10-1-1999

A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems

Xiao-Chuan Cai

Marcus Sarkis
Worcester Polytechnic Institute, msarkis@wpi.edu

TIMELY COMMUNICATION

A RESTRICTED ADDITIVE SCHWARZ PRECONDITIONER FOR GENERAL SPARSE LINEAR SYSTEMS*

XIAO-CHUAN CAI† AND MARCUS SARKIS‡

Abstract. We introduce some cheaper and faster variants of the classical additive Schwarz preconditioner (AS) for general sparse linear systems and show, by numerical examples, that the new methods are superior to AS in terms of both iteration counts and CPU time, as well as the communication cost when implemented on distributed memory computers. This is especially true for harder problems such as indefinite complex linear systems and systems of convection-diffusion equations from three-dimensional compressible flows. Both sequential and parallel results are reported.

Key words. overlapping domain decomposition, preconditioner, iterative method, sparse matrix

AMS subject classifications. 65N30, 65F10

PII. S106482759732678X

1. Introduction. In this paper, we introduce some modified overlapping additive Schwarz preconditioners for sparse linear systems. The original additive Schwarz method (AS) was introduced for solving symmetric positive definite elliptic finite element problems, and was later extended to many other nonsymmetric and nonelliptic systems; see, e.g., [7, 8, 16]. AS is available in several large parallel software libraries, such as PETSc [1] and P-SPARSLIB [14]. Here we propose a very simple change, and the resulting algorithm is more effective in terms of both iteration numbers and CPU time on sequential and parallel computers. We have tested the method, referred to as the restricted additive Schwarz method (RAS), for a wide range of problems including convection-diffusion equations, indefinite complex Helmholtz equations, and the 3D compressible Euler's equation discretized on unstructured meshes. We shall present the new methods as algebraic preconditioners for general sparse linear systems.

RAS was found accidentally. While working on an AS/GMRES algorithm in a Euler simulation, we removed part of the communication routine and surprisingly the "then AS" method converged faster both in terms of iteration counts and CPU time. We note that RAS is the default parallel preconditioner for nonsymmetric sparse linear systems in PETSc [1] and has been used in several applications [11, 13].

The paper is organized as follows. We devote section 2 to the description of RAS and other variants. Several case studies are given in section 3. We conclude the paper by a few remarks in section 4.

2. An RAS preconditioner. We consider a linear system

$$Ax = b, \tag{1}$$

where $A = (a_{ij})$ is an $n \times n$ nonsingular sparse matrix having a nonzero pattern that is symmetric. To describe the algebraic Schwarz algorithm, as in [6], we define a graph $G = (W, E)$, where the set of vertices $W = \{1, \ldots, n\}$ represents the n unknowns and the edge set $E = \{(i, j) \mid a_{i,j} \neq 0\}$ represents the pairs of vertices that are coupled by a nonzero element in A. Since we assume that the nonzero pattern is symmetric, the adjacency graph G is undirected. For multicomponent problems, a vertex in G often represents several unknowns in (1) that are associated with a single mesh point. We confine our discussion to the single component case and the extension to other cases is straightforward. For the remaining discussion, we will assume that the graph partitioning has been applied and has resulted in N nonoverlapping subsets W_i^0 whose union is W. We define the overlapping partition of W as follows. Let $\{W_i^1\}$ be the one-overlap partition of W, where $W_i^1 \supset W_i^0$ is obtained by including all the immediate neighboring vertices of the vertices in W_i^0. Using the idea recursively, we can define a δ-overlap partition of W,

$$W = \bigcup_{i=1}^{N} W_i^\delta,$$

where $W_i^\delta \supset W_i^0$ with δ levels of overlaps with its neighboring subdomains. Here $\delta \geq 0$ is an integer. Associated with each W_i^0 we define a restriction operator R_i^0. In matrix terms, R_i^0 is an $n \times n$ subidentity matrix whose diagonal elements are set to 1 if the corresponding node belongs to W_i^0 and to zero otherwise. Similarly we can define R_i^δ for each W_i^δ. With this we define the matrix

$$A_i = R_i^\delta A R_i^\delta.$$

Note that although A_i is not invertible, we can invert its restriction to the subspace

$$A_i^{-1} \equiv \left((A_i)_{|L_i} \right)^{-1},$$

where L_i is the vector space spanned by the set W_i^δ in $\mathbf{R}^n$. Recall that the regular AS preconditioner is defined as

$$M_{AS}^{-1} = \sum R_i^\delta A_i^{-1} R_i^\delta.$$

Our new RAS algorithm can be simply described as follows.

ALGORITHM 1 (RAS). *Solve the equation*

$$M_{RAS}^{-1} A x = M_{RAS}^{-1} b$$

by a Krylov subspace method, where the preconditioner M_{RAS}^{-1} is defined by

$$M_{RAS}^{-1} = R_1^0 A_1^{-1} R_1^\delta + \cdots + R_N^0 A_N^{-1} R_N^\delta.$$

REMARK 2.1. *Unless $\delta = 0$, M_{RAS}^{-1} is nonsymmetric even for symmetric A. For certain symmetric matrices A, for example, the five-point Poisson matrix, RAS/GMRES*

is more effective than AS/CG in terms of iteration numbers and CPU and communication times, although more memory is needed to save the Krylov vectors in GMRES. If $\delta = 0$, M_{RAS}^{-1} reduces to the usual block diagonal preconditioner.

REMARK 2.2. *A multiplicative version of RAS can be derived easily as in [6].*

REMARK 2.3. *In a parallel implementation, half of the communication cost can be saved because $R_i^0 x$ does not involve any data exchange with the neighboring processors. This is the main motivation for us to design this algorithm in replacing the AS method.*

REMARK 2.4. *There is another version of the preconditioner, referred to as the additive Schwarz with harmonic extension (ASH), defined as follows:*

$$M_{ASH}^{-1} = R_1^\delta A_1^{-1} R_1^0 + \cdots + R_N^\delta A_N^{-1} R_N^0.$$

It turns out RAS and ASH have similar behavior for all the numerical tests we report in the next section.

REMARK 2.5. *Both RAS and ASH can be symmetrized as $M_{RASH}^{-1} = R_1^0 A_1^{-1} R_1^0 + \cdots + R_N^0 A_N^{-1} R_N^0$. Our numerical tests show that M_{RASH}^{-1} is always weaker than M_{RAS}^{-1}.*

REMARK 2.6. *In practice, whenever possible, a coarse preconditioner is often incorporated either additively or multiplicatively to the Schwarz-type preconditioners to make the convergence rate independent of the number of subspaces. Details can be found in [7, 16].*

REMARK 2.7. *Some improvement can be obtained if the restriction operator is defined by using certain weights such that the sum of the operators is equal to the identity matrix. For example, we can define $R_{i,\omega}^\delta$ as an $n \times n$ diagonal matrix whose diagonal element is set to zero if the corresponding node does not belong to W_i^δ, and to $1/k$ if the node belongs to W_i^δ and $k - 1$ other subdomains. This weighted restriction operator can also be used to improve the classical AS. For example, a weighted AS can be defined as $M_{WAS}^{-1} = R_{1,\omega}^\delta A_1^{-1} R_1^\delta + \cdots + R_{N,\omega}^\delta A_N^{-1} R_N^\delta$.*

3. Case studies. We provide some numerical examples that compare RAS with the regular AS method. In all the test runs, we always use GMRES [15] as the accelerator even if the matrix is symmetric positive definite. We stop the iteration when the Euclidean norm of the preconditioned initial residual is reduced by a factor of 10^{-6}. GMRES restarts at 30 for all 2D test problems and at 5 for the 3D test problem. The number of subdomains is denoted by *subd*. A nonnested coarse space [2] is included in all the 2D tests.

The first test is for a 2D convection-diffusion problem $Lu = -\Delta u + b_1 u_x + b_2 u_y$ with zero Dirichlet boundary condition on the unit square. The equation is discretized with the usual five-point finite difference scheme with a first-order upwinding if $b_1 b_2 \neq 0$. A 128×128 fine mesh is partitioned into 16 and 64 subdomains. GMRES restarts at 30. The iteration numbers can be found in Table 1 for the Poisson case when $b_1 = b_2 = 0$ and for a more general case with $b_1 = 10, b_2 = 20$. RAS indeed takes a fewer number of iterations. For the Poisson problem, we plot all the eigenvalues, computed with MATLAB, of the preconditioned matrix in Figure 1. We note that some of the eigenvalues of the RAS preconditioned matrix are complex, and if we consider only the real part, the largest eigenvalue of the RAS preconditioned matrix is much smaller than the largest eigenvalue of the AS preconditioned matrix. The difference between the smallest eigenvalues is tiny. This is probably why RAS is faster than AS, although eigen information may not completely reflect the convergence of GMRES.

TABLE 1
Iteration counts for the Poisson and the convection-diffusion problems on a 128×128 mesh.

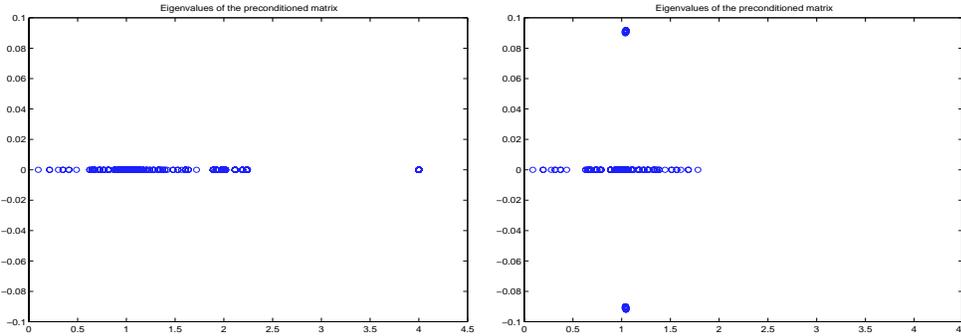|	$subd = 16$, coarse 4×4				$subd = 64$, coarse 8×8			
	$b_1 = b_2 = 0$		$b_1 = 10, b_2 = 20$		$b_1 = b_2 = 0$		$b_1 = 10, b_2 = 20$	
δ	AS	RAS	AS	RAS	AS	RAS	AS	RAS
1	20	17	23	20	20	18	22	20
2	18	14	20	17	18	15	19	17
3	16	13	19	15	16	13	18	15

FIG. 1. *The distribution of eigenvalues of the left-preconditioned matrix. The left figure is for AS and the right one is for RAS. The fine mesh is 64×64, which is partitioned into 16 subdomains. $\delta = 2$. There is no coarse grid.*

Our second test is for a two-dimensional Helmholtz equation with Sommerfeld boundary condition defined on the unit square, i.e.,

$$(2) \qquad \begin{cases} -\Delta u - k^2 u = f & \text{in } \Omega, \\[2mm] \dfrac{\partial u}{\partial n} - iku = 0 & \text{on } \partial\Omega, \end{cases}$$

where $i = \sqrt{-1}$ and $k > 0$ is a real constant. $\partial/\partial n$ is the outward normal derivative. We choose f so that the exact solution $u = \cos(k(x + y)) + i\sin(k(x + y))$. We use a standard h finite element discretization [3, 12]. The linear system is complex symmetric, but not Hermitian. A complex version of GMRES is used. The fine mesh is 128×128 and coarse mesh is 20×20. 16 and 64 subdomains are used here. The frequency parameter in the Helmholtz operator is $k = 10.0$. Table 2 has the iteration numbers. It is surprising that, mostly for AS, the number of iterations increases as δ becomes larger and also that RAS takes so many fewer iterations than AS.

Our third test is for the steady-state transonic ($M_\infty = 0.89$) three-dimensional compressible Euler's equation discretized on a fully unstructured mesh [5]. The system is discretized with a second-order finite volume method [10]. An inexact Newton's method is used to solve the nonlinear system. We report the test results for solving a linear system whose solution provides the inexact Newton direction. The unstructured mesh has 22012 nodes and 118480 tetrahedra, and it is partitioned by using a graph partitioning scheme provided in the TOP/DOMDEC package [9]. The number of unknowns of the linear system is 110060. Minimum overlap is used, i.e., $\delta = 1$ and CFL=400.0. No useful coarse space is available for this test problem. The 10^{-6} stopping condition may appear excessive for the simulation, but it is nevertheless useful to demonstrate the efficiency of the method. We note, however, that a similar

TABLE 2

The Helmholtz problem on a 128×128 mesh with Sommerfeld boundary condition.

	$subd = 16$		$subd = 64$	
δ	AS	RAS	AS	RAS
1	88	41	60	27
2	103	32	71	26
3	109	33	77	27

TABLE 3

Transonic Euler's equation on a $3D$ unstructured mesh. CPU and COMM are in seconds. The number of processors is the same as the number of subdomains subd. For all tests, $\delta = 1$.

	AS			RAS		
$subd$	Iter	CPU	COMM	Iter	CPU	COMM
4	131	24.05	1.10	100	19.31	0.53
8	140	14.80	1.26	105	11.50	0.69
16	145	9.00	1.44	106	6.78	0.90
32	152	7.67	2.45	110	5.77	1.17

behavior is obtained for using looser convergence tolerance and for unsteady Euler simulations. The CPU and communication times, for solving the whole problem, are obtained on an IBM SP machine. Parallel performance for up to 32 processors is reported in Table 3. Other interesting numerical results for using RAS in parallel 3D steady-state Euler calculations with up to 128 processors have also been reported in [11, 13].

4. Concluding remarks. We have introduced a cheaper and faster variant of the classical additive Schwarz method and tested it for a few challenging problems including indefinite Helmholtz equations in 2D and the compressible Euler equation on 3D unstructured meshes. All tests show that the new method is superior to the additive Schwarz method in terms of iteration counts, CPU time, and communication costs if implemented in parallel. For self-adjoint, elliptic finite element problems, a one-level convergence theory has been developed for the case when the domain is decomposed into stripes [4]. The general theory for nonsymmetric or indefinite problems, or for decompositions with cross points, is yet to be developed.

REFERENCES

[1] S. BALAY, W. GROPP, L. MCINNES, AND B. SMITH, *The Portable, Extensible Toolkit for Scientific Computing, version* 2.0.13, http://www.mcs.anl.gov/petsc, code and documentation, 1996.

[2] X.-C. CAI, *The use of pointwise interpolation in domain decomposition methods with nonnested meshes*, SIAM J. Sci. Comput., 16 (1995), pp. 250–256.

[3] X.-C. CAI, M. CASARIN, JR., F. ELLIOTT, JR., AND O. WIDLUND, *Overlapping Schwarz algorithms for solving Helmholtz's equation*, in The Tenth International Conference on Domain Decomposition Methods for Partial Differential Equations, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, Providence, RI, 1998, pp. 391–399.

[4] X.-C. CAI, M. DRYJA, AND M. SARKIS, *A convergence theory for restricted additive Schwarz methods*, Tech. report, Dept. of Computer Science, University of Colorado at Boulder, 1999, to appear.

[5] X.-C. Cai, C. Farhat, and M. Sarkis, *A minimum overlap restricted additive Schwarz precon-ditioner and applications in 3D flow simulations*, in The Tenth International Conference on Domain Decomposition Methods for Partial Differential Equations, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, Providence, RI, 1998, pp. 479–485.

[6] X.-C. Cai and Y. Saad, *Overlapping domain decomposition algorithms for general sparse matrices*, Numer. Linear Algebra Appl., 3 (1996), pp. 221–237.

[7] T. Chan and T. Mathew, *Domain decomposition algorithms*, Acta Numerica (1994), pp. 61–143.

[8] M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput., 15 (1994), pp. 604–620.

[9] C. Farhat, S. Lanteri, and H. Simon, *TOP/DOMDEC: A software tool for mesh partitioning and parallel processing and applications to CSM and CFD computations*, Comput. Sys. Engrg., 6 (1995), pp. 13-26.

[10] L. Fezoui and B. Stoufflet, *A class of implicit upwind schemes for Euler simulations with unstructured meshes*, J. Comput. Phys., 84 (1989), pp. 174–206.

[11] W. Gropp, D. Keyes, L. McInnes, and M. Tidriri, *Parallel implicit PDE computations: Algorithms and software*, Proceedings of Parallel CFD'97, A. Ecer, J. Periaux, and N. Satofuka, eds., Manchester, UK, 1997.

[12] F. Ihlenburg and I. Babuska, *Finite element solution of the Helmholtz equation with high wave number, Part* I: *The h-version of the FEM*, Comput. Math. Appl., 30 (1995), pp. 9–37.

[13] D. Keyes, D. Kaushik, and B. Smith, *Prospects for CFD on petaflops systems*, in CFD Review 1997, M. Hafez and K. Oshima, eds., World Scientific, Singapore, 1997.

[14] Y. Saad and A. Malevsky, *P-SPARSLIB: A portable library of distributed memory sparse iterative solvers*, (version 2.15, May 1997), Technical report UMSI 95-180, Minnesota Supercomputer Institute, Minneapolis, MN, 1995.

[15] Y. Saad and M. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[16] B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.