http://dx.doi.org/10.1090/conm/218/03046

Contemporary Mathematics
Volume **218**, 1998
B 0-8218-0988-1-03046-6

A Minimum Overlap Restricted Additive Schwarz Preconditioner and Applications in 3D Flow Simulations

Xiao-Chuan Cai, Charbel Farhat, and Marcus Sarkis

1. Introduction

Numerical simulations of unsteady three-dimensional compressible flow problems require the solution of large, sparse, nonlinear systems of equations arising from the discretization of Euler or Navier-Stokes equations on unstructured, possibly dynamic, meshes. In this paper we study a highly parallel, scalable, and robust nonlinear iterative method based on the Defect Correction method (DeC), the Krylov subspace method (Krylov), the minimum overlap restricted additive Schwarz method (RAS), and the incomplete LU factorization technique (ILU). To demonstrate the robustness of the method, we test the capability of the DeC-Krylov-RAS solver for several flow regimes including transonic and supersonic flows around an oscillating wing and a moving aircraft. The parallel scalability is also tested on a multiprocessor computer. We consider the unsteady 3D Euler's equation

$$\frac{\partial W}{\partial t} + div(F(W)) = 0 \tag{1}$$

with certain initial and boundary conditions. Unstructured mesh and variable time stepping are used in our numerical simulation, however, for the sake of simplicity in the discussion, we let Δt and h be the fixed time and spatial discretization parameters, and $\Phi_h^{(2nd)}$ a second order MUSCL discretization of $div(F(\cdot))$. A fully discretized scheme, which is of second order in both space and time, can be written as

$$\frac{3W_h^{n+1} - 4W_h^n + W_h^{n-1}}{2\Delta t} + \Phi_h^{(2nd)}(W_h^{n+1}) = 0. \tag{2}$$

Here n is a running time step index and W_h^0 is the given initial solution. Assuming that W_h^n and W_h^{n-1} are known, (2) is a large, sparse, nonlinear algebraic system of equations that has to be solved at every time step to a certain accuracy.

1991 *Mathematics Subject Classification.* Primary 65N55; Secondary 76D05, 65Y05.

Key words and phrases. Restricted additive Schwarz, 3D compressible flows, Unstructured meshes, Parallel performance.

This research was supported in part by NSF ECS-9725504 and AFOSR F49620-97-1-0059.

2. Discretization and the nonlinear solver

We are interested in applying the method of DeC-Krylov-RAS to the system of Euler's equation:

$$\frac{\partial W}{\partial t} + \frac{\partial}{\partial x}F_1(W) + \frac{\partial}{\partial y}F_2(W) + \frac{\partial}{\partial z}F_3(W) = 0,$$

where $W = (\rho, \rho u, \rho v, \rho w, E)^T$ and $(F_1, F_2, F_3)^T$ is the convective flux as defined in [**8**]. Here and in the rest of the paper ρ is the density, $U = (u, v, w)^T$ is the velocity vector, E is the total energy per unit volume, and p is the pressure. These variables are related by the state equation for a perfect gas

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho\|U\|^2\right),$$

where γ denotes the ratio of specific heats ($\gamma = 1.4$ for air).

The computational domain is discretized by a tetrahedral grid. We use unstructured grids since they provide flexibility for tessellating complex, moving geometries and for adapting to flow features, such as shocks and boundary layers. We locate the variables at the vertices of the grid, which gives rise to a cell-vertex scheme. The space of solutions is taken to be the space of piecewise linear continuous functions. The discrete system is obtained via a finite volume formulation; see e.g., Koobus and Farhat [**10**]. We determine the nth time step size Δt^n in the following way. Let CFL be a pre-selected positive number. For each vertex x_i, let h_i be the size of the control volume centered at x_i, and we define the local time step size by

$$\Delta t_i^n = h_i \frac{\mathrm{CFL}}{C_i + \|U_i\|_2}$$

and then the global time step is defined by

$$\Delta t^n = \min_i\{\Delta t_i^n\}. \tag{3}$$

Here C_i is the sound speed, and U_i is the velocity vector.

One of the effective techniques for solving (2) is based on the so-called Defect Correction (DeC) method ([**11**]): Suppose that we have an initial guess $W_h^{n+1,0}$ for W_h^{n+1} obtained by using information calculated at previous time steps, we iterate for $j = 0, 1, \ldots,$

$$W_h^{n+1,j+1} = W_h^{n+1,j} + \xi^j, \tag{4}$$

where ξ^j is the solution of the following linear system of equations

(5)

$$\left(\frac{3}{2\Delta t}I + \partial_W \Phi_h^{(1st)}(W_h^n)\right)\xi^j = -\left(\frac{3W_h^{n+1,j} - 4W_h^n + W_h^{n-1}}{2\Delta t} + \Phi_h^{(2nd)}(W_h^{n+1,j})\right).$$

Here I is an identity matrix and $\Phi_h^{(1st)}(\cdot)$ is a first order MUSCL discretization of $div(F(\cdot))$. To simplify the notation, we use

$$g_{n+1,j} \equiv -\left(\frac{3W_h^{n+1,j} - 4W_h^n + W_h^{n-1}}{2\Delta t} + \Phi_h^{(2nd)}(W_h^{n+1,j})\right)$$

to denote the *nonlinear residual* at the jth DeC iteration of the $(n+1)$th time step and re-write (5) as

$$A_n \xi^j = g_{n+1,j}. \tag{6}$$

We remark that (2) doesn't have to be solved exactly. All we need is to drive the nonlinear residual to below a certain *nonlinear tolerance* $\tau > 0$, i.e.,

$$\|g_{n+1,j}\|_2 \leq \tau \|g_n\|_2 \tag{7}$$

such that $W_h^{n+1,j}$ gives a second order accurate solution in both space and time. Also (6) does not need to be solved very accurately either, as its solution provides only a search direction for the outer DeC iteration. Preconditioned iterative methods are often used for finding a $\xi^j = M_n^{-1}\eta^j$ such that

$$\|A_n M_n^{-1}\eta^j - g_{n+1,j}\|_2 \leq \delta \|g_{n+1,j}\|_2 \tag{8}$$

for certain *linear tolerance* $\delta > 0$. Here M_n^{-1} is a preconditioner for A_n.

The effectiveness of the above mentioned method depends heavily, among other things, on the choice of the preconditioner and a balanced selection of the nonlinear and linear stopping tolerance τ and δ. In this paper, we focus on the study of a parallel restricted additive Schwarz preconditioned iterative method for solving (6) with various δ. More discussions and computational experience with the selection of the nonlinear and linear stopping tolerance τ and δ can be found in [**2**].

3. RAS with minimum overlap

We now describe a version of the RAS preconditioner, which was recently introduced in [**3**], with the smallest possible non-zero overlap. We consider a sparse linear system

$$A\xi = g, \tag{9}$$

where A is an $n \times n$ nonsingular sparse matrix obtained by discretizing a system of partial differential equations, such as (1), on a tetrahedral mesh $\mathcal{M} = \{K_i, i = 1, \ldots, M\}$, where K_i are the tetrahedra. Using an element-based partitioning, $\mathcal{M}$ can be decomposed into N nonoverlapping sets of elements, or equivalently into N overlapping sets of nodes (since tetrahedra in different subsets may share the same nodes). Let us denote the node sets as $W_i, i = 1, \ldots, N$. Let W be the set of all the nodes, then we say that the node-based partition

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

is a minimum overlap partition of W. "minimum" refers to the fact that the corresponding element-based partition has zero overlap. The nodes belonging to more than one subdomains are called interface nodes. To obtain a node-based nonoverlapping partition, we identify a unique subdomain as the sole owner of each interface node. This leads to a node-based nonoverlapping partition of W, as shown in Fig.1 for a 2D mesh, or more precisely $W_i^{(0)} \subset W_i$, and

$$\bigcup_{i=1}^{N} W_i^{(0)} = W \text{ and } W_i^{(0)} \bigcap W_j^{(0)} = \emptyset \text{ for } i \neq j.$$

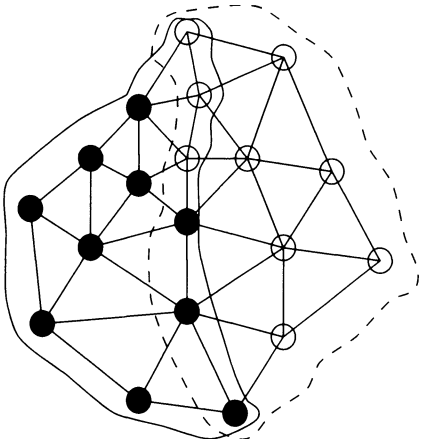

FIGURE 1. A minimum overlap two-subdomain partition. $W_1^{(0)}$ contains all the '•' nodes, and $W_2^{(0)}$ contains all the 'o' nodes, therefore $W_1^{(0)} \cap W_2^{(0)} = \emptyset$. $W_1^{(1)}$ contains all the nodes bounded inside the solid curve, and $W_2^{(1)}$ contains all the nodes bounded inside the dotted curve.

Let m be the total number of nodes in W. Associated with each W_i^0 we define a restriction operator R_i^0. In matrix terms, R_i^0 is an $m \times m$ block-sub-identity matrix whose diagonal blocks are set to $I_{5\times 5}$ if the corresponding node belongs to W_i^0 and to a zero 5×5 block otherwise. Similarly we can define R_i for each W_i. Note that both R_i^0 and R_i are of size $n \times n$. With this we define the matrix,

$$A_i = R_i A R_i \ .$$

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 in $\mathbf{R}^n$. Recall that the regular additive Schwarz (AS) preconditioner is defined as $M_{AS}^{-1} = \sum R_i A_i^{-1} R_i$, e.g., [**4, 13**]. Our RAS algorithm can be simply described as follows: Obtain the solution $\xi = M_{RAS}^{-1}\eta$ by solving the right-preconditioned system

$$A M_{RAS}^{-1} \eta = g$$

with a Krylov subspace method, where the preconditioner is defined by

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

In the numerical experiments to be reported in the next section, all subdomain problems are solved with ILU(0) and GMRES(5) ([**12**]) is used as the Krylov solver. Because of the page limit, we shall restrict our discussion to this particular preconditioner. Other issues can be found in the papers [**1, 2**]. We remark that the action of R_i^0 to a vector does not involve any communication in a parallel implementation, but R_i does. As a result, RAS is cheaper than AS in terms of the communication cost. We will show in the next section that RAS is in fact also cheaper than AS in terms of iteration counts.

TABLE 1. Iteration counts. Euler flow passing an oscillating wing at Mach 0.89. The mesh contains $n = 22014$ nodes. *subd* is the number of subdomains and δ is the stopping condition.

$n = 22014$	$\delta = 10^{-2}$			$\delta = 10^{-8}$		
subd	JAC	AS	RAS	JAC	AS	RAS
4	26	7	6	129	40	30
8	26	8	6	129	41	30
16	26	9	7	129	44	31
32	26	9	6	129	46	32

4. Numerical studies

In this section, we present several numerical simulations of unsteady 3D flows to demonstrate the scalability and robustness of the RAS preconditioner. We also include some comparisons with the regular additive Schwarz method and the simple pointwise Jacobi method (JAC). Note that a point in the mesh represents an 5×5 block matrix. Other recent development in the application of RAS in CFD can be found in [**6, 9**].

4.1. Parallel implementation issues. We implemented the algorithm on a number of parallel machines, and the top-level message-passing calls are implemented through MPI [**7**]. We partition the mesh by using the TOP/DOMDEC package [**5**]. We require that all subdomains have more or less the same number of mesh points. An effort is made to reduce the number of mesh points along the interfaces of subdomains to reduce the communication cost. The mesh generation and partitioning are considered as pre-processing steps, and therefore not counted toward the CPU time reported. The sparse matrix defined by (5) is constructed at every time step and stored in an edge-based sparse format.

4.2. A transonic flow passing a flexible wing. We tested our algorithm for an Euler flow passing a flexible wing at $M_\infty = 0.89$. The wing is clamped at one end and forced into the harmonic motion We test the algorithm on two unstructured meshes with 22014 and 331233 nodes, respectively. The two meshes are generated independently, i.e., one is not a refined version of the other.

We focus on the performance of the algorithm for solving a single linear system. The results on the coarser grid are summerized in Table 1 with CFL=900. Table 2 is for the finer mesh with CFL=100. Due to the special choice of the CFL numbers, the time steps for the two test cases are roughly the same. Comparing the RAS columns in Tables 1 and 2, we see that there is little dependence on the mesh sizes. And, we also see clearly that JAC has a strong dependence on the mesh sizes. As the number of subdomains grows from 4 to 16 or 32, the number of iterations of RAS stays more or less the same without having a coarse space in the preconditioner. Another observation is that RAS requires 20% to 30% fewer number of iterations than AS for the test cases.

4.3. A supersonic flow passing a complete aircraft. We consider a supersonic, $M_\infty = 1.9$, Euler flow passing a complete aircraft. The mesh contains $n = 89144$ nodes. In Table 3, we report the number of iterations for solving a single

TABLE 2. Iteration counts. Euler flow passing an oscillating wing at Mach 0.89. The mesh contains $n = 331233$ nodes. *subd* is the number of subdomains and δ is the stopping condition.

$n = 331233$	$\delta = 10^{-2}$			$\delta = 10^{-8}$		
subd	JAC	AS	RAS	JAC	AS	RAS
4	58	9	7	253	51	36
8	58	9	7	253	52	36
16	58	10	7	253	52	36

TABLE 3. Supersonic Euler's flow on a 3D unstructured mesh at Mach 1.90. The number of processors equals the number of subdomains *subd*. Number of nodes = 89144. The CPU (in seconds) time below is for solving one linear system.

$n = 89144$	$subd = 4$			$subd = 8$		
	JAC	AS	RAS	JAC	AS	RAS
ITER	96	37	28	96	38	29
CPU	33	29	23	16	14	11
COMM	0.2	0.15	0.1	0.3	0.2	0.15

linear system with JAC, AS and RAS preconditioned GMRES(5) methods. The CPU and communication (COMM) times are obtained on a SGI Origin 2000 with 4 and 8 processors. Even though this is a shared memory machine, we still treat it as a message-passing machine. The results are given in Table 3 for $\delta = 10^{-6}$ and CFL=1000.

5. Concluding remarks

We studied the performance of a newly introduced RAS preconditioner and tested it in several calculations including a transonic flow over an oscillating wing and a supersonic flow passing a complete aircraft. RAS compares very well against the regular additive Schwarz method in terms of iteration counts, CPU time and communication time when implemented on a parallel computer. Even though we do not have a coarse space, the number of iterations is nearly independent of the number of subdomains for all the test cases.

References

1. X.-C. Cai, M. Dryja, and M. Sarkis, *A convergence theory for restricted additive Schwarz methods*, in preparation, 1998.
2. X.-C. Cai, C. Farhat, B. Koobus, and M. Sarkis, *Parallel restricted additive Schwarz based iterative methods for general aerodynamic simulations*, in preparation, 1998.
3. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, Tech. Report CU-CS-843-97, Department of Computer Science, University of Colorado at Boulder, 1997.
4. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput. **15** (1994), 604–620.

5. 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), 13–26.
6. W. Gropp, D. Keyes, L. McInnes, and M. Tidriri, *Parallel implicit PDE computations: Algorithms and software*, Proceedings of Parallel CFD'97, A. Ecer, et al., edt., Manchester, UK, 1997., 1997, to appear.
7. W. Gropp, E. Lusk, and A. Skjellum, *Using MPI - Portable Parallel Programming with the Message-Passing Interface*, The MIT Press, 1995.
8. C. Hirsch, *Numerical Computation of Internal and External Flows, Vol I*, Wiley, New York, 1990.
9. D. Keyes, D. Kaushik, and B. Smith, *Prospects for CFD on petaflops systems*, CFD Review 1997, M. Hafez et al., edt., 1997, to appear.
10. B. Koobus and C. Farhat, *Time-accurate schemes for computing two- and three-dimensional viscous fluxes on unstructured dynamic meshes*, AIAA-96-2384, 1996.
11. R. Martin and H. Guillard, *A second order defect correction scheme for unsteady problems*, Computers and Fluids **25** (1996), 9–27.
12. Y. Saad and M. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput. **7** (1986), 856–869.
13. B. Smith, P. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309
E-mail address: cai@cs.colorado.edu

Department of Aerospace Engineering, University of Colorado at Boulder, Boulder, CO 80309
E-mail address: charbel@alexandra.colorado.edu

Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309
E-mail address: msarkis@cs.colorado.edu