioners. It should be noted that the flow direction has purposely been selected to confound the interface preconditioners in this example, being skew to every interface. Alignment of (half of) the interfaces with a predominant flow direction, which can often be arranged in practice, is seen in [9] to lead to improved convergence rates for nonoverlapping methods. The interface preconditioner employed in CGK makes no adaptation to the presence of the convection terms regardless of their alignment.

The ILU results are rather interesting. For this particular constant-coefficient test problem on a fine grid of 128^2 cells, global ILU methods do not perform well for the small-convection problems that yield rapidly to domain-decomposed methods, but they work very well for large-convection problems, especially ILU(1). ILU is sensitive to the signs and magnitudes of the coefficients of the nonsymmetric terms, as well as the discretization parameter h. Some analysis was given in [11]. The central-difference ILU results begin to

$\sigma =$	0	30	70	110	150	300	0	30	70	110	150	300
Methods	$H = 1/8$						$H = 1/16$					
MSR(ovlp=h)	10	11	20	∞	∞	∞	6	6	6	7	21	∞
MSR(ovlp=$2h$)	7	7	16	∞	∞	∞	5	5	5	5	14	∞
MSR(ovlp=$4h$)	6	6	14	∞	∞	∞	4	4	5	5	12	∞
MSM(ovlp=h)	5	5	7	9	13	35	3	4	4	4	6	8
MSM(ovlp=$2h$)	4	4	6	8	12	37	3	3	4	4	6	9
MSM(ovlp=$4h$)	4	4	5	8	13	> 100	3	3	4	4	6	9
ASM(ovlp=h)	11	12	14	19	23	62	8	9	9	10	11	16
ASM(ovlp=$2h$)	10	10	14	18	23	61	8	8	9	10	10	16
ASM(ovlp=$4h$)	10	10	13	15	22	78	8	9	10	10	12	17
CGK	11	13	18	25	31	80	10	10	12	14	16	23
CSPD	9	13	14	17	32	49	8	11	11	13	21	33

TABLE 3

Iteration counts for solving the Helmholtz equation. The fine-mesh size is uniformly $1/h = 128$.

deteriorate once the cell Peclet number, $\delta \cdot h$, exceeds 2 (which lies beyond the range of the upper part of Table 2). ILU results for a variable-coefficient test problem are given in Table 4.

3.3. The Helmholtz Equation. Our third test problem is a Helmholtz equation with constant coefficients, $\mathcal{L}u = - \triangle u - \sigma u$. It is selfadjoint, but indefinite. The eigenvalues of this equation are $(i^2 + j^2)\pi^2 - \sigma$, where i and j are positive integers. We choose σ so as to avoid putting any eigenvalue in a small neighborhood of zero, but there may be several eigenvalues of both signs.

For slightly indefinite (small σ) problems, it is shown that in Table 3 that all methods are similar to the case when $\sigma = 0$. However, as σ increases with the grid held fixed, the iteration counts grow rapidly. A finer coarse mesh (more coarse-mesh points per wavelength) is needed to counteract high wavenumber.

With a sufficiently fine coarse mesh, the MSM is seen to be the most rapidly converging among all methods. However, theoretical specification of a sufficiently fine coarse mesh size is not (in general) easy, and our current method is simply to try a few different H's. The "sufficiently fine" hypothesis is seen to be extremely important for MSM in the two entries with $\sigma = 300$ with an overlap of $4h$: with $H = 1/8$ more than 100 iterations are required for convergence, while 9 suffice with $H = 1/16$. Curiously, increasing overlap seems to degrade convergence in the strongly indefinite case, whereas it always improves the convergence of definite operators. For instance, when $H = 1/8$ and $\sigma = 300$, overlaps of h, $2h$, $3h$ (not listed in Table 3), and $4h$ lead to iteration counts of 35, 37, 43, and > 100, respectively. Loss of orthogonality likely plays a contributing role in the upturn.

ILU preconditioners cannot generally be defined for discrete operators as indefinite as these and do not appear in Table 3.

3.4. A Variable-Coefficient, Nonsymmetric Indefinite Problem. Our last test problem has variable (oscillatory) coefficients and is nonsymmetric and indefinite:

$$\begin{aligned}\mathcal{L}u &= -((1 + \tfrac{1}{2}\sin(50\pi x)u_x)_x - ((1 + \tfrac{1}{2}\sin(50\pi x)\sin(50\pi y)u_y)_y \\ &\quad +20\sin(10\pi x)\cos(10\pi y)u_x - 20\cos(10\pi x)\sin(10\pi y)u_y - 70u.\end{aligned}$$

The diffusion coefficients oscillate but do not vary in sign. The coefficients of the first-order terms represent a ten-by-ten array of closed convection cells, with no convective transport

$h =$	1/32	1/64	1/128	1/32	1/64	1/128	1/64	1/128
Methods	$H = 1/4$			$H = 1/8$			$H = 1/16$	
MSR	∞	∞	∞	∞	∞	∞	∞	∞
MSM	15	15	14	16	15	15	10	10
ASM	33	35	35	29	26	25	19	18
CSPD	35	30	30	33	30	29	21	19
GK90	36	41	50	52	60	69	29	38
GK91	21	23	31	27	35	50	21	34
CGK	38	37	37	33	30	33	22	24
	H=1							
ILU(0)	41	75	266					
ILU(1)	86	69	161					
ILU(2)	25	41	92					

TABLE 4
Iteration counts for solving the variable-coefficient, nonsymmetric indefinite problem.

between cells. However, the subdomain boundaries do not in general align with the convection cell boundaries, so this property is not exploited. The operator $\mathcal{L}$ is discretized by the five-point central-difference method. A fixed overlapping factor of 25% in both x and y directions is employed in all overlapping methods. That is to say, the distance between the boundaries of an extended subregion to the subregion is 25% (or greater, in corners) of the diameter of the subregion, which is approximately H.

This problem is difficult for all of the methods, but the iteration count for MSM is smaller than that of others by almost a factor of 2, or more. MSR diverges in all cases. For a fixed coarse mesh size H, some methods tend to require fewer iterations when the fine mesh is refined; others require more. We believe that this behavior is related to oscillatory coefficients in the second-order terms of $\mathcal{L}$. The discretization becomes more stable when h gets smaller relative to the oscillatory coefficients.

The nonoverlapping method CGK, which includes an interface preconditioner based solely on the diffusive terms of $\mathcal{L}$, behaves reasonably well, probably because the magnitude of the convection is not large and averages to zero over the domain.

For this variable-coefficient problem, the ILU preconditioners are outperformed by the domain decomposition-preconditioned methods.

4. Concluding Remarks. We have implemented and tested five domain decomposition methods recently proposed for nonsymmetric, indefinite PDEs. In applications, a number of parameters need to be selected for each algorithm, such as subregion geometry and granularity, extent of overlap, exactness of subproblem solves, and balancing parameter. The volume of parameter space renders a complete numerical comparison impractical. As permitted by space, we have highlighted a few comparisons we consider interesting and useful in suggesting theorems that may be provable.

Domain-decomposed preconditioners cannot be ranked in any uniform way. The performance of some of the algorithms depends strongly on the discretization scheme, as well as on the underlying differential operator. Even when attention is confined to a given problem, the effect of the discretization scheme and the outer iterative method should be considered along with the preconditioning strategy.

Finally, we note that attention has been confined in this paper to iteration count, as opposed to overall computational complexity or execution time. Indeed, though the codes share the same basic data structure, they have received varying degrees of attention to execution optimization. However, an extension of this work now in preparation addresses complexity issues and parallelism in detail.

REFERENCES

[1] J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The Construction of Preconditioners for Elliptic Problems by Substructuring, I*, Math. Comp., 47(1986), pp. 103–134.

[2] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence Estimates for Product Iterative Methods with Applications to Domain Decomposition and Multigrid*, 1990, Cornell University.

[3] X.- C. Cai, *Some Domain Decomposition Algorithms for Nonselfadjoint Elliptic and Parabolic Partial Differential Equations*, Ph.D. thesis, Courant Institute, Sept. 1989.

[4] X.- C. Cai, *An Additive Schwarz Algorithm for Nonselfadjoint Elliptic Equations*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, 1990, pp. 232-244.

[5] X.- C. Cai, W. D. Gropp, and D. E. Keyes, *Convergence Rate Estimate for a Domain Decomposition Method*, Technical Report 827, Department of Computer Science, Yale University, Oct. 1990, (to appear in Numer. Math.).

[6] X.- C. Cai, W. D. Gropp, and D. E. Keyes, *A Comparison of Some Domain Decomposition and ILU Preconditioned Iterative Methods for Nonsymmetric Elliptic Problems* (in preparation).

[7] X.- C. Cai and O. B. Widlund, *Domain Decomposition Algorithms for Indefinite Elliptic Problems*, SIAM J. Sci. Stat. Comp. 13 (1992).

[8] X.- C. Cai and O. B. Widlund, *Multiplicative Schwarz Algorithms for Nonsymmetric Elliptic and Parabolic Problems*, TR CCS-90-7, Center for Computational Science, University of Kentucky, Oct. 1990.

[9] T. F. Chan and D. E. Keyes, *Interface Preconditionings for Domain-Decomposed Convection-Diffusion Operators*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, 1990, pp. 245–262.

[10] M. Dryja and O. B. Widlund, *Towards a Unified Theory of Domain Decomposition Algorithms for Elliptic Problems*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, 1990, pp. 3–21.

[11] H. C. Elman, *A Stability Analysis of Incomplete LU Factorizations*, Math. Comp., 47 (1986), pp. 191–217.

[12] A. Greenbaum, C. Li and Z. Han, *Parallelizing Preconditioned Conjugate Gradient Algorithms*, Comp. Phys. Comm., 53(1989), pp. 295–309.

[13] W. D. Gropp and D. E. Keyes, *Domain Decomposition with Local Mesh Refinement*, 1990 (to appear in SIAM J. Sci. Stat. Comp.).

[14] W. D. Gropp and D. E. Keyes, *Parallel Performance of Domain-Decomposed Preconditioned Krylov Methods for PDEs with Locally Uniform Refinement*, SIAM J. Sci. Stat. Comp., 13 (1992).

[15] D. E. Keyes and W. D. Gropp, *A Comparison of Domain Decomposition Techniques for Elliptic Partial Differential Equations and Their Parallel Implementation*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s166–s202.

[16] D. E. Keyes and W. D. Gropp, *Domain-Decomposable Preconditioners for Second-Order Upwind Discretizations of Multicomponent Systems*, in R. Glowinski, Y. A. Kuznetsov, G. A. Meurant, J. Périaux, and O. B. Widlund, eds., Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1991, pp. 129–139.

[17] P. L. Lions, *On the Schwarz Alternating Method. I*, in R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988, pp. 1–42.

[18] Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 865–869.

[19] J. Xu and X. - C. Cai, *A Preconditioned GMRES Method for Nonsymmetric or Indefinite Problems*, 1990 (to appear in Math. Comp.).