CHAPTER 12

Domain-Decomposable Preconditioners for Second-Order Upwind Discretizations of Multicomponent Systems

David E. Keyes*
William D. Gropp†

Abstract. Discrete systems arising in computational fluid dynamics applications often require wide stencils adapted to the local convective direction in order to accommodate higher-order upwind differencing, and involve multiple components perhaps coupling strongly at each point. Conventional exactly or approximately factored inverses of such operators are burdensome to apply globally, especially in problems complicated by non-tensor-product domain geometry or adaptive refinement, though their "forward" action is not. Such problems can be solved by iterative methods by using either point-block preconditioners or combination space-decoupled/component-decoupled preconditioners that are based on lower-order discretizations. Except for a global implicit solve on a coarse grid, each phase in the application of such preconditioners has simple locally exploitable structure.

1. Introduction. Computational fluid dynamics has spawned a vast literature whose spectrum ranges from elegant analyses of model systems to intricately detailed analyses of realistic engineering or physical systems whose executions require hundreds of hours of supercomputer time. Of course, these categories are not mutually exclusive. However, the source of efficiency and optimality is often — almost by definition — the exploitation of special structure, with the result that the gap in attainable computational performance on ideal and state-of-the-art practical CFD problems has little prospect for closing completely. In fact, since it is usually the case that problems with less uniform structure are harder to map efficiently onto multi-processors, parallel computing would appear only to widen the performance gap between the ideal and the real, while bringing improvements to both.

* 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 IBM Corporation, and by the 3M Company.
† 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.

The combination of domain decomposition with preconditioned iterative methods extends the usefulness of numerical techniques for certain special partial differential equation problems to those of more general structure. The domains of problems with features inhibiting the global exploitation of optimal algorithms can often be decomposed into smaller subdomains of simpler structure on which extant solvers serve as local components of a parallelizable global approximate inverse. The computational advantages are usually sufficient to allow for the iteration required to enforce consistency at the artificially introduced subdomain boundaries, often even apart from parallelism. Size alone is often a sufficient advantage, when it is considered that the computational complexity of many solution algorithms is a superlinear function of the discrete dimension, and thus p problems of size $\frac{n}{p}$ may be solved more cheaply than one of size n.

Iterative methods based on choosing the best solution in incrementally expandable subspaces allow the tailoring of computations to specified accuracy requirements. These methods can use multiple representations of the same underlying operator, ultimately converging in terms of a desired "high-quality" representation through a series of applications of the inverse of a "lower-quality" representation, called a preconditioner, that is cheap or parallelizable or possesses some other advantage. Though already useful in linear problems and on serial computers, the ability to operate with multiple representations of the operator proves even more significant in nonlinear problems and in parallel. Domain decomposition can be regarded as a way of creating parallelizable preconditioners for iterative methods. The iteration required to piece together the solution at the artificial subdomain boundaries may be folded in with the iteration between the multiple levels of operator representation and, possibly, an outer nonlinear iteration as well.

Domain decomposition is a natural basis for partitioning programs across processors and partitioning data across memories, and allows a natural integration of local refinement, including refinements of mesh, of discretization order, or even of operator and the representation of the unknown fields. Though domain decomposition is as old as the analysis of engineering systems, the past decade has provided a significant theoretical foundation for model problems which has, in turn, provided heuristics for others. An aspect of interest to us is the migration in problem parameter space from the theoretically richly endowed "point" of the linear, selfadjoint problem for a scalar equation on a (quasi-)uniformly refined grid to the region of nonlinear multicomponent problems spawning a sequence of non-selfadjoint adaptively refined systems. Furthermore, we are interested in formulating such problems in a modular manner convenient to the design and maintenance of parallel software. For reasons of flexibility and inertia in the modeling of chemically reacting flows, in particular, we are primarily interested in finite difference or finite volume discretizations, but without relying on first-order methods since they are almost never competitive when the measure is fewest operations for a fixed accuracy.

This contribution focuses on movement along two "axes" of the just-mentioned problem parameter space: from first- to second-order upwinding, and from one to several unknown fields. After a summary of our evolving algorithmic philosophy in Section 2, we treat these topics in Sections 3 and 4, respectively, offering some numerical examples. The concluding remarks make brief links to the nonlinear parallel context.

2. Algorithmic Philosophy. The domains of dependence of resolvents of elliptic operators, such as the spatial terms of the momentum and energy equations of (subsonic) fluid mechanics, are global, but there is a decay with the distance between the source and field points. The decay motivates decomposition by domain, in contrast to decomposition by some less physical criterion, such as a purely graph-theoretic approach to the corresponding matrix operator, when searching for threads of computation that can usefully be executed independently. The global dependence implies that data must still travel across the grid from each point to all others in the course of the solution process. This requires a number of local data exchanges approximately equal to the discrete diameter of the grid or, possibly, a smaller number of longer-range exchanges derived from the use of multiple spatial scales.

2.1. Global Data Transport. Now "classical" results quantifying the trade-offs between purely local and global data transport are given in [2] and [6]. These papers show how preconditioned conjugate gradient iteration may be used to obtain solutions to self-adjoint elliptic problems in a number of iterations at most weakly dependent on the fine grid resolution through the logarithm of the ratio of the diameter of subdomains into which the global domain is divided, H, to the fine mesh parameter, h. The cost in each case is the iterated solution of a subdomain vertex problem equivalent to a coarse discretization of the original operator with the subdomains as elements, along with the solution of the independent subdomain problems themselves. Thus, the preconditioner is two-scale and requires non-nearest neighbor data exchanges. If the subdomain vertex solve in the preconditioner is replaced with a weighted identity operator, which requires no data exchanges, the bound on the iteration count rises in proportion to the discrete diameter of the coarse grid of vertices. If the subdomain solves themselves are likewise replaced with a simple diagonal scaling, the iteration count is, of course, proportional to the discrete diameter of the fine grid.

These results for selfadjoint operators have been extended in [3] and [4] to non-selfadjoint operators. Conjugate gradient iteration is replaced with the Generalized Minimum Residual (GMRES) method, and the bounds worsen by one or more powers of the factor $(1 + \log(H/h))$. It is required in the proofs that the coarse grid be sufficiently fine; in particular, a subdomain Reynolds number must be bounded. (A similar bound is required in convergence proofs for multigrid extensions to non-selfadjoint problems.) Bastardized predecessors of the non-selfadjoint form of the algorithm have been described for a scalar PDE in [9] and [10]. These algorithms require more iterations of cheaper preconditioners and are roughly as effective (measured in execution time) as those possessing optimal convergence rates until H and h take on small values.

Two-scale preconditioners significantly alleviate the parallel bottleneck of global preconditioners, whether the subdomains are assigned indivisibly to processors (as in our current codes), or whether the uniform tasks they represent are further subdivided in SIMD fashion. However, truly massive parallelism may require a richer hierarchy of scales.

2.2. Preconditioned Krylov Iteration. Current algorithms for nonsymmetric problems involve nearly parallel preconditioners in conjunction with the Krylov iterative method GMRES, described algorithmically in [15] and analyzed theoretically in [8] and [7] (in the equivalent form of the generalized conjugate residual method). Each iteration of GMRES involves a matrix-vector multiply, which for elliptic problems requires local data exchanges only, and a preconditioner solve of variable complexity. GMRES converges in a number of iterations proportional to the number of distinct clusters of one or more eigenvalues of the preconditioned operator. Loosely speaking, the greater the accuracy required, or the closer the cluster to the origin, the smaller the tolerance on what constitutes a single "cluster." Efficient use of GMRES in elliptic problems generally requires preconditioning to produce clustering. The new nonsymmetric solver Bi-CGSTAB [16] has been found to function as a reasonable replacement for GMRES in some recent comparisons.

In two dimensions, the preconditioner involves three phases: a global coarse grid solve, independent solves along interfaces between subdomains, and independent solves in the subdomain interiors. Because the phases must be synchronized, it is important to balance the load in each. This is, in principle, not difficult whenever the number of subdomains sufficiently exceeds the number of available processors, since work estimates for each phase on each subdomain are easy to obtain [10].

2.3. Domain-Decomposed Adaptations. Geometric complexity is relatively straightforward to accommodate within the domain decomposition framework, this being perhaps the original motivation for the technique. The matrix-vector multiply is a sparse operation requiring only nearest-neighbor exchanges between the subdomains. The same is true of the preconditioner, except for the global vertex solve. The latter is rendered simpler if a tensor product tessellation of the domain into subdomains is observed. Irregular and multiply-connected domains are permissible; see [9] for examples. Local transformations from body-fitted coordinates allow preservation of tensor-product structure in highly irregular regions.

In many cases, the problems generating the discrete systems have several different physical length scales, requiring some form of mesh refinement in order to produce an accurate solution. Locally uniform mesh refinement [9] is an adaptive resolution technique that is well suited to domain decomposition. By it, rectangular subdomains are refined with locally computationally regular tensor-product meshes. This refinement permits easy and efficient vectorization and allows consideration of fast solvers as components of domain-decomposed preconditioners. Different subdomains may have different mesh refinement, but the refinement is of a uniform scale within a single subdomain. This regularity allows a concisely expressed and flexible algorithm. Changes in grid refinement at interfaces between subdomains are accommodated with mutually overlapping phantom points and biquadratic interpolation. The phantom points allow the use of conventional finite-difference techniques (for second-order differential operators) in generating the difference equations at the subdomain interfaces. The selection of general refinement criteria is well examined in the literature and beyond the scope of the present contribution; in the specific examples presented, a good refinement strategy is fairly obvious.

To summarize, a general framework for iterative domain decomposition methods consists of

- a global operator, A;
- a global approximate inverse, B^{-1};
- an iterative method requiring only the action of A and B^{-1}; and
- a geometry-based, contiguity-preserving partition of unknowns inducing a block structure on A and B.

In the present contribution we dust off and incorporate two ideas from the literature of stationary iterative methods: (1) using a pair (A, B) in which B is of lower-order (defect correction), and (2) using B's derived from alternately blocked splittings of A (operator splitting). To avoid a proliferation of block-matrix notation, we discuss only the single domain case. We claim straightforward parallel multidomain extensions by applying these ideas within blocks, and illustrate them numerically with decompositions into boxes and strips.

3. Second-Order Upwinding. We have found an accelerated version of the defect correction paradigm to be useful in maintaining second-order accuracy in a CFD finite-difference discretization while employing only the more convenient first-order upwind differencing for the convective terms in the preconditioner.

A conventional defect correction method for solving the (possibly nonlinear) system of equations

$$N(u) = 0, \tag{1}$$

where N depends continuously on u, is as follows. We suppose that we can easily solve a related problem

$$\bar{N}(u) = f.$$

Then we initialize u by solving

$$\bar{N}(u^0) = 0,$$

and iterate:

$$\bar{N}(u^{k+1}) = \bar{N}(u^k) - N(u^k).$$

If the iterations converge, they converge to a solution of (1). In our context, $\bar{N}$ is simply a lower-order discretization of N.

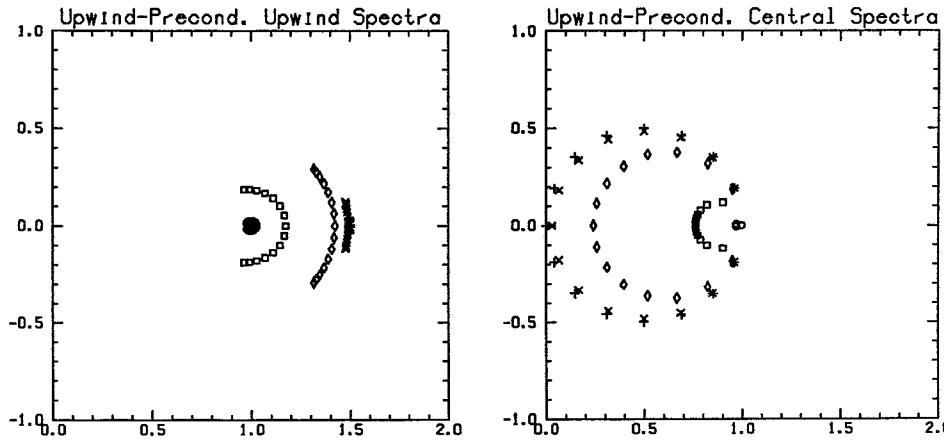

Figure 1. Spectra in the complex plane of preconditioned discrete systems AB^{-1} derived from Equation (2), for two different A's and values of c ranging from 10^{-1} (dots clustered at the point 1) to 10^4 ('+' symbols), the other symbols tracing intermediate orders of magnitude.

3.1. Spectrum of a Model Problem. Though there is no advantage to this technique in one dimension, we first illustrate for this case. Consider the scalar convection-diffusion problem

$$-\frac{\partial^2 u}{\partial x^2} + c\frac{\partial u}{\partial x} = f, \; c > 0. \tag{2}$$

A second-order upwind discretization on a uniform grid with mesh interval h is $Au = h^2 f$, where u and f now represent vectors of nodal values, and A is the banded matrix (modulo boundary effects)

$$A = \text{tri}(-1, 2, -1) + c \cdot h \cdot \text{pent}(\frac{1}{2}, -2, \frac{3}{2}, 0, 0). \tag{3}$$

For a preconditioner, we use (again, modulo boundary effects)

$$B = \text{tri}(-1, 2, -1) + c \cdot h \cdot \text{tri}(-1, 1, 0). \tag{4}$$

In dimension d, B represents either a savings of nearly half (to be precise, $\frac{2d+1}{4d+1}$) in storage, relative to A, if all possible nonzero entries are to be allocated at compile time, or a savings of d conditional evaluations per grid point if a minimum storage scheme based on knowledge of the direction of the velocity field is employed instead. The spectrum of AB^{-1} clusters well as $c \to 0$ or $c \to \infty$ and is reasonably compact for intermediate values. The inset box in Figure 1 contains all the eigenvalues for the Dirichlet problem on the unit interval for the case $h^{-1} = 16$.

Not all imaginable (A, B) pairs are recommended in the context of accelerated defect correction. For instance, if we keep (4) but use the more compact

$$A = \text{tri}(-1, 2, -1) + c \cdot h \cdot \text{tri}(-\frac{1}{2}, 0, \frac{1}{2}), \tag{5}$$

in place of (3), the outer rings of eigenvalues in Figure 1 result for the preconditioned operator AB^{-1}. For the periodic case, it can be shown that this spectrum spreads along the unit disk centered at 1 as $c \to \infty$, a very poor preconditioning for such a simple problem.

Table 1. Resolution parameter h^{-1}, 2-norm of the error e, iteration count I, and execution time T (sec) on a SPARCstation 1 for the solution of Equation (6). Convergence was defined as a reduction in the initial residual by 10^{-5}, except 10^{-6} as asterisked (necessary to achieve truncation error potential). The first-order computations incorporate an optimization that is not possible in the second-order cases, viz., large blocks of the preconditioned operator collapse to the identity matrix when the subdomains of A and B are identically discretized.

		1st Order			2nd Order		
c	h^{-1}	$\|\|e\|\|_2$	I	T	$\|\|e\|\|_2$	I	T
0.5	16	3.0(-3)	6	0.2	4.3(-3)	8	0.2
	32	2.0(-3)	12	0.8	1.4(-3)	14	1.2
	64	1.2(-3)	13	3.4	4.5(-4)	15	5.0
	128	6.9(-4)	16*	17.0	1.4(-4)	18*	24.5
2.0	16	4.9(-2)	6	0.1	1.7(-2)	9	0.2
	32	2.7(-2)	13	1.0	3.3(-3)	14	1.2
	64	1.4(-2)	13	3.4	6.8(-4)	19*	6.6
	128	7.1(-3)	10	11.0	1.7(-4)	16*	21.5
8.0	16	2.7(-1)	4	0.1	1.9(-1)	7	0.2
	32	1.5(-1)	13	1.0	5.8(-2)	16	1.4
	64	8.1(-2)	15	3.9	1.4(-2)	17	5.8
	128	4.2(-2)	14	15.0	3.5(-3)	16	21.9

3.2. A Two-Dimensional Example. To illustrate the technique in two dimensions for a non-constant velocity field, consider Problem 10 from [9], set on an L-shaped domain consisting of the square with vertices at ± 1 with the unit square deleted. The convection-diffusion operator in cylindrical coordinates

$$-\nabla^2 u + \frac{c}{r}\frac{\partial u}{\partial r} = 0 \tag{6}$$

models radial outflow from the origin when $c > 0$. An exact solution for any c is

$$u = r^{\alpha} \sin\left[\frac{2}{3}(\theta - \frac{\pi}{2})\right],$$

where $\alpha = \frac{1}{2}\left[c + \sqrt{c^2 + \frac{16}{9}}\right]$, so there is a nondifferentiable singularity at the origin when $c < 5/9$. In Table 1, we study this problem numerically for $c = 1/2$, 2, and 8 at different refinements. A domain-decomposed construction of A and B was used, as in [9], where the subdomains consisted of small squares of 8 subintervals on a side. The $h^{-1} = 16$ cases were discretized with just three subdomains, with a common vertex at the origin. For the $h^{-1} = 32$ cases, these three squares were subdivided into four squares each, so twelve 8×8 subdomains were employed. Finally, the $h^{-1} = 128$ cases required 192 8×8 subdomains, with a large number of internal vertices making up the global coarse grid system.

The column headers "1st Order" and "2nd Order" pertain to the order of discretization used in the A operator. A first-order B was employed in both sets of cases. We note first that the asymptotic error improvements follow the formal orders of accuracy closely; that is, the error is halved with each doubling of grid resolution for first order, and it is quartered for second order. For the $c = 0.5$ rows, we do not quite achieve these error reduction ratios, since the solution is nondifferentiable at the origin. The orders verified, we compare execution times T for comparable errors. Looking at $c = 2$, we note that the 3.4 sec required to obtain an error of 1.4×10^{-2} at $h^{-1} = 64$ for a first-order scheme compares with the 0.2 sec required to obtain an error of 1.7×10^{-2} at $h^{-1} = 16$ for a second-order scheme; thus the savings of using mixed orders in the A and B

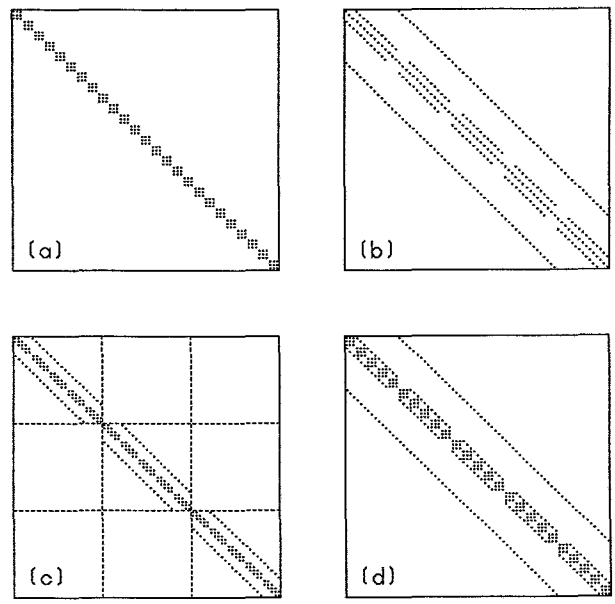

Figure 2. Sparsity plots of a C-D-R operator and some of its summands: (a) R, (b) S, (c) P^TSP, and (d) A. For clarity, r (= 3) and n (= 5) are small.

operators can be more than an order of magnitude in execution time, and the iteration counts to comparable accuracy are themselves comparable.

4. Multicomponent Systems. Multicomponent problems provide an excellent example of how preconditioners can be constructed from substructuring. In this case, the substructuring is not spatial, but derives from alternate blockings of the original discrete operator. In one blocking, the degrees of freedom defined at the same grid point are ordered most rapidly, and the small dense diagonal blocks represent coupling between various equations. In the other, the degrees of freedom corresponding to the same field are ordered most rapidly, and the large sparse diagonal blocks represent convection-diffusion processes within one equation. Neglecting all but the diagonal blocks in either factorization represents a physical extreme that greatly simplifies preconditioner construction and application. We consider whether the resulting preconditioners have any value, and how they might be employed in combination.

A generic convection-diffusion-reaction (C-D-R) system can be written

$$\vec{c}(\vec{u}) \cdot \nabla u_k - \nabla \cdot D_k(\vec{u}) \nabla u_k = f_k(\vec{u}), \quad k = 1, \ldots, r. \tag{7}$$

The source/sink terms f_k often arise from reactions between the constituents u_k; hence, our nomenclature. The discrete operator A in the corresponding $Au = f$ can be decomposed as $A = R+S+T$. R, coming from the reaction terms, is block diagonal. On an $n \times n$ grid with r components per point it consists of n^2 $r \times r$ blocks (Figure 2a). In a standard five-point discretization, S (Figure 2b), coming from the convection and diffusion terms, can be permuted so that P^TSP (Figure 2c) is block diagonal, consisting of r sparse banded $n \times n$ matrices. T contains any left-over dependencies from the nonlinear dependence of $\vec{c}$ and the D_k upon the u_k. It vanishes in the linear case. When the only intercomponent coupling occurs through the f_k, A has the sparsity structure of Figure 2d. More generally, the off-diagonal blocks of Figure 2d can fill in.

Some practical choices that suggest themselves for preconditioning C-D-R systems will be denoted:

- $B_0^{-1} \equiv U^{-1}L^{-1}$, $A = LU + E$

- $B_1^{-1} \equiv [R + \text{block-diag}(S)]^{-1}$
- $B_2^{-1} \equiv U^{-1}L^{-1}$, $P^T[S + \text{diag}(R)]P = LU + E$

Here L and U represent the lower and upper factors of an incomplete LU factorization. B_0 is a "brute-force" block-ILU factorization of A itself. B_1 is an exact factorization of the elements of A that fall on the nonzeros in Figure 2a. B_2 is a set of independent point-ILU factorizations of the diagonal blocks of Figure 2c.

The computational complexities of applying these operators as needed in a preconditioned iterative method are as follows:

- Multiply by A: $r^2n^2 + 4rn^2$
- Solve with B_0^{-1}: $5r^2n^2$
- Solve with B_1^{-1}: r^2n^2
- Solve with B_2^{-1}: $5rn^2$

In a computational reacting flow problem, where r can easily be $O(10)$, the savings of B_1^{-1} and B_2^{-1} over B_0^{-1} are potentially appreciable. We therefore tried compound preconditioners such as

$$\tilde{B} = B_1^{-1} + B_2^{-1} - B_2^{-1}AB_1^{-1}$$

(which can be implemented with one call to B_1^{-1} and two to B_2^{-1}). Note that if $AB_1^{-1} = I + \Delta$ and $AB_2^{-1} = I + \Sigma$, where the smallness of Δ and Σ measure the quality of the preconditioners, then $A\tilde{B} = I - \Sigma\Delta$. However, in the representative examples below, neither AB_1^{-1} nor AB_2^{-1} is sufficiently close to the identity for the higher-order gains motivating the compound preconditioner to be realized, and $A\tilde{B}$ is worse conditioned than A itself.

Nevertheless,, two practical schemes can be built from the components B_1^{-1} and B_2^{-1} when one is used as an initial scaling, yielding a transformed A for the other. (Such a use is called "pre-preconditioning" in [14].) To solve $Au = f$ when the intercomponent coupling is relatively weak, one may prescale to get $\tilde{A} = AB_1^{-1}$, and permute to get $\bar{A} = P^T\tilde{A}P$ and $\bar{f} = P^Tf$. Then $\bar{A}v = \bar{f}$ may be solved by block-Gauss-Seidel iterations. Each block is a scalar convection-diffusion problem, and may be solved by, for instance, GMRES using a point-ILU preconditioning as in B_2^{-1}. Finally, the permutation-scaling is unwound to get $u = B_1^{-1}Pv$. This idea is taken from [1], where it was denoted an Alternate-Block-Factorization (ABF) procedure. For moderate coupling one may again prescale to get $\tilde{A} = AB_1^{-1}$. Then, GMRES may be applied to $(\tilde{A}B_2^{-1})v = f$, where B_2 is based only on the elements of $\tilde{A}$ falling on the nonzeros of Figure 2b. (Figure 2c shows how such a preconditioner can be computed efficiently, though the data can actually remain "in place" in implementation.) Finally, the preconditioning is unwound with $w = B_2^{-1}v$, and the scaling is unwound with $u = B_1^{-1}w$. For strong coupling, one can always use GMRES on $(AB_0^{-1})v = f$ and set $u = B_0^{-1}v$.

4.1. A Singly-Parameterized Model. We consider two examples with variable degrees of coupling. The first is a model diffusion-reaction system which is motivated by (but not directly related to) an electrostatics problem in semi-conductor physics, in which the first component represents the electrostatic potential and the other two represent electron and hole concentrations, respectively:

$$\begin{array}{cccc} -\nabla^2 u_1 & +cu_2 & -cu_3 & = f_1 \\ -cu_1 & -\nabla^2 u_2 & & = f_2 \\ cu_1 & & -\nabla^2 u_3 & = f_3 \end{array}$$

The parameter c controls the strength of the coupling. As $|c| \to 0$, three independent Poisson problems are obtained. As $|c| \to \infty$, n^2 independent equilibrium systems result. At all other c, a coupled D-R system of size $3n^2$ remains to be solved.

Some results are shown in Table 2, where c takes on some extreme values. Focusing on the $p = 1$ rows, we note that all methods are good for weak coupling, and the weaker methods execute more efficiently in this regime. However, the block-ILU method is most successful at strong coupling and appears to have its greatest trouble, in absolute terms, in the intermediate regime, in which neither

Table 2. Iteration counts I and parallel execution times T on the Encore Multimax for the diffusion-reaction system under three algorithms and three decompositions p into either 1 or 8 strips residing on different processors, for three different intensities of coupling between the components c, for reduction in the initial residual of 10^{-5}. Iteration counts for the ABF algorithm are of the outer Gauss-Seidel iterations only (the inner GMRES iterations are not shown). The first in each set of $p = 8$ problems has decoupled subdomains. The asterisked second member in each set has subdomains coupled by the IP(0) technique [5,12].

		Block-ILU		Prescaled		ABF	
c	p	I_{GMRES}	T	I_{GMRES}	T	I_{GS}	T
10^{-6}	1	13	51.6	13	31.9	3	28.9
	8	24	13.6	24	9.5	3	8.8
	8*	16	8.8	15	5.4	3	8.4
1	1	18	72.9	17	44.0	4	34.2
	8	32	18.9	32	13.6	4	10.0
	8*	21	11.7	20	7.6	4	9.0
10^{6}	1	13	51.0	34	107.0	> 50	-
	8	24	13.6	36	16.0	> 50	-
	8*	15	8.2	34	15.1	> 50	-

S nor R can be neglected relative to the other. Comparing the three rows within each set gives an indication of their parallelizability on a coarse-grained multiprocessor. Though parallelizability is not the focus of this paper, the example is new, so we note that the form of the preconditioner that retains some of the subdomain coupling is more effective than the decoupled form, especially for the block-ILU subdomain preconditioning. Note that speedups of about 6 are obtained on 8 processors when the subdomains are coupled, under the Block-ILU and prescaled preconditioners.

4.2. A Reacting Flow Problem. Our other example is a three-component flamesheet model using the streamfunction-vorticity formulation for the fluid flow discretized with a nine-point stencil on a stretched tensor-product grid. The Block-ILU results in Table 3 are repeated from [13], but the prescaled and ABF results are new. The initial column lists four different Jacobian matrices drawn from successive stages of a Newton algorithm for solving the nonlinear system. Roughly speaking, the intercomponent coupling becomes more dominant in going from J31-A to J31-B to J31-C as a time step parameter is increased, and the grid resolution is doubled in going to J63-A.

As in Table 2, the weaker methods are more effective with weak coupling, but the ABF method actually fails to converge in the hardest problem. However, the nonlinear solution process spends considerable time working with Jacobians similar to J31-A, so the performance differential is worth capitalizing on in practice. Unlike the Table 2 results, decoupling the subdomains is not always worse than coupling them, a situation that reflects the dominance of the intercomponent coupling over the spatial coupling in this problem. Parallel speedups vary considerably.

5. Concluding Remarks. Summarizing Sections 3 and 4, we note that second-order convection can be accommodated without strongly degrading the diagonal dominance of any operators whose inverse action is required, and that for sufficiently weak coupling, preconditioners for multicomponent problems are available that are less expensive than full block-ILU. A variety of component-adaptive options exist for the spatial preconditioning in such cases. This pair of optimizations has not yet been combined; however, the first step in this direction has been taken: the accelerated defect-correction has been successfully used in multicomponent problems with exact subdomain solves, in work to be reported elsewhere.

Implicit nonlinear problems handled by fixed-point or Newton algorithms present no difficulties to an iterative domain decomposition algorithm not already present in the linear case, except the

Table 3. Iteration counts I and execution times T on the Encore Multimax for four Jacobians from the Flamesheet problem, presented in the same format as Table 2.

		Block-ILU		Prescaled		ABF	
Matrix	p	I_{GMRES}	T	I_{GMRES}	T	I_{GS}	T
J31-A	1	6	24.9	6	14.1	4	15.0
	8	9	4.9	9	3.0	4	3.2
	8*	5	3.0	12	4.2	4	3.9
J31-B	1	14	56.0	27	76.2	44	231.5
	8	20	11.2	32	13.6	39	38.5
	8*	21	11.9	39	18.1	46	63.6
J31-C	1	14	56.0	32	96.3	> 50	-
	8	22	12.6	38	17.1	> 50	-
	8*	30	18.2	44	21.5	> 50	-
J63-A	1	18	307.0	> 50	-	∞	-
	8	19	45.9	> 50	-	∞	-
	8*	17	39.0	> 50	-	∞	-

usual one of computational steering. (Though damping, continuation, and Jacobian re-evaluation strategies are crucial, they require insignificant computational cycles compared to solving the linear systems.) No alternate layout of data on a multiprocessor is necessary. The construction of Jacobian matrices, approximated by finite differences of residual vectors, and the convergence checking by Euclidean norms of residuals or solution update vectors require the same type of data exchanges already present in the GMRES algorithm, and the linear and nonlinear phases run at comparable efficiency [11]. Thus, a domain-decomposed parallel implementation of the nonlinear problem is no more challenging than the union of serial nonlinear and domain-decomposed linear implementations.

Our domain decomposition approach raises a number of questions relating specifically to parallel implementations which we have begun to address in [11] and references therein. We mention here only that the question of appropriate granularity of the decomposition is a nontrivial one and goes beyond the convergence rate issue mentioned in Section 2. Other criteria that collide in this issue are the "surface-to-volume" ratio of the subdomains (in conjunction with the communication-to-computation ratio of the hardware), the geometric refinement requirements of the underlying PDE (which are often solution-dependent in nonlinear problems), and the processor load balancing. Inevitably, the optimal parameterization of domain decomposition algorithms will retain some problem and machine dependence.

REFERENCES

[1] R. E. Bank, T. F. Chan, W. M. Coughran, Jr., and R. K. Smith, The Alternate-Block-Factorization Procedure for Systems of Partial Differential Equations, AT&T Bell Laboratories Numer. Anal. Ms. 89-5, 1989.

[2] 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.

[3] X.-C. Cai, An Additive Schwarz Algorithm for Nonselfadjoint Elliptic Equations, 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, pp. 232-244.

[4] X.-C. Cai, W. D. Gropp, and D. E. Keyes, Convergence Rate Estimate for a Domain Decomposition Method, 1990. (Manuscript)

[5] T. F. Chan, Boundary Probe Domain Decomposition Preconditioners for Fourth Order Prob-

lems, in *Domain Decomposition Methods*, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia, 1989, pp. 160-167.

[6] M. Dryja and O. B. Widlund, An Additive Variant of the Schwarz Alternating Method for the Case of Many Subregions, NYU, Courant Institute TR 339, December 1987.

[7] 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), pp. 345-357.

[8] H. C. Elman, Y. Saad, and P. E. Saylor, A Hybrid Chebyshev-Krylov Subspace Algorithm for Solving Nonsymmetric Systems of Linear Equations, Yale Univ., Dept. of Comp. Sci., RR-301, February 1984.

[9] W. D. Gropp and D. E. Keyes, Domain Decomposition with Local Mesh Refinement, Yale Univ., Dept. of Comp. Sci., RR-726, Aug. 1989.

[10] W. D. Gropp and D. E. Keyes, Parallel Performance of Domain-Decomposed Preconditioned Krylov Methods for PDEs with Adaptive Refinement, Yale Univ., Dept. of Comp. Sci., RR-773, March 1990.

[11] W. D. Gropp and D. E. Keyes, Parallel Domain Decomposition and the Solution of Nonlinear Systems of Equations, *these proceedings.*

[12] D. E. Keyes and W. D. Gropp, Domain Decomposition Techniques for Nonsymmetric Systems of Equations: Examples from Computational Fluid Dynamics, in *Domain Decomposition Methods*, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia, 1989, pp. 321-339.

[13] D. E. Keyes and W. D. Gropp, Domain Decomposition Techniques for the Parallel Solution of Nonsymmetric Systems of Elliptic Boundary Value Problems, Appl. Num. Math., 6 (1989/90), pp. 281-301.

[14] F. Shakib, T. J. R. Hughes, and Z. Johan, Element-by-Element Algorithms for Nonsymmetric Matrix Problems Arising in Fluids, in *Solution of Super Large Problems in Computational Mechanics*, J. H. Kane, A. D. Carlson, and D. L. Cox, eds., Plenum, New York, 1989, pp. 1-33.

[15] 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.

[16] H. A. Van der Vorst, Bi-CCSTAB: A More Smoothly Converging Variant of CG-S for the Solution of Nonsymmetric Linear Systems, 1990. (Manuscript)