1

Schwarz Methods for the Unsteady Compressible Navier-Stokes Equations on Unstructured Meshes

Xiao-Chuan Cai[1], Charbel Farhat[2] and Marcus Sarkis[3]

1.1 Introduction

Overlapping Schwarz is a family of preconditioners for solving large sparse linear systems arising from the discretization of partial differential equations, see e.g. [CS96, DW94, SBG95]. Here we report on our preliminary experiences on using it in the implicit solution of unsteady Navier-Stokes (N.-S.) equations discretized on two-dimensional unstructured meshes. One of the advantages of implicit methods is that they allow the time steps to be determined solely based on the physics of the fluid flow, not on the stability property of the time discretization scheme, [Ven95, VM95]. To advance in time, a large linear system of equations must be solved. Depending of the size of the time step, and other flow parameters, the conditioning of the matrix may change drastically from time step to time step. To solve these systems iteratively, it is necessary to have a family of preconditioners whose strength can be controlled. Overlapping Schwarz methods do have these properties, for

[1] Dept. of Comp. Sci., Univ. of Colorado, Boulder, CO 80309. *cai@cs.colorado.edu*. The work was supported in part by NSF ASC-9457534, NSF ASC-9217394 and NASA NAG5-2218.

[2] Dept. of Aerospace Eng., Univ. of Colorado, Boulder, CO 80309. *charbel@colorado.edu*. The work was supported in part by NSF ASC-9217394, NSF ASC-9217394 and NASA NAG5-2218.

[3] Dept. of Comp. Sci., Univ. of Colorado, Boulder, CO 80309. *msarkis@cs.colorado.edu*. The work was supported in part by NSF ASC-9406582, NSF ASC-9217394 and NASA NAG5-2218.

examples, they have adjustable strength, controlled by (1) using the inexact solution techniques for solving local problems; (2) including or excluding the coarse preconditioner; (3) changing the size of the coarse mesh.

In this paper, we investigate the difference between the Schwarz family of preconditioners and the global ILU preconditioners. Within the Schwarz preconditioners, we try to understand the role of the overlapping size between subdomains, the effect of the number of subdomains and inexact subdomain solvers. At each time step, we solve the resulting global linear system by the preconditioned GMRES method, and in the preconditioning stage, we solve the local subdomain problems again by the preconditioned GMRES method, with different preconditioners and stopping conditions. Since the construction of the preconditioner is very expensive, we explore the possibility of re-using the preconditioner for several time steps. For steady state problems, some studies can be found in [GKM94]. For other recent development in unsteady calculations, we refer the reader to [BL95, Ven95, VM95].

1.2 Governing Equations

Let $\Omega \subset \Re^2$ be the flow domain and Γ its boundary. The conservative form of the N.-S. equations is given by

$$\frac{\partial}{\partial t} W + \nabla_{\vec{x}} \cdot \mathcal{F}(W(\vec{x},t)) = \frac{1}{\mathrm{Re}} \nabla_{\vec{x}} \cdot \mathcal{R}(W(\vec{x},t)), \tag{1.1}$$

where $W = (\rho, \rho u, \rho v, E)^T$, and $\vec{x}$ and t denote the spatial and temporal variables. The detailed definition of $\mathcal{F}$ and $\mathcal{R}$ can be found in [FFL93]. In the above expressions, ρ is the density, $\vec{U} = (u, v)^T$ is the velocity vector and E is the total energy per unit volume.

We are interested in unsteady, external flows around an airfoil as pictured in Fig.1. The domain boundary is $\Gamma = \Gamma_w \cup \Gamma_\infty$ and the far field velocity is $\vec{U}_\infty$. On the wall boundary Γ_w, a no-slip condition on $\vec{U}$ and a Dirichlet condition on the temperature T are imposed, i.e., $\vec{U} = \vec{0}$ and $T = T_w$. No boundary conditions are specified for the density. In the far field, the viscous effect is assumed to be negligible, therefore a uniform free-stream velocity $\vec{U}_\infty$ is imposed on Γ_∞. More precisely, $\rho = \rho_\infty$, $\vec{U}_\infty = (cos\ \alpha, sin\ \alpha)^T$, and the pressure $p_\infty = 1/(\gamma M_\infty^2)$, where α is the angle of attack and M_∞ is the free-stream Mach number.

1.3 Discrete Formulation

Let the temporal variable t be discretized as $t^{n+1} = t^n + \delta t^n$, where δt^n is the discrete time increment. We also consider the increment $\delta W^{n+1} = W^{n+1} - W^n$, where W^n is an approximation of $W(\cdot, t^n)$. We note that

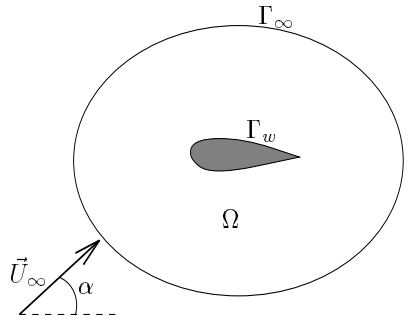

Figure 1 The computational domain

when an algorithm is written in the "delta" form, the increment δW^{n+1} is the unknown variable rather than W^{n+1}. Here, we use a first-order accurate temporal difference approximation, namely, the backward Euler scheme given as

$$\frac{\delta W^{n+1}}{\delta t^{n+1}} + (\nabla_W(\mathcal{F}^n)\cdot\nabla_{\vec{x}})\,\delta W^{n+1} - \frac{1}{\mathrm{Re}}(\nabla_W(\mathcal{R}^n)\cdot\nabla_{\vec{x}})\,\delta W^{n+1} = \nabla\cdot(-\mathcal{F}^n) + \frac{1}{\mathrm{Re}}\nabla\cdot\mathcal{R}^n, \qquad (1.2)$$

where $\mathcal{F}^n$ and $\mathcal{R}^n$ are approximations of $\mathcal{F}(W(\cdot,t^n))$ and $\mathcal{R}(W(\cdot,t^n))$, respectively.

The computational domain is discretized by an unstructured, triangular grid. We locate the variables at the vertices of the grid. This gives rise to a cell-vertex scheme. The space of solutions is taken to be the the space of piecewise linear functions. The discrete system is obtained via a mixed Galerkin finite element/finite volume formulation; see e.g. Farhat et al. [FFL93], and Fezoui and Stoufflet [FS89]. In short, the discrete system for (1.2) is obtained by using a "mass-lumping" technique for the time derivative, a first-order MUSCL scheme with a Roe approximate Riemann solver for the convective terms of the LHS, and a Galerkin finite element (first-order quadrature integration) for the diffusive terms of the LHS. For the RHS, we use a second-order MUSCL scheme with a Roe approximate Riemann solver and Van Albada's limiting procedure for the convective terms, and a Galerkin finite element method for the diffusive terms.

For the initial values, we assume that W_0 satisfies strongly the wall boundary conditions on Γ_w and the far field boundary conditions at infinity. In the interior nodes of Ω, W_0 takes the free-stream boundary condition.

1.4 Algebraic Schwarz Algorithms

At each time step, we solve a linear system, $Au = f$, where A is a nonsymmetric sparse matrix with symmetric non-zero pattern. Each element of A can be considered as a 4×4 matrix, and each unknown of the vector u is a '4-size' vector. Thus, there is a bijection between unknowns and vertices. We

denote the set of vertices (or nodes) by $\mathcal{N} = \{1, \cdots, n\}$, where n represents the total number of nodes (or unknowns). To define algebraic Schwarz algorithms [CS96], we first partition the set $\mathcal{N}$ into n_0 nonoverlapping subsets $\mathcal{N}_i$ whose union is $\mathcal{N}$. We use the TOP/DOMDEC mesh partitioning package of Farhat et al. [FLS95] to obtain sets $\mathcal{N}_i$. The recursive spectral bisection method with certain optimization is used in the partitioning. The number of nodes in each $\mathcal{N}_i$ is roughly the same. To generate an overlapping partition, we further expand each subgrid $\mathcal{N}_i$ by *ovlp* number of neighboring nodes, denoted as $\tilde{\mathcal{N}}_i$.

We denote by L_i the vector space spanned by the set $\tilde{\mathcal{N}}_i$. For each subspace L_i we define an orthogonal projection operator I_i as follows: I_i is a $n \times n$ matrix whose diagonal elements are set to 4×4 identity matrices if the corresponding nodes belong to $\tilde{\mathcal{N}}_i$ and to 4×4 zero matrices otherwise. With this we define $A_i = I_i A I_i$, which is an extension to the whole subspace, of the restriction of A to L_i. Note that although A_i is not invertible, we can invert its restriction to the subspace spanned by $\tilde{\mathcal{N}}_i$, and define $A_i^{-1} \equiv I_i \left((A_i)_{|L_i} \right)^{-1} I_i$. The additive and multiplicative Schwarz algorithms can now be simply described as follows: Solve the equation $MAu = Mf$ by a Krylov subspace method, where $M = A_1^{-1} + \cdots + A_{n_0}^{-1}$, for the additive Schwarz algorithm, and $M A = I - (I - A_1^{-1} A) \cdots (I - A_{n_0}^{-1})$ for the multiplicative Schwarz algorithm. We remark that in the algorithms discussed above all subproblems are assumed to be solved exactly; e.g., with sparse Gaussian elimination. In our numerical experiments we also consider inexact solvers.

1.5 Numerical Results

The main goal of this section is to compare the effectiveness of various preconditioners for unsteady subsonic and transonic flows. The experiments were performed using PETSc [GSM95] on a DEC Sable workstation.

We consider flows past a NACA0012 airfoil at an angle of attack of 30 degrees and Reynolds number 800. Problem 1 corresponds to a Mach number 0.1 and CFL 100, Problem 2 to a Mach number 0.8 and CFL 100, and Problem 3 to a Mach number 0.8 and CFL 25. The iteration numbers and CPU times that we report are for one single time step and for a non-dimensionalized time far from the initial transient regime, i.e. time equals to 3.0. In the CPU time, we do not include the time for constructing the preconditioner since the same preconditioner can be frozen for several time steps. We used left preconditioners and we stopped the iterations when the l_2 norm of the preconditioned residuals were reduced by a factor of 10^{-6}.

We found that we can take CFL up to 100 without losing much accuracy for the unstructured grid with 12280 nodes; see the left figure in Fig.2. Therefore, using the implicit methods has a clear advantage over the explicit one in which

the CFL must be less than 1.0.

Table 3 illustrates the results for additive and multiplicative Schwarz methods. The multiplicative OSM had better convergence properties although it is not as parallelizable as the additive OSM. We can also see that a small overlap gave generally less GMRES iterations than zero overlap. For Problem 2 and 3, we detected some pathological cases in which an increase in the overlap resulted in more GMRES iterations. We note, however, that when the subdomains were relatively large, a small overlap resulted in a significant decrease in the CPU time. We will expect promising results when we run our code on parallel machines with coarse granularity. We observed a slight increase in the number of GMRES iterations when we increased the number of subdomains, thus, it is not clear that for unsteady problems an additional coarse space would decrease the CPU time.

Table 4 represents the behavior of the GMRES/OSM iteration numbers with different inexact local solvers. We found that if we increased the overlap, the number of GMRES iterations increased when the local problems are not solved to a certain tolerance. We tested several cases of additive OSM (8 subdomains) with GMRES/ILU($fill$) as the inexact solvers. The best performance in terms of CPU time was zero overlap and zero $fill$ with local stopping tolerance 1.0^{-2}; see Table 1. We noted also that as a local solvers, using LU gave smaller CPU time than using GMRES/ILU(0). We note, however, that for larger problems we might obtain different conclusions.

We studied the behavior of other preconditioners; see Table 2. We observed that the 4×4 block Jacobi iterative method required a very large number of iterations. The family of global ILU($fill$) gave the best results but it would not be trivial to code them on parallel machines.

Finally, we examined the effect of using the same preconditioner for several time steps, thus, we did not need to form the preconditioner or factorize it at every time step; see the right figure in Fig.2. We considered a case in which we used LU local solver in one time step and then used this LU factorization as a preconditioner for local problems (with one Richardson iteration) for the following time steps. We found that for additive OSM with 8 subdomains and 1 overlap the convergence did not deteriorate for 25 time steps.

1.6 Conclusions

We report the performance of GMRES/ILU($fill$) and GMRES/OSM methods for solving systems that arise from the discretization of unsteady, compressible N.-S equations. The best results in terms of CPU time were obtained for the GMRES /ILU($fill$) methods, where $fill$ is around 5. However, the ILU($fill$) is not easy to parallelize. We then tested several additive OSM. For problems in which the size of the subdomains were not too small, we observed that we

reduced the CPU time if we used a small overlap rather than zero overlap. We note that the local problems were solved by Gaussian elimination. When using inexact local solvers, we noted that an overlap did not improve performance, and that the inexact local solver GMRES/ILU(0) gave the smallest CPU time; we must mention however that this observation may change if we can run problems with large subdomains. We also found that a slight increase in the number of GMRES iterations occurred when we increased the number of subdomains. We suspect that an improvement in performance might be obtained if we were to add a coarse space.

REFERENCES

[BL95] Barth T. J. and Linton S. W. (Jan. 1995) An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation. *AIAA Paper 95-0221*.

[CS96] Cai X.-C. and Saad Y. (1996) Overlapping domain decomposition algorithms for general sparse matrices. *Numer. Lin. Alg. Applics* To appear.

[DW94] Dryja M. and Widlund O. B. (1994) Domain decomposition algorithms with small overlap. *SIAM J. Sci. Comp.* 15(3): 604–620.

[FFL93] Farhat C., Fezoui L., and Lanteri S. (1993) Two-dimensional viscous flow computation on the Connection Machine: Unstructured meshes, upwind schemes and parallel computation. *Comput. Methods Appl. Mech. Engrg.* 102: 61–88.

[FLS95] Farhat C., Lanteri S., and Simon H. (1995) TOP/DOMDEC: A software tool for mesh partitioning and parallel processing and applications to CSM and CFD computations. *Comput. Sys. Engrg.* 6(1): 13–26.

[FS89] Fezoui L. and Stoufflet B. (1989) A class of implicit upwind schemes for Euler simulations with unstructured meshes. *J. Comp. Phys.* 84: 174–206.

[GKM94] Gropp W. D., Keyes D. E., and Mounts J. S. (1994) Implicit domain decomposition algorithms for steady, compressible aerodynamics. In Quarteroni A., Periaux J., Kuznetsov Y. A., and Widlund O. B. (eds) *Sixth Conference on Domain Decomposition Methods for Partial Differential Equations.* AMS, Providence, RI.

[GSM95] Groop W. D., Smith B. F., and McInnes L. C. (1995) PETSc 2.0 User's Manual. Technical Report ANL-95/11, Argonne National Laboratory.

[SBG95] Smith B. F., Bjørstad P. E., and Gropp W. D. (1995) *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press. To appear.

[Ven95] Venkatakrishnan V. (Feb. 1995) A perspective on unstructured grid flow solvers. Technical Report ICASE Report No. 95-3, ICASE, NASA Langley Research Center.

[VM95] Venkatakrishnan V. and Mavriplis D. J. (Aug. 1995) Implicit method for the computation of unsteady flows on unstructured grids. Technical Report ICASE Report No. 95-60, ICASE, NASA Langley Research Center.

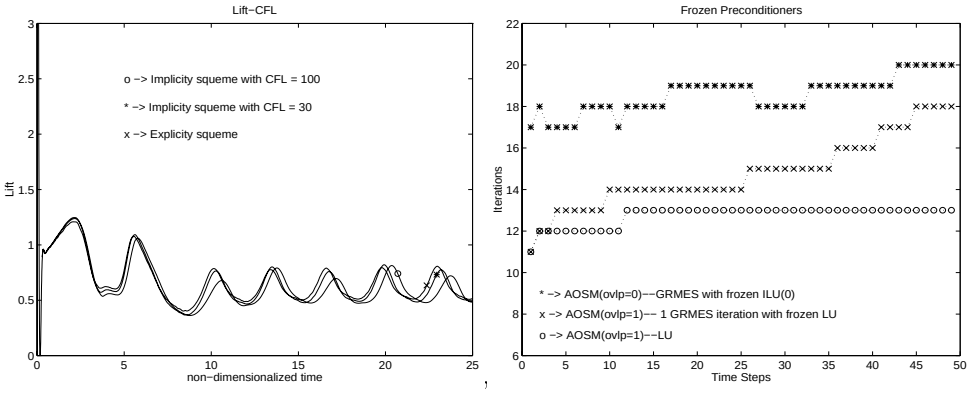

Figure 2 The left figure compares the solutions obtained by both explicit and implicit methods. The right figure shows the number of GMRES iterations when preconditioners are frozen for a number of time steps.

Table 1 GMRES iteration number and CPU time in seconds using Additive OSM with zero overlap. Number of nodes is 12280. We use GMRES/ILU(0) as local solvers reducing the local preconditioned residual by $= 2.0e^{-1}$.

	Problem 1	Problem 2	Problem 3
SUB = 8	17 (31.9)	14 (22.5)	8 (11.4)
SUB = 16	20 (29.3)	13 (17.6)	8 (9.9)
SUB = 32	23 (33.5)	15 (19.3)	10 (11.8)
SUB = 64	25 (33.3)	17 (21.1)	10 (11.6)
SUB = 128	29 (37.5)	20 (23.6)	12 (13.5)
SUB = 256	35 (43.9)	22 (25.5)	13 (14.0)
SUB = 512	41 (49.9)	27 (30.2)	16 (16.7)

Table 2 Iteration numbers and (CPU time in seconds) to reduce the preconditioned residual to $1.0e^{-6}$ using GMRES/ILU($fill$) and Richardson/($4\times$ 4 block Jacobi).

	Problem 1	Problem 2	Problem 3
4X4 BJ	219 (44.4)	190 (40.3)	98 (18.0)
FILL = 0	34 (29.1)	21 (17.1)	10 (7.7)
FILL = 1	23 (21.4)	13 (13.6)	7 (6.8)
FILL = 2	19 (20.1)	12 (13.5)	6 (6.9)
FILL = 3	9 (10.9)	6 (8.18)	4 (5.7)
FILL = 4	7 (9.1)	5 (7.5)	3 (4.8)
FILL = 5	6 (8.4)	4 (6.5)	3 (5.1)
FILL = 6	6 (8.7)	4 (6.7)	2 (3.6)

Table 3 Iteration numbers and CPU time per time step GMRES/OSM (tolerance = $1.0e^{-6}$) for Problems 1, 2 and 3 using 12280 nodes. As local solvers we use LU.

Problem 1	ovlp 0	ovlp 1	ovlp 2	ovlp 3
SUB = 8 (a)	15 (27.8)	11 (22.4)	10 (22.5)	9 (21.8)
(m)	8 (17.4)	5 (12.5)	4 (11.4)	4 (12.3)
SUB = 16(a)	17 (27.4)	13 (23.9)	13 (26.9)	12 (27.9)
(m)	10 (18.9)	6 (13.6)	5 (13.0)	5 (14.7)
SUB = 32(a)	19 (25.4)	16 (26.5)	15 (30.1)	15 (36.0)
(m)	11 (17.5)	7 (14.4)	6 (15.2)	5 (15.5)
SUB = 64(a)	21 (25.6)	20 (31.1)	20 (40.0)	19 (47.7)
(m)	12 (16.8)	9 (17.4)	7 (17.9)	6 (19.7)
Problem 2				
SUB = 8 (a)	13 (24.0)	9 (18.3)	7 (16.1)	8 (19.5)
(m)	7 (15.6)	3 (8.2)	3 (9.1)	3 (9.7)
SUB = 16(a)	13 (20.8)	9 (16.7)	10 (20.8)	10 (23.3)
(m)	7 (13.5)	4 (9.5)	4 (10.8)	3 (9.6)
SUB = 32(a)	14 (18.4)	12 (19.8)	11 (22.0)	12 (28.6)
(m)	8 (12.9)	5 (10.6)	4 (10.7)	3 (10.1)
SUB = 64(a)	15 (17.7)	15 (23.6)	15 (30.6)	15 (38.6)
(m)	9 (12.6)	5 (10.2)	4 (11.0)	4 (13.9)
Problem 3				
SUB = 8 (a)	8 (15.1)	7 (14.6)	6 (14.0)	6 (15.0)
(m)	5 (11.4)	2 (6.1)	2 (6.7)	2 (7.2)
SUB = 16(a)	8 (13.0)	7 (13.3)	8 (16.9)	8 (19.1)
(m)	5 (10.0)	3 (7.6)	3 (8.5)	2 (7.1)
SUB = 32(a)	10 (13.3)	9 (15.2)	8 (16.5)	9 (21.8)
(m)	6 (9.9)	3 (7.0)	3 (8.5)	2 (7.6)
SUB = 64(a)	10 (11.8)	11 (17.0)	11 (22.1)	12 (30.3)
(m)	6 (8.7)	3 (6.7)	3 (8.8)	3 (11.1)

ovlp=0	Problem 1	Problem 2	Problem 3
SUB = 128	25 (26.7)	17 (17.5)	11 (11.3)
SUB = 256	31 (29.7)	20 (18.1)	13 (11.4)
SUB = 512	39 (35.7)	26 (21.5)	15 (11.7)

Table 4 GMRES iteration numbers to reduce the preconditioned residual of Problem 1 (12280 nodes) to $1.0e^{-6}$ using GMRES/(Additive OSM) with 8 subdomains. We use GMRES/ILU(0) inexact local with different local solvers stopping criteria.

tolerance	ovlp = 0	ovlp = 1	ovlp = 2	ovlp = 3
1 iteration	35	43	46	48
$5.0e^{-1}$	23	32	33	33
$3.0e^{-1}$	18	20	19	19
$1.0e^{-1}$	16	15	14	14
$1.0e^{-2}$	15	12	11	10
exact	15	11	10	9