A COMPARISON OF SOME DOMAIN DECOMPOSITION AND ILU PRECONDITIONED ITERATIVE METHODS FOR NONSYMMETRIC ELLIPTIC PROBLEMS

XIAO-CHUAN CAI
Department of Mathematics, University of Kentucky
Lexington, KY 40506, USA

WILLIAM D. GROPP
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439, USA

DAVID E. KEYES
Department of Mechanical Engineering, Yale University
New Haven, CT 06520, USA

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. A central question is 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, the classical multiplicative Schwarz algorithm, an accelerated multiplicative Schwarz algorithm, the tile algorithm, the CGK algorithm, the CSPD algorithm, and also the popular global ILU-family of preconditioners, on some nonsymmetric or indefinite elliptic model problems discretized by finite difference methods. The preconditioned problems are solved by the unrestarted GMRES method. A version of the accelerated multiplicative Schwarz method is a consistently good performer.

Keywords: domain decomposition, preconditioning, iterative methods, nonsymmetric, indefinite, elliptic problems.

1. Introduction

The focus of this paper is domain decomposition methods for the solution of large linear systems of nonsymmetric 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 fast 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" algorithms, respectively, though we note that these adjectives pertain to the convergence rate only, and not to the overall computational complexity. The nearly optimal algorithms may still retain terms that are superlinear in $1/H$ or in H/h, depending upon how the component problems are solved. For nonsymmetric and 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 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 [32]. 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. One must 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 with GMRES. 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 their 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 this is less true for nonsymmetric problems than for symmetric. Whereas experimental papers for symmetric problems, such as [20] and [25], predominantly played the role of *verifying* theory, in this paper we hope to *stimulate* it.

Algorithms based on preconditioned iterative solution of the normal equations are beyond the scope of this paper, though they continue to undergo development [3, 4, 28, 31].

The outline of this paper is as follows. In Section 2, we describe five domain decomposition methods, their convergence properties, and related implementation issues. Some issues related to parallelism and parallel complexity are discussed in Section 3. Section 4 contains numerical results for four different test problems, followed by some brief conclusions in Section 5.

An early version of this work has appeared in an abridged proceedings form [8]. This paper completes and supersedes the earlier version.

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, matrix notation is more convenient, instead. The convergence rate of each algorithm is given in terms of the spectral bounds of the iteration matrix. These bounds may be related to the number of iterations required by each algorithm to achieve a given accuracy.

2.1. An elliptic problem, a two-level discretization, and notation

Let Ω be a two-dimensional polygonal region with boundary $\partial\Omega$, and let

$$\begin{cases} \mathcal{L}\mathbf{u} &= f \quad \text{in} \quad \Omega \\ \mathbf{u} &= 0 \quad \text{on} \quad \partial\Omega \end{cases} \tag{2.1}$$

be a second-order linear elliptic operator with a homogeneous Dirichlet boundary condition. The finite difference approximation of Dirichlet problem (2.1) is denoted by

$$B_h \mathbf{u}_h = (A_h + N_h)\mathbf{u}_h = f_h, \tag{2.2}$$

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. Let $(\cdot,\cdot)$ denote the Euclidean inner product with the corresponding norm $\|\cdot\|$. We denote the energy norm associated with the matrix A_h as

$$\|\cdot\|_A = (A_h\cdot,\cdot)^{1/2}.$$

The total number of interior nodes of the h-level grid of Ω will be denoted as n. Two finite difference discretizations are employed alternately for the first-order terms in (2.1), namely, the central and upwinding discretizations. In practice, there are many other discretization techniques such as the artificial diffusion and streamline diffusion methods [24] and the methods in [1]. Multiple discretizations can usefully be combined in the same iterative process; see, e.g., [26].

Our methods require a coarse grid over Ω containing n_0 interior nodes, or crosspoints, $\{c_k, k = 1, \cdots, n_0\}$; we call this the H-level grid. $B_{h,0}$ is an $n_0 \times n_0$ matrix representing the finite difference discretization of $\mathcal{L}$ on the

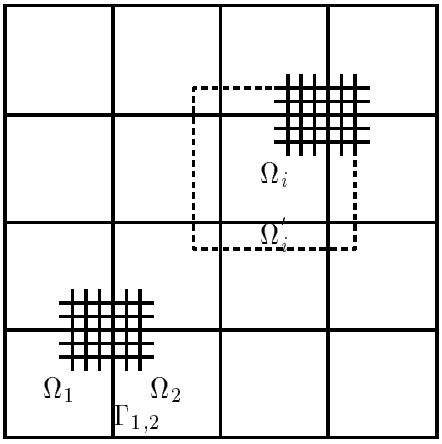

Fig. 1. The overlapping $\{\Omega_i'\}$ and nonoverlapping $\{\Omega_i\}$ decompositions of Ω

H-level grid. Let $\Omega_i, i = 1, \cdots, N$, be nonoverlapping subregions of Ω with diameters of order H, such that $\bigcup \bar{\Omega}_i = \bar{\Omega}$. The vertices of any Ω_i not on $\partial\Omega$ coincide with the h-level nodes. Let Γ_{ij} be the intersection of the adjacent pair of subdomains Ω_i and Ω_j excluding the end points. We refer to the set of subdomains

$$\{\Omega_i, \Gamma_{ij}, c_k, j \neq i = 1, \cdots, N, \, k = 1, \cdots, n_0\}$$

as a nonoverlapping decomposition, or substructuring, of Ω; see Figure 1.

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 as

$$B_h = \begin{pmatrix} B_{II} & B_{IE} & B_{IC} \\ B_{EI} & B_{EE} & B_{EC} \\ B_{CI} & B_{CE} & B_{CC} \end{pmatrix}, \tag{2.3}$$

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 [10, 15], we can obtain an overlapping decomposition of Ω, denoted by

$$\{\Omega_i', i = 0, \cdots, N\}.$$

For $i \neq 0$, we extend each Ω_i to a larger region $\Omega_i^{'}$ which is cut off at the physical boundary of Ω. Let $n_i^{'}$ be the total number of h-level interior nodes in $\Omega_i^{'}$, and let $B_{h,i}^{'}$ denote the $n_i^{'} \times n_i^{'}$ stiffness matrix corresponding to the finite difference discretization of $\mathcal{L}$ on the fine grid in $\Omega_i^{'}$. 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 "$_0$" 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 $\Omega_i^{'}$. 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 $\Omega_i^{'}$. 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. GMRES for the preconditioned system

The GMRES method, introduced in [32], is mathematically equivalent to the generalized conjugate residual (GCR) method [17] and can be used to solve the linear system of algebraic equations:

$$Px = b, \tag{2.4}$$

where P is a nonsingular matrix, which may be nonsymmetric or indefinite, and b is a given vector in R^n. The method begins with an initial approximate solution $x_0 \in R^n$ and an initial residual $r_0 = b - Px_0$. At the m^{th} iteration, a correction vector z_m is computed in the Krylov subspace

$$\mathcal{K}_m(r_0) \;=\; \mathrm{span}\{r_0, Pr_0, \cdots, P^{m-1} r_0\}$$

that minimizes the residual, $\min_{z \in \mathcal{K}_m(r_0)} |||b - P(x_0 + z)|||$ for some appropriate norm $|||\cdot|||$. The m^{th} iterate is thus $x_m = x_0 + z_m$. According to the theory of [17], the rate of convergence of the GMRES method can be estimated by the ratio of the minimal eigenvalue of the symmetric part of the operator to the norm of the operator. Those two quantities are defined by

$$c_P \;=\; \inf_{x \neq 0} \frac{[x, Px]}{[x, x]} \quad \text{and} \quad C_P \;=\; \sup_{x \neq 0} \frac{|||Px|||}{|||x|||},$$

where $[\cdot, \cdot]$ is an inner product on R^n that induces the norm $|||\cdot|||$. Following [17], the rate of convergence can be characterized, not necessarily tightly, as follows: If $c_P > 0$, which means that the symmetric part of the operator P

is positive definite with respect to the inner product $[\cdot,\cdot]$, then the GMRES method converges and at the m^{th} iteration, the residual is bounded as

$$\|r_m\| \leq (1 - \frac{c_P^2}{C_P^2})^{m/2} \|r_0\|,$$

where $r_m = b - Px_m$. The algorithm is parameter-free and quite robust. Its main disadvantage is its linear-in-m memory requirement. To fit the available memory, one is sometimes forced to use the k-step restarted GMRES method [32]. However, in this case neither an optimal convergence property nor even convergence is guaranteed. Methods generally less rapidly convergent per matrix-vector-multiply than GMRES have recently been devised [19, 34] in order to overcome this limitation. In our applications, we restrict ourselves to preconditioners sufficiently "strong" that the total number of GMRES iterations is relatively small, and therefore no restarting is required. In the present paper, the linear operator P corresponds to the left- or right-preconditioned linear system, and b is the properly modified right-hand side. The simple L^2 inner product, together with its induced norm, is used in R^n.

2.3. Multiplicative Schwarz method (MSM)

The multiplicative Schwarz algorithm is a direct extension of the classical Schwarz alternating algorithm, introduced in 1870 by H. A. Schwarz in an existence proof for some elliptic boundary value problems in certain irregular regions. This method has attracted much attention as a convenient computational method for the solution of a large class of elliptic or parabolic equations. 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 and its theoretical aspects can be found in [6, 11, 27].

The coloring is realized as follows. Associated with the decomposition $\{\Omega_j^{'}\}$, we define an undirected graph in which nodes represent the extended subregions and the edges intersections of the extended subregions. This graph can be colored by using colors $0, \cdots, J$, such that no connected nodes have the same color. Obviously, colorings are not unique. Numerical experiments support the expectation that the minimizing the number of colors enhances convergence. An optimal five-color strategy ($J = 4$) is shown for the decomposition in Figure 2, in which the total number of subregions (including the coarse grid on the global region) is $N + 1 = 17$.

This algorithm can be employed in the stationary, Richardson sense or as a preconditioner for another algebraic iterative process. Along with the

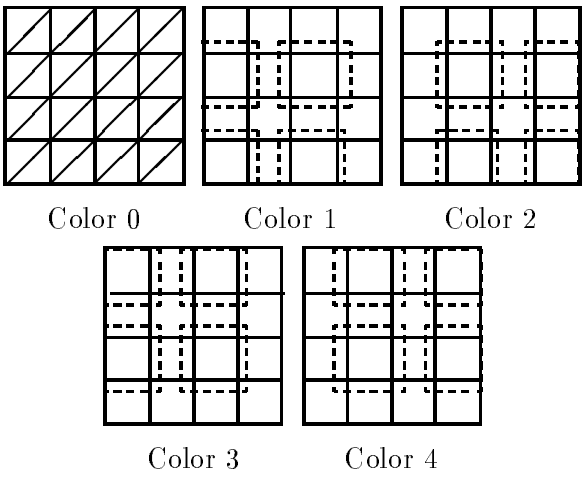

Fig. 2. The coloring pattern of 16 fine grid overlapped subregions and a coarse grid region. Color "0" is for the global coarse grid. The extended subregions of the other colors are indicated by the dotted boundaries.

other algorithms to be described below, we shall normally employ it as a preconditioner for GMRES, but because of its historical importance, and to illustrate certain robustness advantages of acceleration, 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 that corresponds to the classical Schwarz alternating algorithm with an extra coarse grid solver.

Letting $B_{h,0}^{'} = B_{h,0}$ and $R_0^{'} = R_0$, we describe the MSR algorithm in terms of a subspace correction process.

MSR algorithm: Let u_h^k be the current approximate solution. Then u_h^{k+1} is computed as follows. For $j = 0, 1, \cdots, J$:

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

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

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

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

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

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

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 series of steps (i)–(iii), the operations in subregions sharing the same color can be done in parallel.

Let us define the $n \times n$ matrices

$$M_i^{-1} = (R_i^{'})^t (B_{h,i}^{'})^{-1} R_i^{'} \quad \text{and } P_i = M_i^{-1} B_h, \text{ for } i = 0, \cdots, N.$$

For $j = 0, 1, \cdots, J$, if we denote Q_j as the sum of all $P_i^{'}$s and N_j^{-1} as the sum of all $M_i^{-1^{'}}$s that correspond to subregions of the j^{th} color, then MSR can be written in the following more compact form: For a given initial approximate solution u_h^0, and $k = 0, 1, \cdots$,

$$u_h^{k+1} = u_h^k + (I - E_{J+1})(f_h - B_h u_h^k) = E_{J+1} u_h^k + f_h^{'},$$

where the error propagation operator E_{J+1} is defined as

$$E_{J+1} = (I - Q_J) \cdots (I - Q_0)$$

and $f_h^{'} \equiv g_h^J$ is computed at a pre-iteration step by the following $J + 1$ sequential steps:

$$\begin{aligned} g_h^0 &= N_0^{-1} f_h \\ g_h^1 &= g_h^0 + N_1^{-1}(f_h - B_h g_h^0) \\ &\vdots \\ g_h^J &= g_h^{J-1} + N_J^{-1}(f_h - B_h g_h^{J-1}). \end{aligned}$$

Next, we shall discuss an accelerated version of MSR. We begin with the observation that if the matrix $I - E_{J+1}$ is invertible, then the exact solution of equation (2.2) also satisfies

$$(I - E_{J+1})\mathbf{u}_h = f_h^{'}, \tag{2.5}$$

which is sometimes referred as the *transformed*, or preconditioned, system corresponding to (2.2).

We next observe that for a given vector $v_h \in R^n$, the matrix-vector product $(I - E_{J+1})v_h$, denoted as v_h^J, can be computed in a manner similar to that of $f_h^{'}$, namely,

$$\begin{aligned} v_h^0 &= Q_0 v_h \\ v_h^1 &= v_h^0 + Q_1(v_h - v_h^0) \\ &\vdots \\ v_h^J &= v_h^{J-1} + Q_J(v_h - v_h^{J-1}). \end{aligned} \tag{2.6}$$

Now, the multiplicative Schwarz preconditioned GMRES method (MSM) can be described as follows: Find the solution of equation (2.2) by solving

the equation (2.5) with the GMRES method for a given initial guess and inner product.

Even in the case that the matrix B_h is symmetric positive definite, the iteration matrix $I - E_{J+1}$ is not symmetric. An obvious symmetrization exists, upon which a conjugate gradient method can be used as the acceleration method; however, we shall not emphasize the case of a symmetric B_h in this paper.

Inexact subdomain solves can easily be incorporated with either MSR or MSM. For $i \neq 0$, let $\tilde{B}_{h,i}^{'}$ be an $n_i^{'} \times n_i^{'}$ matrix that is spectrally equivalent to $B_{h,i}^{'}$. Then MSR or MSM with an inexact solver can be prescribed as follows: Repeat the preceding derivation except for the replacement of (ii) with

$$\tilde{B}_{h,i}^{'} e_h^{k+\frac{j}{J+1}} = R_i^{'} r_h^{k+\frac{j}{J+1}}.$$

Many inexact variants of the methods can be formulated. For example, for $j \neq 0$, let $A_{h,j}^{'}$ denote the matrix corresponding to the central difference discretization of the Poisson operator in the subregion $\Omega_j^{'}$. Then step (ii) can be carried out by

$$A_{h,j}^{'} e_h^{k+\frac{j}{J+1}} = R_j^{'} r_h^{k+\frac{j}{J+1}}.$$

Note that a "fast" solver, such as one based on FFTs, can now be applied.

Using an inexact solver for the interior subproblems, or an exact solver for approximate interior subproblems, can significantly reduce the overall computational complexity. This is, in fact, one of the major advantages of domain decomposition methods, in that they allow the use of fast solvers designed for special differential operators on regions of special shape. A somewhat disappointing experimental observation is that inexact solutions seem not to work well for the coarse grid solver. In fact, the existing theory for MSM [11], as well as the theory for ASM [10], requires an exact solve on the coarse grid.

In the piecewise linear finite element case, the convergence of MSR has been proved in [11], under certain assumptions. The rate of convergence is

$$\|u_h^k - \mathbf{u}_h\|_A \leq \left(\sqrt{1 - \frac{C_{\mathrm{MSR}}}{(J+1)^2}} \right)^k \|u_h^0 - \mathbf{u}_h\|_A,$$

where $C_{\mathrm{MSR}} > 0$ is a constant independent of h, H and J. The estimate holds in both two- and three-dimensional spaces. The assumptions include: (1) the overlap is uniform and must be $\mathcal{O}(H)$; (2) H must be sufficiently small; and (3) the number of colors, J, must be independent of the size of

the subregions H. The same estimate, with a different constant, holds for MSR with either exact or spectrally equivalent inexact solvers.

For the accelerated version MSM, under the same assumptions, we have that there exist two constants $C_{\rm MSM} > 0$ and $c_{\rm MSM} > 0$, independent of both h and H, such that the transformed system is uniformly bounded:

$$\|(I - E_{J+1})x\|_A \leq C_{\rm MSM}\|x\|_A, \ \ \forall x \in R^n, \tag{2.7}$$

and the symmetric part of the transformed system is positive definite in the inner product $(A_h\cdot,\cdot)$:

$$(A_h(I - E_{J+1})x, x) \geq c_{\rm MSM}\|x\|_A^2, \ \ \forall x \in R^n. \tag{2.8}$$

2.4. Additive Schwarz algorithm (ASM)

An additive variant of the Schwarz alternating method was originally proposed in [13, 14, 30] for selfadjoint elliptic problems and extended to non-selfadjoint elliptic cases in [7, 10]. The idea is simply to give up the data dependency between the subproblems defined on subregions with different colors, as in going from Gauss-Seidel to Jacobi. Instead of iterating with (2.6), one uses

$$\begin{array}{rcl} v_h^0 & = & Q_0 v_h \\ v_h^1 & = & v_h^0 + Q_1 v_h \\ & \vdots & \\ v_h^J & = & v_h^{J-1} + Q_J v_h . \end{array} \tag{2.9}$$

Of course, similar changes have to be made to the right-hand side vector. Coloring does not play a role at all in (2.9). Because of the lack of data dependency, the method is usually not to be recommended as a simple Richardson process, but as a preconditioner for some algebraic iterative methods of CG type. We denote by $M_{\rm ASM}$ the preconditioning part of (2.9). Following [10] and using the notation of the previous subsection, we can define the inverse of the matrix $M_{\rm ASM}$, referred to as the additive Schwarz preconditioner, as

$$\begin{array}{rcl} M_{\rm ASM}^{-1} & = & (R_0)^t(B_{h,0})^{-1}R_0 + (R_1^{'})^t(B_{h,1}^{'})^{-1}R_1^{'} + \cdots \\ & & +(R_N^{'})^t(B_{h,N}^{'})^{-1}R_N^{'}. \end{array} \tag{2.10}$$

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.

To obtain an optimal convergence rate, one does not have to solve these subproblems exactly. As proposed in [10], the following preconditioner is also optimal:

$$\begin{aligned} \tilde{M}^{-1}_{\mathrm{ASM}} &= (R_0)^t (B_{h,0})^{-1} R_0 + (R_1^{'})^t (\tilde{B}_{h,1}^{'})^{-1} R_1^{'} + \cdots \\ &\quad + (R_N^{'})^t (\tilde{B}_{h,N}^{'})^{-1} R_N^{'}. \end{aligned} \tag{2.11}$$

The $\tilde{B}_{h,i}^{'}$ are those defined in the previous subsection. It has been shown [7, 10] that, in the piecewise linear finite element case, both preconditioners M^{-1}_{ASM} and $\tilde{M}^{-1}_{\mathrm{ASM}}$ are optimal under the same first two assumptions made for MSM in the sense that there exist two constants $C_{\mathrm{ASM}} > 0$ and $c_{\mathrm{ASM}} > 0$, which may be different for exact and inexact subdomain solvers and are independent of both h and H, such that the preconditioned linear system is uniformly bounded:

$$\| M^{-1}_{\mathrm{ASM}} B_h x \|_A \leq C_{\mathrm{ASM}} \| x \|_A, \quad \forall x \in R^n \tag{2.12}$$

and the symmetric part of the preconditioned linear system is positive definite in the inner product $(A_h \cdot, \cdot)$

$$(A_h M^{-1}_{\mathrm{ASM}} B_h x, x) \geq c_{\mathrm{ASM}} \| x \|_A^2, \quad \forall x \in R^n. \tag{2.13}$$

In the case $B_h = A_h$, which means that the original elliptic operator is symmetric positive definite, the left-preconditioned system is symmetric positive definite in the $(A_h \cdot, \cdot)$ inner product; thus one can use a CG method. In the nonsymmetric case, the preconditioned system is nonsymmetric regardless of inner product. Therefore, instead of the A_h-inner product, we use the Euclidean inner product for the implementation presented in this paper. By giving up the symmetry requirement of the preconditioned system, we could also use ASM as a right-preconditioner. Neither of the pair of estimates (2.12) and (2.13) has been proved in the L^2 norm, but in the numerical experiments section, variability in ASM convergence rates measured (as is customary) with respect to L^2 residuals clearly diminishes as mesh and subdomain parameters are both refined, leading us to conjecture that analogous results hold.

The ASM discussed in this subsection can be used recursively for the solving the subdomain problems. The result is the multilevel ASM, as developed in [2, 16, 39].

2.5. Coarse grid plus SPD preconditioning (CSPD)

The low-frequency modes of the error are the hardest to damp with nearly any iterative method. Therefore, a direct solver is usually employed on the

coarse grid, as in multigrid methods. In the case of nonsymmetric problems, it is even more important to employ a direct coarse grid solver. Based on this observation, it is proved in [38] that a good preconditioner for B_h can be constructed by combining a properly weighted coarse-grid matrix, obtained by discretizing the original nonsymmetric elliptic operator, and some local symmetric positive definite matrices, obtained by discretizing the symmetric positive definite part of the elliptic operator. Other methods making special use of the coarse-grid matrix can also be found in [3, 37].

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 [38]. More precisely, let $\tilde{A}_h$ 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

$$M_{\rm CSPD}^{-1} = \omega (R_0)^t (B_{h,0})^{-1} R_0 + (\tilde{A}_h)^{-1}, \tag{2.14}$$

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 corresponds to the discretization of the second-order terms of $\mathcal{L}$ in Ω'_i with homogeneous boundary conditions. Then, we have

$$\begin{aligned} (\tilde{A}_h)^{-1} &= (R_0)^t (A_{h,0})^{-1} R_0 + (R'_1)^t (A'_{h,1})^{-1} R'_1 + \cdots \\ &\quad + (R'_N)^t (A'_{h,N})^{-1} R'_N. \end{aligned} \tag{2.15}$$

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, which use (2.15) in (2.14), the choice of $\omega = 1.0$ is acceptable. The issue of finding the optimal ω in the general case is not fully understood, but seems not to be critical.

The coarse-grid-plus-SPD-preconditioner (CSPD) method was analyzed in [38]. Suppose that the minimal and maximal eigenvalues of the preconditioned symmetric, positive definite part $(\tilde{A}_h)^{-1} A_h$ are λ_0 and λ_1. Then, if the coarse-mesh size, H, is fine enough, there exists a constant ω, depending on λ_0 and λ_1, such that $M_{\rm CSPD}^{-1}$ is an optimal preconditioner for B_h in the $(A_h \cdot, \cdot)$ inner product. That is, there exist constants $C_{\rm CSPD} > 0$ and $c_{\rm CSPD} > 0$ such that

$$\| M_{\rm CSPD}^{-1} B_h x \|_A \leq C_{\rm CSPD} \| x \|_A, \ \forall x \in R^n \tag{2.16}$$

and

$$(A_h M_{\rm CSPD}^{-1} B_h x, x) \geq c_{\rm CSPD} \| x \|_A^2, \ \forall x \in R^n. \tag{2.17}$$

It is clear that if $(\tilde{A}_h)^{-1}$ is an optimal preconditioner for A_h, which means that both λ_0 and λ_1 are independent of h and H, then with such an ω, also independent of λ_0 and λ_1, the convergence rate of the preconditioned nonsymmetric system is independent of h and H.

Although the analysis given in [38] holds only in the A_h inner product, and it is not known whether results similar to (2.16) and (2.17) hold in the Euclidean inner product, we again mention that all error and residual measurements behind Table 7 and Figure 7 are in L^2.

2.6. Tile algorithms (GK90/GK91)

The tile algorithm, proposed in [22], is designed especially for two-dimensional nonsymmetric problems and can be characterized as a nonoverlapping, multiplicative method. Numerical experiments indicate that the method converges at a rate that deteriorates logarithmically in the fine-mesh parameter, with the coarse-mesh size fixed. This method has been tested for a large class of linear and nonlinear problems on various parallel machines [23]. Our tables include results labeled GK90 and GK91.

The GK90 preconditioner is

$$M_{\mathrm{GK}}^{-1} = \begin{pmatrix} B_{II} & B_{IE} & B_{IC} \\ & T_{EE} & B_{EC} \\ & & B_{h,0} \end{pmatrix}^{-1} \begin{pmatrix} I & & \\ & I & \\ & W_E & W_C \end{pmatrix}, \tag{2.18}$$

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 dropped. The matrices W_E and W_C define a so-called ramp-weighted averaging method used to modify the right-hand side of the coarse mesh problem involving $B_{h,0}$. W_C is diagonal with all positive elements, all elements of W_E are nonnegative, and their rows together sum to unity. A detailed description can be found in [22].

To perform the triangular solve, one needs three sequential steps: 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.

It is observed in [22] that some additional saving can be achieved if $M_{\rm GK}^{-1}$ is used as a right-preconditioner, because then a simple calculation shows that

$$B_h M_{\rm GK}^{-1} = \begin{pmatrix} I & 0 & 0 \\ * & * & * \\ * & * & * \end{pmatrix} \begin{pmatrix} I & & \\ & I & \\ & W_E & W_C \end{pmatrix},$$

where $*$ denotes a nonzero block. The identity block row means that $O(h^{-2})$ of the unknowns in the Krylov vectors can go untouched (except for scaling) throughout the entire solution process until the preconditioning is unwound in the final step, after the interface and crosspoint values have converged. Since B_{II}^{-1} is needed to advance the solution on these separator sets, we cannot escape solving subdomain problems, but substantial arithmetic work is saved.

A more recent tile algorithm (GK91) incorporates an additional refinement. The right-hand side of each interface problem is modified prior to its solution using T_{EE} to include an approximation to the nontangential terms of the PDE on the interface. These terms are formed from bivariate interpolation of the coarse-mesh solution, quadratic normal to the interface and linear tangential to it. As is apparent from the tables to follow, the additional sequential stage can have a substantial impact on the convergence rate of the tile algorithm without any additional subdomain solves per iteration. The extra communication required, relative to GK90, is all of near-neighbor type.

The GK91 preconditioner, written out in factored form, is

$$M_{\rm GK}^{-1} = \begin{pmatrix} B_{II} & & \\ & I & \\ & & I \end{pmatrix}^{-1} \begin{pmatrix} I & B_{IE} & B_{IC} \\ & I & \\ & & I \end{pmatrix} \begin{pmatrix} I & & \\ & T_{EE} & \\ & & I \end{pmatrix}^{-1}$$

$$\begin{pmatrix} I & & \\ & I & [B_{EC} - N_{EC}] \\ & & I \end{pmatrix} \begin{pmatrix} I & & \\ & I & \\ & & B_{h,0} \end{pmatrix}^{-1} \begin{pmatrix} I & & \\ & I & \\ & W_E & W_C \end{pmatrix}.$$

The normal derivative corrections are in the N_{EC} term. The three sequential solves shown are optionally followed by a relaxation step for the crosspoints on the fine-grid.

2.7. Substructuring algorithm (CGK)

The CGK algorithm, developed in [9], was motivated by the tile algorithm of [22] and also the iterative substructuring algorithm of [5]. Its major differences relative to the tile algorithms are that it needs an extra set of interior solves and that the interface and interior solves are made independent of the coarse-grid problem. The first allows a proof of near-optimal convergence rate, and the second offers more flexible parallelization. When solving a highly nonsymmetric problem, one must use a coarse grid that is often too large to be handled by a single processor or solved redundantly in each processor. The independence of the coarse-grid problems from the others makes it possible to solve the coarse-grid problem on a collection of MIMD processors, while other processors perform interface and interior problems in parallel.

The major difference between CGK and the BPS-I algorithm [5] for SPD systems is that the coarse-grid solve depends sequentially on the interface and interior solves in BPS-I. The price for the extra parallelism in CGK is that the convergence bound suffers an extra factor of $(1 + \ln(H/h))$, as seen below.

The nearly optimal, nonoverlapping, partly additive, partly multiplicative CGK method can be expressed as

$$M_{\mathrm{CGK}}^{-1} = (R_0)^t B_{h,0}^{-1} R_0 + \begin{pmatrix} B_{II}^{-1} + B_{II}^{-1} B_{IE} K_{EE}^{-1} B_{EI} B_{II}^{-1} & -B_{II}^{-1} B_{IE} K_{EE}^{-1} & 0 \\ -K_{EE}^{-1} B_{EI} B_{II}^{-1} & K_{EE}^{-1} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.19}$$

where K_{EE} is a block diagonal matrix, and each block corresponds to an interface. In our current implementation, every block in K_{EE} has the form of the square root of the negative one-dimensional Laplacian along the interface, with size equal to the number of interior interface nodes. There are other possibilities for K_{EE} (see, e.g., [12]) which better adapt the preconditioner to nonsymmetric and variable coefficient problems.

To form the action of $M_{\mathrm{CGK}}^{-1} u_h$, 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.

This algorithm is analyzed in [9] for a piecewise linear finite element discretization in R^2. The convergence rate degenerates logarithmically with the grid size. More precisely speaking, there exists a small positive constant

Table 1
The *sequential stages* of the algorithms as well as BPS(I)

MSR	MSM	ASM	CSPD	GK	CGK	BPS(I)
$J+1$	$J+1$	1	1	3	3	4

H_0, such that if

$$H(1+\ln(H/h)^3) \leq H_0$$

holds, then one has

$$\|M_{\rm CGK}^{-1} B_h x\|_A \leq C_{\rm CGK} \|x\|_A, \ \forall x \in R^n \tag{2.20}$$

and

$$(A_h M_{\rm CGK}^{-1} B_h x, x) \geq c_{\rm CGK}/(1+\ln(H/h)^3) \|x\|_A^2, \ \forall x \in R^n, \tag{2.21}$$

where $C_{\rm CGK}$ and $c_{\rm CGK}$ are positive constants independent of H and h but possibly dependent on H_0. It was also proved that if the matrix B_h is symmetric positive definite, then the estimates are independent of H_0; in other words, the coarse grid does not need to be "sufficiently small" in this case.

3. Parallelism and Complexity Analysis

In this section, we discuss some the issues of parallelism and parallel arithmetic complexity of the algorithms described in the preceding section.

A natural measure of the parallelism of an algorithm is the number of *sequential stages* it contains. Normally, parallelization can be accomplished only within each stage and not between the stages, which act as synchronization points. The counts of sequential stages for the six preconditioner algorithms introduced in the preceding section, as well as for the BPS(I) algorithm of [5], are summarized in the Table 1.

For convenience in providing a simplified parallel complexity analysis, focusing only on computer architecture-independent factors, we begin with the assumption that each subdomain is undivided in the mapping onto processors. Furthermore, we assume that all interior problems, defined on any unextended substructures, are of relatively the same size and require t_I unit time (or number of arithmetic operations) to solve. Of course, t_I depends not only on how many unknowns each subregion has but also on the method

used to solve the interior problem. Similarly, we denote by t_I' the time required for the interior problems on extended subregions, t_C for the crosspoint problem, t_E for each interface problem.

Let ρ_{ovlp} denote the overlapping factor, which is defined as the ratio of the number of protruding h-level grid points in Ω_i' relative to the number of h-level grid points on a side of Ω_i. If we assume that the complexity of the subdomain solve is proportional to leading order to the γth power of the number of gridpoints, then, approximately,

$$t_I' = (1 + 2\rho_{ovlp})^{\gamma d} \cdot t_I,$$

Here, d (= 2 or 3) is the dimension of the problem, and a uniform overlap is assumed in all directions. We are able to recommend a small ρ_{ovlp} in practice, corresponding to just one or two fine mesh cells, but we include examples below that use 50% overlap, implying that every point not on a separator belongs to at least two different interiors.

The interface arithmetic complexity t_E can usually be ignored compared with t_I or t_C, at least for two-dimensional problems. The times t_I and t_C depend heavily on what kind of solver being used. For example, if each interior differential operator is approximated by a constant times the Laplacian, a uniform grid is used on the square subregion, and the matrix is suitably preprocessed, then the work of the FFT-based fast Poisson solver is approximately

$$t_I = O((n/N)\log_2(n/N)),$$

where the term n/N corresponds approximately the number of h-level grid points in Ω_i. Sometimes a multiple of the Laplacian is not good enough to approximate the interior matrix, and then the original matrix may be used. In this case, a more expensive banded direct solver is used and

$$t_I = O((n/N)^{(2-1/d)}).$$

Other, more efficient solvers for the interior problems have been developed. For example, a few cycles of multigrid or a few Gauss-Seidel iterations perform well for some test problems, as in [2] or [33].

The crosspoint matrix is usually obtained by discretizing the original differential equation. Using an approximate operator in this case usually results in large iteration counts for our nonsymmetric problems, for reasons that are not yet explained theoretically. A banded direct solver for the crosspoint system needs approximately

$$t_C = O(N^{(2-1/d)}).$$

For $i \neq 0$, the actions of the restriction and extension operators R_i, R_i', $(R_i)^t$, and (R_i') on vectors do not involve any arithmetic operations. This is not the case for the coarse-grid problem. If one uses the piecewise linear interpolation, based on the coarse-mesh triangulation shown for "Color 0" in Figure 2, then the cost of applying R_0 and $(R_0)^t$ is approximately $6n$. Other restriction and extension operators, such as those usually used in multigrid methods, may also be applied. We denote by t_0 the number of arithmetic operations for applying the coarse grid interpolation operator.

In the tile algorithms a ramp-weighted averaging method is used to calculate the right-hand-side of the coarse-mesh problem at each GMRES iteration. This method uses the fine-mesh values only on the interfaces. Approximately $t_{0,\mathrm{GK}} \equiv 4N\sqrt{n/N}$ flops are needed. In GK91 an comparable additional number of flops are needed to adjust the right-hand sides of the interfacial systems with the non-tangential corrections. Neither form of the tile algorithm requires an $O(n)$ interpolation from the crosspoint to the subdomain interior degrees of freedom, or back again.

The complexity of the GMRES method was analyzed in [32]. Since we consider only the five-point finite difference discretization, the action of the stiffness matrix on a given vector requires approximately $5n$ flops. In the sequential case, I steps of GMRES require $I(I+2)n + I(5n)$ multiplications. For simplicity, we identify the number of flops with the number of multiplications. The parallel arithmetic complexity of GMRES, with p processors, is then approximately

$$C_{\mathrm{GMRES}}(I,p) \;=\; I(I+1)\;\left(\frac{n}{p} + \log_2(p)\right) \;+\; I\,\frac{n}{p} \;+\; I\,\frac{5n}{p},$$

where the term $I(I+1)(\frac{n}{p} + \log_2(p))$ is from the dot products and DAXPYs of the Gram-Schmidt process, the term $\frac{n}{p}I$ from the forming of the new approximate solution after I steps are complete, and $\frac{5n}{p}I$ from the matrix-vector multiply.

The parallel arithmetic complexity is estimated in Table 2 for p equal to the number of subproblems and in Table 3 for p less than the number of subproblems.

It is also important to consider the parallel communication complexity, though its impact is architecture dependent. Whether the sequential stages of Table 1 are enforced by barriers or by the arrival of data, they contribute a term proportional to the message latency of the multiprocessor. The global inner products and matrix-vector products of GMRES also add latency terms. In addition, an amount of data proportional to $\sqrt{n/N}$ and the discrete width of the overlap must be communicated in both directions over each boundary at least once in each preconditioner application. Finally,

Table 2
The parallel complexity of the algorithms with the number of processors p equal to the number of subproblems

Method	Number of iterations $I =$	Total flops $p =$ number of subproblems
MSR	$O(1)$	$I\left(Jt_I^{'} + t_C + \frac{t_0}{p}\right) + 5\frac{n}{p}$
MSM	$O(1)$	$C_{\mathrm{GMRES}}(I,p) + I\left(Jt_I^{'} + t_C + \frac{t_0}{p}\right)$
ASM	$O(1)$	$C_{\mathrm{GMRES}}(I,p) + I\left(\max\{t_I^{'}, t_C\} + \frac{t_0}{p}\right)$
CSPD	$O(1)$	same as ASM
GK		$C_{\mathrm{GMRES}}(I,p) + I\left(t_I + t_C + t_E + \frac{t_{0,\mathrm{GK}}}{p}\right)$
CGK	$O(1+\ln(H/h))^3$	$C_{\mathrm{GMRES}}(I,p) + I\left(\max\{2t_I + t_E, t_C\} + \frac{t_0}{p}\right)$

Table 3
The parallel complexity of the algorithms with the number of processors p less than the number of subproblems

Method	Total flops $p \leq$ number of subproblems
MSR	$I\left(J\frac{(N/J)\ t_I^{'}}{p} + t_C + \frac{t_0}{p}\right)$
MSM	$C_{\mathrm{GMRES}}(I,p) + I\left(J\frac{(N/J)\ t_I^{'}}{p} + t_C + \frac{t_0}{p}\right)$
ASM	$C_{\mathrm{GMRES}}(I,p) + I\left(\max\{\frac{Nt_I^{'}}{p}, t_C\} + \frac{t_0}{p}\right)$
CSPD	same as ASM
GK	$C_{\mathrm{GMRES}}(I,p) + I\left(\frac{Nt_I}{p} + t_C + \frac{2Nt_E}{p} + \frac{t_{0,\mathrm{GK}}}{p}\right)$
CGK	$C_{\mathrm{GMRES}}(I,p) + I\left(\max\{2\frac{Nt_I}{p} + \frac{2Nt_E}{p}, t_C\} + \frac{t_0}{p}\right)$

the global coarse grid problem constitutes (either in its setting up or in its solution) a communication bottleneck. For an extended discussion of the parallel communication complexity of domain decomposition algorithms for two-dimensional problems, see [21].

4. Numerical Experiments

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

$$\left\{\begin{array}{rcll} \mathcal{L}\mathbf{u} & = & f & \text{in } \Omega \\ \mathbf{u} & = & 0 & \text{on } \partial\Omega, \end{array}\right. \tag{4.1}$$

where different elliptic operators $\mathcal{L}$ will be specified and $\Omega = [0,1] \times [0,1]$. In all cases, the exact solution $\mathbf{u} = e^{xy}\sin(\pi x)\sin(\pi y)$, 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 then 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 with zero initial guess. The stopping criterion is the reduction of the initial (preconditioned) residual by five orders of magnitude in the L^2 norm, namely

$$(r_k, r_k)^{1/2} \leq 10^{-5}(r_0, r_0)^{1/2},$$

where $r_k = M^{-1}(f_h - B_h u_h^k)$, for $k \geq 0$, and M^{-1} is one of the preconditioners discussed previously.

We remark that this stopping criterion is preconditioner dependent. Let $e_k = \mathbf{u} - u_h^k$ be the true error. The preconditioned residual $r_k = M^{-1}B_h e_k$ reflects the true error only if the preconditioner is so strong that both

$$\|(M^{-1}B_h)^{-1}\|_{L^2} \text{ and } \|M^{-1}B_h\|_{L^2}$$

are close to order one. From the finite element theory associated with most of the methods, this is true asymptotically in H and h if the L^2 norm is replaced by the energy norm $\|\cdot\|_A$. To see whether this is so in the range of mesh, subdomain, and problem parameters in our experiments, we take advantage of the exact solution to complement the residual-based iteration count data of Tables 4 through 7, with plots of the true error as a function of iteration index in Figures 3 through 7. We point out some cases in which monitoring the residual does not give a complete picture of the relative quality of different methods. Unfortunately, this is probably often the case for results reported in the literature for nonsymmetric elliptic problems; however, in many situations authors have no other practical recourse than to base convergence on (preconditioned) residual.

Incomplete LU (ILU) decomposition [29] results are shown for zero, one, and two levels of fill [35].

Double precision is used throughout. Only machine-independent information, namely, the iteration count and true error as a function of iteration, is presented. Throughout this section, we use "ovlp" to denote the size of overlap. That is to say, the distance between the boundaries of an extended subregion and the original subregion is equal to "ovlp" except near corners, where it can be up to a factor of $\sqrt{2}$ greater.

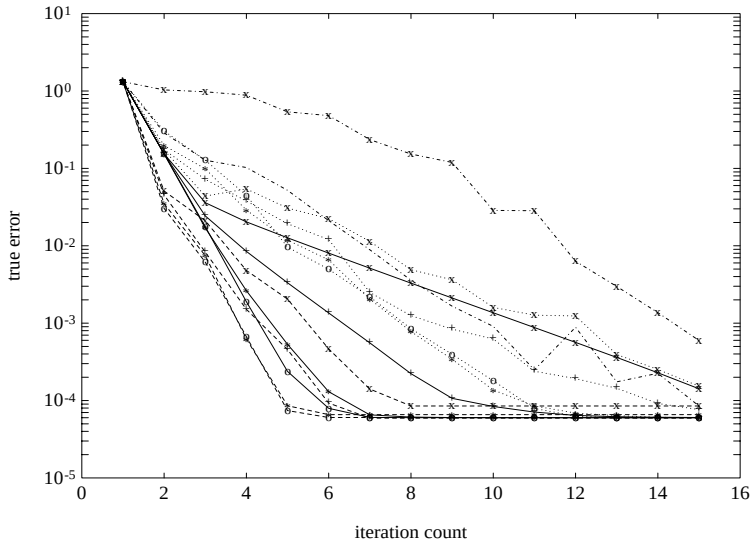

Fig. 3. The L_∞ norm true error reduction of the Poisson equation with $h = 1/128$ and $H = 1/4$. "solid x": MSR(ovlp=h); "solid +": MSR(ovlp=$2h$); "solid $*$ ": MSR(ovlp=$4h$); "solid o ": MSR(ovlp=$8h$); "dashed x": MSM(ovlp=h); "dashed +": MSM (ovlp=$2h$); "dashed $*$ ": MSM(ovlp=$4h$); "dashed o ": MSM(ovlp=$8h$); "dotted x": ASM(ovlp=h); "dotted +": ASM(ovlp=$2h$); "dotted $*$ ": ASM(ovlp=$4h$); "dotted o ": ASM(ovlp=$8h$); "dashdot": CGK; "dashdot x": GK91.

4.1. 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 positive definite; therefore, with a suitable inner product, CG is more efficient. The iteration matrices of MSM, MSR, and the tile algorithm are nonsymmetric. CSPD is not designed for such a test problem and therefore is not tested. The iteration counts are given in Table 4. (Entries that would have required overlap greater than 50% are left blank.) Among all algorithms with small overlap, the MSM takes the least number of iterations. Since MSR does not depend on an outer algebraic iterative method such as GMRES, it requires the least amount of computer memory.

Within each group of columns sharing a common H, the iteration counts are constant along diagonals of the table for the overlapping methods (MSR,

Table 4
Iteration counts for solving the Poisson equation. The overlapping factors for MSR are given as the subscript of iteration numbers. The corresponding overlapping factors for MSM and ASM are the same as that for MSR and are therefore omitted. The number, such as $(2h)$, which appears next to the name of each method indicates the size of overlap.

$h^{-1} =$	32	64	128	32	64	128	64	128
Methods	$H = 1/4$			$H = 1/8$			$H = 1/16$	
MSR(h)	$7_{(\frac{1}{8})}$	$11_{(\frac{1}{16})}$	$19_{(\frac{1}{32})}$	$6_{(\frac{1}{4})}$	$7_{(\frac{1}{8})}$	$10_{(\frac{1}{16})}$	$5_{(\frac{1}{4})}$	$6_{(\frac{1}{8})}$
MSR($2h$)	$6_{(\frac{1}{4})}$	$7_{(\frac{1}{8})}$	$11_{(\frac{1}{16})}$	$5_{(\frac{1}{2})}$	$6_{(\frac{1}{4})}$	$7_{(\frac{1}{8})}$	$4_{(\frac{1}{2})}$	$5_{(\frac{1}{4})}$
MSR($4h$)	$5_{(\frac{1}{2})}$	$6_{(\frac{1}{4})}$	$7_{(\frac{1}{8})}$		$5_{(\frac{1}{2})}$	$6_{(\frac{1}{4})}$		$4_{(\frac{1}{2})}$
MSR($8h$)		$5_{(\frac{1}{2})}$	$6_{(\frac{1}{4})}$			$5_{(\frac{1}{2})}$		
MSM(h)	5	6	7	4	4	5	3	3
MSM($2h$)	5	5	6	4	4	4	3	3
MSM($4h$)	4	5	5		4	4		3
MSM($8h$)		4	5			4		
ASM(h)	11	13	15	10	10	11	9	8
ASM($2h$)	11	11	13	10	10	10	8	8
ASM($4h$)	10	11	11		10	10		8
ASM($8h$)		10	11			10		
GK91	14	19	28	9	17	32	9	16
CGK	12	13	13	12	11	11	10	10

MSM and ASM). The ratio $\frac{ovlp}{H}$ is the same for each case along a diagonal, as tabulated in parentheses. This suggests that the iteration count depends only on the ratio $\frac{ovlp}{H}$, but not on the actual mesh sizes h or H. This observation has recently been proved in [36].

The true error reduction curves corresponding to some entries of the $H = 1/4, h = 128$ column of Table 4 are given in Figure 3. The multiplicative Schwarz preconditioned GMRES with sufficiently large overlap takes the least number of iterations to reduce the initial error to the discretization error. We also note that the curve for the nonoverlapping method CGK is very close to the curve for ASM with the minimum amount of overlap.

4.2. A nonsymmetric problem

Our second test problem is a nonsymmetric, constant coefficient problem corresponding to a uniform convection skewed at 45° to the coordinate axes:

$$\mathcal{L}u = -\triangle u + \delta u_x + \delta u_y.$$

We specify different values for the constant convection strength $\delta > 0$ in Table 5. The first-order terms of the elliptic operator are discretized by two

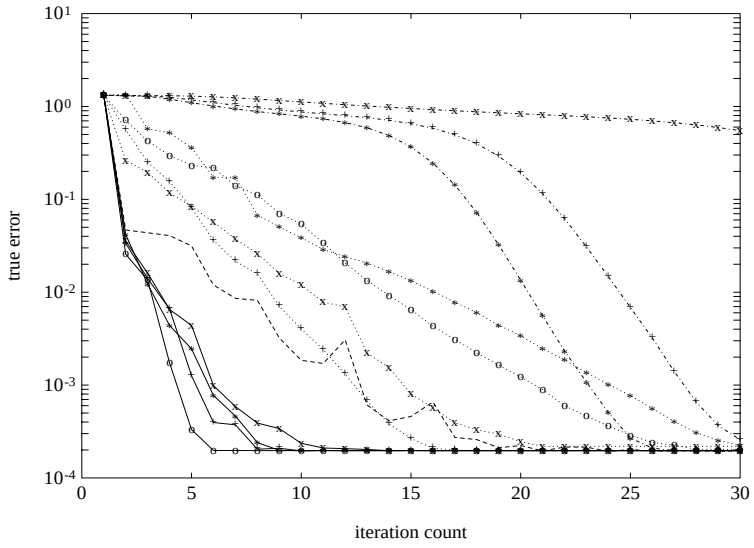

Fig. 4. The L_∞ norm true error reduction of the central differenced nonsymmetric problem with parameters $\delta = 50$, $h = 1/128$ and $H = 1/8$. "solid x": MSR(ovlp=h); "solid +": MSR(ovlp=$4h$); "solid $*$": MSM(ovlp=h); "solid o": MSM(ovlp=$4h$); "dotted x": ASM(ovlp=h); "dotted +": ASM(ovlp=$4h$); "dotted $*$": CSPD($\omega = 1.0$); "dotted o": CGK; "dashed": GK91; "dashdot x": ilu(0); "dashdot +": ilu(1); "dashdot $*$": ilu(2).

schemes, namely, the central difference method for δ of $O(h^{-1})$ or less and the upwind-difference method for larger δ.

When using the central difference method, for a fixed fine mesh size $h^{-1} = 128$, we observe that as δ is increased beyond a certain size (near 10), all methods, except MSM with sufficient overlap, show a sharp upturn in the iteration count. The MSR loses convergence if δ is larger than this transitional δ for essentially all overlapping sizes. All other GMRES-based methods continue to converge but at a slower rate, especially the nonoverlapping methods. The nonoverlapping methods with uncorrected interfaces have difficulty handling large convection terms; however, GK91 improves in this limit and even surpasses ASM when $H^{-1} = 4$. Even when GK91 trails ASM in iteration count based on a fixed preconditioned residual reduction, its true error versus iteration count curve lies mostly below that of ASM. Distinct from all other methods, MSM converges in a small number of steps that, in cases of generous overlap, is almost independent of the strength of convection. The error reduction curves that correspond to the column $h = 1/8$, $\delta = 50$ in the upper half of Table 5 are given in Figure 4.

The general complexion changes when we switch to the upwind-difference method. The iteration counts for the overlapping methods remain nearly

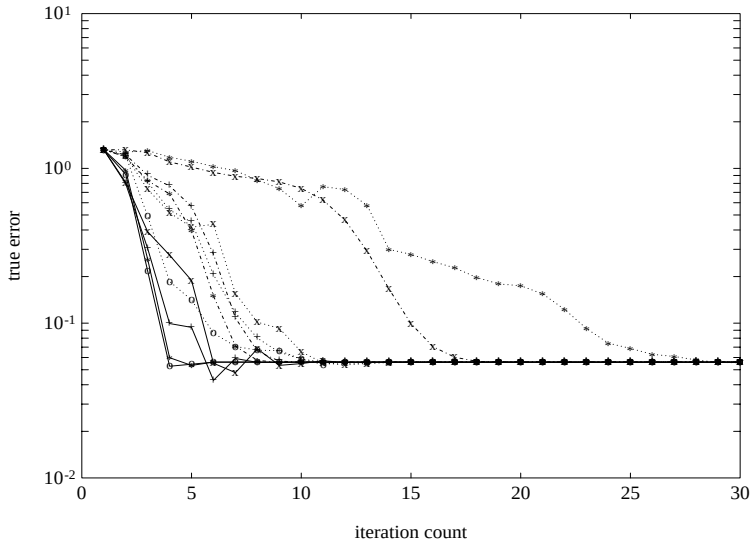

Fig. 5. The L_∞ norm true error reduction of the upwind-differenced nonsymmetric problem with parameters $\delta = 500$, $h = 1/128$ and $H = 1/4$. "solid x": MSR(ovlp=h); "solid +": MSR(ovlp=$4h$); "solid $*$": MSM(ovlp=h); "solid o": MSM(ovlp=$4h$); "dotted x": ASM(ovlp=h); "dotted +": ASM(ovlp=$4h$); "dotted $*$": CGK; "dotted o": GK91; "dashdot x": ilu(0); "dashdot +": ilu(1); "dashdot $*$": ilu(2).

constant even for large δ. With modest overlap — just 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 δ increases, again with the exception of GK91, which again surpasses ASM.

From these results, a strong connection is evident 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, with upwind-differencing, refining the coarse mesh may not always reduce the number of iterations.

For this set of problem parameters, MSM is the most robust method and behaves well in all cases. The unaccelerated multiplicative Schwarz algorithm (MSR) is too sensitive to the stability of the discretization.

The nonoverlapping methods without interface correction do not behave well if the constant δ is large with either discretization scheme. From comparing GK90 and GK91, we believe that this result is mostly caused by the

interface preconditioner. Experimentation with different flow directions in [12] showed that a skew orientation of the flow with respect to the interface was worse for the tangential preconditioner than either normal or tangential flow orientation. Meanwhile, the interface preconditioner employed in CGK makes no adaptation whatever to the presence or alignment of the convection terms.

The error reduction curves that correspond to the column $H = 1/4$, $\delta = 500$ in the lower half of Table 5 are given in Figure 5. For the MSM cases, it takes only five iterations to reach the discretization error and for the MSR cases, the true errors reach the discretization error at about the eighth iteration and have almost no further reductions after this point while the L^2 norm of preconditioned residual is still decreasing. All other methods need more iterations to reach the discretization error. Of course, the use of first-order upwinding in the overall system matrix — not just in the preconditioner — limits the terminal accuracy of the scheme (note the compressed vertical axis in Figure 5 relative to Figure 4).

The ILU results (applied to the global domain, $H = 1$) are complementary to the rest. For this particular constant-coefficient test problem, they do not work well for problems that yield easily to domain-decomposed methods, but work very well on the other end. This is because ILU is sensitive to the signs and magnitudes of the coefficients of the nonsymmetric terms, as well as to the discretization parameter h. Some analysis was given in [18]. The central difference ILU results begin to 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 included in Table 7 below.

4.3. The Helmholtz equation

Our third test problem is a Helmholtz equation with constant coefficients

$$\mathcal{L}u = -\triangle u - \sigma u.$$

It is self-adjoint, but indefinite. The eigenvalues of the continuous 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 6 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.

Table 5

Iteration count for solving the nonsymmetric model equation. The fine mesh size is uniformly $1/h = 128$. $(*h)$ denotes the overlap size. $\omega = 1.0$ for CSPD. $H = 1$ for the ILU results.

Methods	$H = 1/4$						$H = 1/8$					
Central-difference Method												
$\delta =$	1	5	10	50	100	150	1	5	10	50	100	150
MSR(h)	19	18	15	∞	∞	∞	10	10	10	13	∞	∞
MSR($2h$)	12	11	9	21	∞	∞	7	7	7	14	∞	∞
MSR($4h$)	7	8	8	22	∞	∞	6	6	6	10	35	∞
MSR($8h$)	6	7	8	24	∞	∞	5	5	5	7	10	21
MSM(h)	7	7	7	10	10	9	5	5	5	8	10	12
MSM($2h$)	6	6	6	8	8	8	4	4	4	7	8	11
MSM($4h$)	5	5	6	7	7	7	4	4	4	5	7	9
MSM($8h$)	5	5	5	6	6	6	4	4	4	4	5	7
ASM(h)	15	17	18	22	22	21	11	12	12	20	26	32
ASM($2h$)	13	15	15	20	20	21	10	10	11	18	23	27
ASM($4h$)	12	13	13	18	19	20	10	11	11	15	20	23
ASM($8h$)	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	11	13	15	37	57	73	10	12	13	26	42	55
ILU(0)	60	84	81	59	41	27						
ILU(1)	38	53	51	34	22	15						
ILU(2)	31	46	42	28	19	13						
Upwind-difference Method												
$\delta =$	10	50	100	500	10^3	10^4	10	50	100	500	10^2	10^4
MSR(h)	18	14	13	18	18	18	10	13	14	21	23	23
MSR($2h$)	14	14	15	16	16	16	10	13	16	17	16	15
MSR($4h$)	12	13	14	13	12	12	9	12	12	12	12	12
MSR($8h$)	10	10	10	11	11	11	8	9	9	8	8	8
MSM(h)	9	9	8	7	7	7	7	9	9	10	11	11
MSM($2h$)	8	8	7	7	7	7	7	8	8	9	9	9
MSM($4h$)	7	7	6	6	6	6	7	7	6	6	6	6
MSM($8h$)	5	5	5	5	5	5	5	5	5	5	5	6
ASM(h)	19	20	19	18	17	17	14	19	21	22	22	23
ASM($2h$)	17	18	16	16	17	17	14	17	19	19	20	19
ASM($4h$)	15	16	16	16	16	16	14	15	16	17	17	18
ASM($8h$)	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
ILU(0)	82	61	50	23	16	6						
ILU(1)	51	36	28	12	9	4						
ILU(2)	42	30	24	11	8	4						

Table 6
Iteration count for solving the Helmholtz equation. The fine mesh size is uniformly $1/h = 128$. $(*h)$ denotes the overlap size.

$\sigma =$	0	30	70	110	150	300	0	30	70	110	150	300
Methods	$H = 1/8$						$H = 1/16$					
MSR(h)	10	11	20	∞	∞	∞	6	6	6	7	21	∞
MSR($2h$)	7	7	16	∞	∞	∞	5	5	5	5	14	∞
MSR($4h$)	6	6	14	∞	∞	∞	4	4	5	5	12	∞
MSM(h)	5	5	7	9	13	35	3	4	4	4	6	8
MSM($2h$)	4	4	6	8	12	37	3	3	4	4	6	9
MSM($4h$)	4	4	5	8	13	> 100	3	3	4	4	6	9
ASM(h)	11	12	14	19	23	62	8	9	9	10	11	16
ASM($2h$)	10	10	14	18	23	61	8	8	9	10	10	16
ASM($4h$)	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

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. The error reduction curves that corresponds to the column $H = 1/16, \sigma = 70$ in Table 6 are given in Figure 6.

4.4. 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 coefficients of the second-order terms oscillate but do not vary in sign. The coefficients of the first-order terms physically represent a ten-by-ten

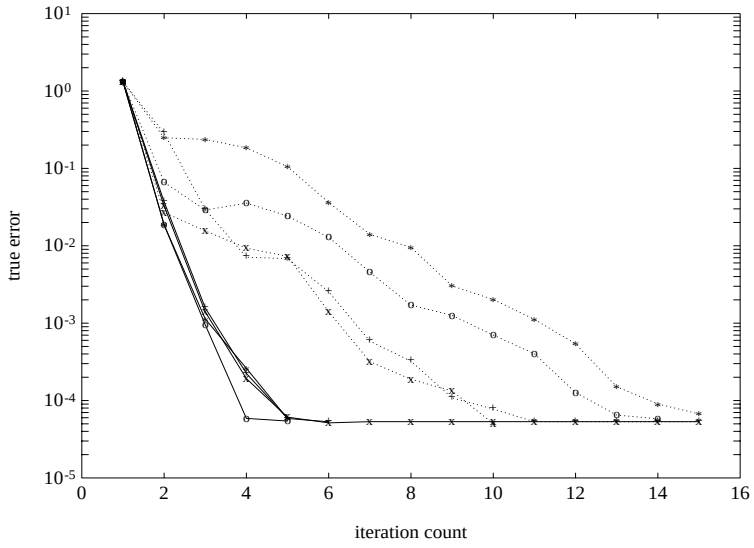

Fig. 6. The L_∞ norm true error reduction of the central differenced Helmholtz equation with parameters $\sigma = 70$, $h = 1/128$ and $H = 1/16$. "solid x": MSR(ovlp=h); "solid +": MSR(ovlp=$4h$); "solid $*$": MSM(ovlp=h); "solid o": MSM(ovlp=$4h$); "dotted x": ASM(ovlp=h); "dotted +": ASM(ovlp=$4h$); "dotted $*$": CSPD($\omega = 1.0$); "dotted o": CGK;

array of closed convection cells, with no convective transport between cells. However, the subdomain boundaries do not in general align with the convection cell boundaries, so this zero-convective-flux 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.

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 the oscillatory coefficients in the second-order terms of $\mathcal{L}$. The discretization becomes more stable when h gets smaller relative to the coefficient oscillation wavelength.

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.

The error reduction curves that correspond to the last column of Table 7, except the curve for MSR, are given in Figure 7. The nonoverlapping

Table 7
Iteration counts for solving the variable-coefficient, nonsymmetric indefinite problem.

	$H = 1/4$			$H = 1/8$			$H = 1/16$	
$h =$	1/32	1/64	1/128	1/32	1/64	1/128	1/64	1/128
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)	44	78	312					
ILU(1)	28	44	99					
ILU(2)	22	36	76					

methods GK90, GK91, and CGK appear poorer as a group than overlapping methods from the residual reduction tabulations. However, the error curves show that they achieve the same quality of solution in only a few extra iterations.

For this variable-coefficient problem, the global ILU preconditioners are overwhelmingly outperformed at fine mesh sizes by the domain decomposition preconditioners. Well beyond the iteration counts at which all of the domain decomposed methods have achieved truncation error accuracy, the ILU curves appear to plateau, but truncation error accuracy is ultimately achieved by these methods.

5. Concluding Remarks

We have implemented and tested several 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 and it is nontrivial to choose a single criterion, such as the number of iterations to reach a given residual reduction, by which to make universal comparisons. We have highlighted some comparisons we consider interesting and some conjectures 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

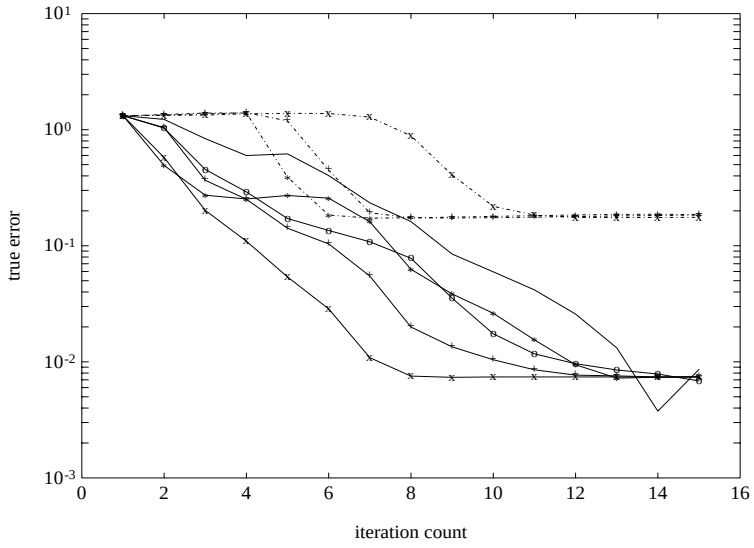

Fig. 7. The L_∞ norm true error reduction of the central differenced variable-coefficient, nonsymmetric indefinite problem. $h = 1/128$ and $H = 1/16$. "solid x": MSM(ovlp=$2h$); "solid +": ASM(ovlp=$2h$); "solid $*$": CSPD($\omega = 1.0$); "solid o": CGK; "solid": GK91; "dashdot x": ilu(0); "dashdot +": ilu(1); "dashdot $*$": ilu(2).

discretization scheme, not only on the underlying differential operator. In general, both the discretization and the outer iterative method must be considered in conjunction with the preconditioning strategy. However, the modest-overlap GMRES-accelerated multiplicative Schwarz method can be recommended as a consistently good performer throughout the test suite. We have only begun to extend the performance characterization to parallel architectures. On parallel machines, the robustness of MSM comes with a dual price: the parallel granularity is intrinsically less, for a given number of subdomains, than additive or other methods; and subdomains must be mapped to to processors in multiples of the number of colors in order to obtain good utilization throughout each of the differently colored parallel stages.

Acknowledgement

The work of first author was supported in part by the National Science Foundation and the Kentucky EPSCoR Program under grant STI-9108764. The work of second 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. The work of third author was supported in part by the NSF under contract ECS-8957475 and by the National Aeronautics and Space Administration under NASA Contract NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center Hampton, VA 23665.

The helpful criticism of anonymous referees is gratefully acknowledged.

References

[1] R. E. Bank, J. F. Bürgler, W. Fichtner and R. K. Smith, *Some upwinding techniques for the finite element approximation of convection-diffusion equations* Numer. Math., 58 (1990) 185–202.

[2] P. E. Bjørstad and M. D. Skogen, *Domain decomposition algorithms of Schwarz type, designed for massively parallel computers,* in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs and R. G. Voigt, eds., SIAM, Philadelphia (1992).

[3] J. H. Bramble, Z. Leyk and J. E. Pasciak, *Iterative schemes for nonsymmetric and indefinite elliptic boundary value problems*, Cornell University, preprint, (1991).

[4] J. H. Bramble and J. E. Pasciak, *Preconditioned iterative methods for nonselfadjoint or indefinite elliptic boundary value problems*, in Unification of Finite Elements, H. Kardestuncer, ed., Elsevier Science Publishers B. B., North-Holland (1984) 167–184.

[5] J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp., 47 (1986) 103–134.

[6] J. H. Bramble, J. E. Pasciak, J. Wang and J. Xu, *Convergence estimates for product iterative methods with applications to domain decomposition and multigrid*, Math. Comp., 57 (1991) 1–22.

[7] X.- C. Cai, *Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations*, Ph.D. thesis, Courant Institute, Sept. (1989).

[8] X.- C. Cai, W. D. Gropp and D. E. Keyes, *A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs and R. G. Voigt, eds., SIAM, Philadelphia (1992).

[9] X.- C. Cai, W. D. Gropp and D. E. Keyes, *Convergence rate estimate for a domain decomposition method*, Numer. Math., 61 (1992) 153–169.

[10] X.- C. Cai and O. B. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Stat. Comp., 13 (1992) 243–258.

[11] X.- C. Cai and O. B. Widlund, *Multiplicative Schwarz algorithms for nonsymmetric and indefinite problems* (1992) (SIAM J. Numer Anal., to appear).

[12] 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. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia (1990).

[13] M. Dryja, *An additive Schwarz algorithm for two- and three- dimension finite element elliptic problems,* in Domain Decomposition Methods for Partial Differential Equations II, T. F. Chan, R. Glowinski, G. A. Meurant, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia (1989).

[14] M. Dryja and O. B. Widlund, *An additive variant of the alternating method for the case of many subregions,* TR 339, Dept. of Comp. Sci., Courant Institute (1987).

[15] 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. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia (1990).

[16] M. Dryja and O. B. Widlund, *Multilevel additive methods for elliptic finite element problems*, in Parallel Algorithms for Partial Differential Equations, Proceedings of the Sixth GAMM-Seminar, Kiel, Jan. 19–21, W. Hackbusch, ed., Braunschweig, Germany (1991), Vieweg & Son.

[17] S. C. Eisenstat, H. C. Elman and M. H. Schultz, *Variational iterative methods for nonsymmetric system of linear equations*, SIAM J. Numer. Anal., 20 (1983) 345–357.

[18] H. C. Elman, *A stability analysis of incomplete LU factorizations*, Math. Comp., 47 (1986) 191–217.

[19] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991) 315–339.

[20] A. Greenbaum, C. Li and Z. Han, *Parallelizing preconditioned conjugate gradient algorithms*, Comp. Phys. Comm., 53 (1989) 295–309.

[21] W. D. Gropp and D. E. Keyes, *Domain decomposition on parallel computers*, Impact of Comp. in Sci. and Eng., 1 (1989) 421–439.

[22] W. D. Gropp and D. E. Keyes, *Domain decomposition with local mesh refinement*, SIAM J. Sci. Stat. Comp., 13 (1992) 967–993.

[23] W. D. Gropp and D. E. Keyes, *Parallel performance of domain-decomposed preconditioned Krylov methods for PDEs with adaptive refinement*, SIAM J. Sci. Stat. Comp., 13 (1992) 128–145.

[24] C. Johnson, U. Nävert and J. Pitkäranta, *Finite element methods for linear hyperbolic problems*, Comp. Meth. Appl. Mech. Eng., 45 (1984) 285–312.

[25] 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) s166–s202.

[26] 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).

[27] 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).

[28] T. A. Manteuffel and S. V. Parter, *Preconditioning and boundary conditions*, SIAM J. Numer. Anal., 27 (1990) 656–694.

[29] J. A. Meijerink and H. A. Van der Vorst, *Guidelines for the usage of incomplete decompositions in the solving sets of linear equations as they occur in practical problems*, J. Comp. Phys., 44 (1981) 134–155.

[30] S. V. Nepomnyaschikh, *Domain decomposition and Schwarz methods in a subspace for the approximate solution of elliptic boundary value problems*, Ph. D. Thesis, Computing Center of the Siberian Branch of the USSR Academy of Sciences (1986).

[31] S. V. Parter and S.-P. Wong, *Preconditioning second order elliptic operators: condition numbers and the distribution of the singular values*, J. of Sci. Comp., 6 (1991) 129–157.

[32] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986) 865–869.

[33] B. F. Smith, *A parallel implementation of an iterative substructuring algorithm for problems in three dimensions*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs and R. G. Voigt, eds., SIAM, Philadelphia (1992).

[34] H. A. Van der Vorst, *Bi-CGSTAB: A more smoothly converging variant of CG-S for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 13 (1992) 631–644.

[35] J. W. Watts, III, *A conjugate gradient-truncated direct method for the iterative solution of the reservoir simulation equation*, Soc. of Petroleum Eng. J., 21 (1981) 345–353.

[36] O. B. Widlund, *Some Schwarz methods for symmetric and nonsymmetric elliptic problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs and R. G. Voigt, eds., SIAM, Philadelphia (1992).

[37] J. Xu, *A new class of iterative methods for nonselfadjoint or indefinite problems*, SIAM J. Numer. Anal., 29 (1992) 303–319.

[38] J. Xu and X. - C. Cai, *A preconditioned GMRES method for nonsymmetric or indefinite problems*, Math. Comp. (1992) (to appear).

[39] X. Zhang, *Multilevel additive Schwarz methods*, CS-TR-2894 (UMIACS-TR-92-52), Dept. of Comp. Sci., University of Maryland (1992), to appear in Numer. Math.