80

Variable-degree Schwarz Methods for Unsteady Compressible Flows

Xiao-Chuan Cai, Charbel Farhat, and Marcus Sarkis

1 Introduction

We introduce a new variant of the overlapping Schwarz method (OSM) for solving unsteady problems. In particular we study implicit methods ([FFL93, FS89, VM95]) for obtaining the time accurate solution of the compressible Navier-Stokes equations discretized on two-dimensional unstructured meshes. When using implicit methods, a large, sparse linear system must be constructed and solved at each time step. Depending of the size of the time step, and several other flow parameters, the conditioning of the matrix may change from well-conditioned to mildly ill-conditioned. Furthermore, due to the complexity of the flow pattern, at a given time step the matrix may be ill-conditioned in certain subregions, for example near the airfoil, and relatively well-conditioned elsewhere. To solve these systems iteratively, it is necessary to have a family of preconditioners, such as OSM, whose strength can be adjusted locally in each subdomain according to the flow condition.

It is known that when constructing a preconditioner for a single linear system $Au = f$ all the information needed is from the matrix A. However, the issue for time dependent problems is different. A sequence of interrelated systems $A^{(k)}u^{(k)} = f^{(k)}$ have to be solved. If the matrix, especially in its (often inexactly) factorized form, obtained at a previous time step can be properly used, then the preconditioner at the current time step can be obtained cheaply. More precisely, at each time step, we solve the linear system by a preconditioned GMRES method and in the preconditioning stage, following the general OSM framework, we solve the local subdomain problems by another preconditioned GMRES method with different preconditioners and stopping conditions. In each subdomain the preconditioner is built by using a polynomial in two matrix variables, namely the matrix, in its *unfactorized* form, of the current time step k and another matrix, in its *factorized* form obtained at a previous time step j. The degree of the matrix polynomial reflects the conditioning of the subdomain matrix. Note that classical Schwarz methods correspond to the case where the degree of the matrix polynomials always equals to one. In our new method, the degree of the polynomial varies from subdomain to subdomain depending on the flow conditions,

---

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

and therefore we refer to the methods as variable degree Schwarz methods (VDS).

In this paper, we also study the effects of the overlapping size, the number of subdomains, and the inexact subdomain solvers. Since the construction of the preconditioner is expensive, we also explore the possibility of reusing the preconditioner for several time steps.

2 Variable-degree Schwarz Methods

Suppose that at each time step k we need to solve $A^{(k)}u^{(k)} = f^{(k)}$ by an iterative method with a preconditioner $M^{(k)}$ to a certain accuracy, i.e.,

$$\left\| M^{(k)} \left(A^{(k)} u^{(k)} - f^{(k)} \right) \right\|_2 \leq \tau \| M^{(k)} f^{(k)} \|_2, \tag{2.1}$$

where τ is a given tolerance. Let n be the total number of unknowns and $\mathcal{N} = \{1, \cdots, n\}$. To define algebraic Schwarz algorithms, see e.g. [CS96], we first partition $\mathcal{N}$ into n_0 nonoverlapping subsets $\{\mathcal{N}_i\}$ whose union is $\mathcal{N}$. To generate an overlapping partitioning with overlap *ovlp*, we 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 and $A_i^{(k)} = I_i A^{(k)} I_i$, which is an extension to the whole subspace, of the restriction of $A^{(k)}$ to L_i. We define its "inverse" by $(A_i^{(k)})^{-1} \equiv I_i \left((A_i^{(k)})_{|L_i} \right)^{-1} I_i$. The classical additive and multiplicative Schwarz algorithms can be described as follows([CS96, CM94, DW94]): Solve $MA^{(k)}u^{(k)} = Mf^{(k)}$ by a Krylov subspace method, where

$$M = (A_i^{(k)})^{-1} + \cdots + (A_{n_0}^{(k)})^{-1} \ , \quad \text{and} \tag{2.2}$$

$$M\,A^{(k)} = I - \left(I - (A_i^{(k)})^{-1} A^{(k)} \right) \cdots \left(I - (A_{n_0}^{(k)})^{-1} A^{(k)} \right) \tag{2.3}$$

for the additive and multiplicative Schwarz algorithms, respectively.

There are three major steps in the construction of the Schwarz preconditioners, namely 1) the construction of the matrix $A^{(k)}$; 2) the construction of the matrices $A_i^{(k)}$; and 3) the incomplete factorization of the matrices $A_i^{(k)}$. In fact Step 1) is not necessary since the matrices constructed in Step 2) can be used to calculate the matrix-vector multiplications. Since we are interested in implicit methods, Step 2) has to be done at every time step no matter how expensive it is. One expensive step in the construction of the preconditions as formulated above for time dependent problems is Step 3). One way to avoid the frequent factorization of $A_i^{(k)}$ is to simply use some old factorized matrix $A_i^{(j)}$ calculated at time step j, where $j < k$. However, this method may not be very effective if j and k are too far apart. More discussion on using frozen preconditioners can be found later in the paper.

Another problem with the Schwarz preconditioners (2.2) and (2.3) is that all subdomains are treated equally in terms of the level of preconditioning in the sense that the number of applications of $(A_i^{(k)})^{-1}$, or its inexact version, is the same on all subdomains, regardless of the fact that the subdomain matrices $A_i^{(k)}$ have vary different condition numbers. Physically speaking, the behavior of the flows in subdomains near the body of the airfoil, or near the shocks, is very different from the other regions. More preconditioning is needed only in subdomains where the real action take place.

We propose a method that places different levels of preconditioning in different subdomains and will also show by numerical experiments that the methods remain to be effective even if j and k are far apart from each other. The idea is simple. We replace the matrix-vector multiply in (2.2) or (2.3)

$$w = (A_i^{(k)})^{-1} v \tag{2.4}$$

by another *iterative procedure* with $(B_i^{(j)})^{-1}$ as the preconditioner. Here $B_i^{(j)}$ is an incomplete factorization of $A_i^{(j)}$ with certain levels of fill-in at time step j. More precisely speaking, to obtain w for a given v, we run several steps of GMRES in the subspace L_i such that

$$\left\| (B_i^{(j)})^{-1}(v - A_i^{(k)}\tilde{w}) \right\|_2 \leq \delta \left\| (B_i^{(j)})^{-1} v \right\|_2 . \tag{2.5}$$

We then set $w := \tilde{w}$. Here δ is a pre-selected small value. Examples can be found in Σ3. In the matrix language, we replace the matrix $(A_i^{(k)})^{-1}$ in (2.4) by a matrix polynomial $poly_i\left((B_i^{(j)})^{-1} A_i^{(k)}\right)$ of a certain degree. The actual degree depends on the number of GMRES iterations needed in the subspace L_i. To put them into a single form, the additive Schwarz preconditioner becomes

$$M = poly_1\left((B_1^{(j)})^{-1} A_1^{(k)}\right) + \cdots + poly_{n_0}\left((B_{n_0}^{(j)})^{-1} A_{n_0}^{(k)}\right).$$

Note that this preconditioner does not contain $(A_i^{(k)})^{-1}$, but it contains certain spectral information from $(A_i^{(k)})^{-1}$. This makes it very effective. In fact, M is a truncated series representation of $(A_i^{(k)})^{-1}$ based on a splitting of $A_i^{(k)}$ into the sum of $B_i^{(j)}$ and $A_i^{(k)} - B_i^{(j)}$. A discussion on a related polynomial preconditioning method can be found in [GO93]. We note that in a given subdomain, the number of GMRES iterations, or the degree of the polynomial, is determined by the conditioning of the local stiffness matrix. The multiplicative version can be constructed in a similar way.

We remark that since the preconditioner changes in the GMRES loop due to the stopping condition determined by δ, it is generally more appropriate to use the flexible GMRES [Saa93], which is slightly more expensive than the regular one. We do not use the flexible GMRES in this paper since the regular GMRES presents no problem for our test cases.

3 Numerical Results

The goal of this section is to demonstrate the usefulness of the family of VDS preconditioners in the implicit solution of compressible flow problems. We apply our algorithms to the simulation of two-dimensional low Reynolds number flows past a

NACA0012 airfoil at high angle of attack ($30^{\rm deg}$) and two different Mach numbers. No steady state solutions exists for both test cases described below. **Test 1**: The subsonic case with $M_\infty = 0.1$ and $Re = 800.0$. We use a pre-generated shape regular triangular mesh, Mesh12k, with 12280 nodes. **Test 2**: The transonic case with $M_\infty = 0.84$ and $Re = 1600.0$. We use a mesh, Mesh48k, with 48792 nodes obtained by uniformly refining the mesh used in **Test 1**. Because of the page limit, we do not discuss the discretization, time stepping and mesh partitioning in this paper; interested reader should consult [CFS96] for details.

In the implementation of VDS, we partition the mesh by using the recursive spectral bisection method. The sparse matrix is constructed at every time step, and stored in the Compressed Sparse Row format. The subdomain matrices are obtained by taking elements, according to a pre-selected index set, from the global matrix. A symbolic ILU(0) factorization of the subdomain matrix is performed at the very first time step, and reused at all the later time steps. This is possible due to the fact that the matrices, constructed at every time step, share the same non-zero pattern. We also tested the ILU(k) ($k > 0$) preconditioners, which are not competitive with ILU(0) in terms of the CPU time in our implementation for both test cases. We remark that if ILU with drop tolerance is used then the non-zero pattern of the matrices may change and the previously obtained symbolic factorizations cannot be reused.

We note that at the beginning of the motion of the flow, i.e., when the non-dimensionalized time $t \leq 1.0$, the flow changes so drastically that the use of any time step size δt^n that makes the corresponding CFL number larger than 1.0 would result in the loss of time accuracy for the entire calculation. This implies that small δt^n have to be used when $t \leq 1.0$, and therefore, the implicit method has to be abandoned for this initial period of time. In our experiments, the implicit solver is turned on at $t = 1.0$. The solution for the period $0 < t \leq 1.0$ is obtained with an explicit method with CFL=0.8.

The reports given below are based on running our implicit methods for 100 time steps starting at $t = 1.0$. We shall use **MaxIt** to denote the maximum number of global GMRES iterations and **TotalIt** the total number of global GMRES iterations within this 100 linear system solves. To measure the approximate cost of the methods, we use **EMatVec** to denote the equivalent number of global matrix-vector multiplications, which includes the actual stiffness matrix-vector multiplications and the preconditioning-matrix-vector multiplications.

Let us first discuss the dependence of the convergence rate on the number of subdomains. We use 5 different decompositions of Ω, with both Mesh12k and Mesh48k. The number of subdomains goes from 8 to 128. We run both **Test 1** and **2**, with *ovlp* equals to one fine mesh cell. In Table 1, we present the maximum number of global GMRES iterations within one hundred time steps and its corresponding **EMatVec**. If multiplicative VDS is used even without the special subdomain coloring or ordering, **MaxIt** is independent of the number subdomains for reasonably large number of subdomains, such as 128. An interesting case is shown on the top left portion of Table 1 which indicates that if additive VDS is used for the subsonic problem the number of maximum iterations does grow, though not very fast, as the number of subdomains becomes large. In this case, we believe that a coarse space may be useful to reduce the dependence on the number of subdomains. However, we have not implemented the coarse grid solver yet. For transonic problems, our tests show that the use of a coarse level grid is not necessary with both additive and multiplicative VDS preconditioners.

Table 1 CFL=50, $\tau = 10^{-3}$, *ovlp* $= 1$. We use GMRES/ILU(0) as inexact local solvers with $\delta = 10^{-1}$.

ASM	**Test 1**			**Test 2**		
# subdomains	MaxIt	TotalIt	EMatVec	MaxIt	TotalIt	EMatVec
ASM 8	6	545	3150	6	519	1471
ASM 16	7	585	3529	6	506	1628
ASM 32	9	673	3851	6	560	1864
ASM 64	10	756	4223	6	600	2184
ASM 128	11	842	5621	7	603	2192
MSM 8	4	292	1613	3	300	832
MSM 16	4	316	1812	3	300	915
MSM 32	4	320	1834	3	300	994
MSM 64	4	344	1900	3	300	1089
MSM 128	4	351	2335	3	300	1084

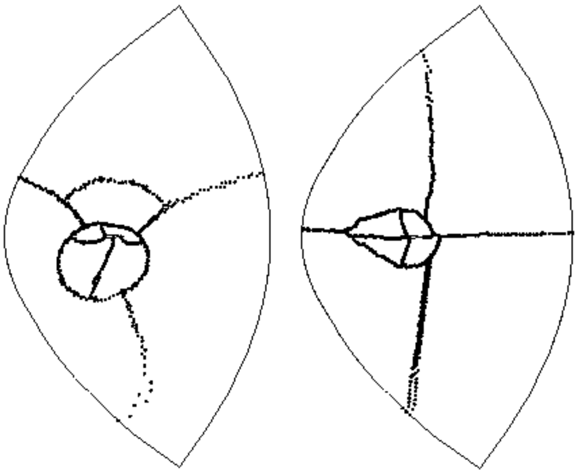
Table 2 Global GMRES/(multiplicative VDS) with CFL=50, $\tau = 10^{-3}$, $\delta = 10^{-1}$, *ovlp* $= 1$ and the local solvers are GMRES/ILU(0).

	Ω_1	Ω_2	Ω_3	Ω_4	Ω_5	Ω_6	Ω_7	Ω_8
Test 1, MaxIt	2	2	2	3	4	6	3	5
TotalIt	150	106	113	225	307	597	291	421
Test 2, MaxIt	2	2	2	2	2	2	1	1
TotalIt	127	200	109	107	132	185	150	100

Whether overlap is useful or not is a rather subtle issue. It depends on the global linear stopping parameter τ defined in (2.1) and the local linear stopping parameter δ defined in (2.5). According to a large number of tests we did large overlaps can reduce the number of iterations and CPU time only if the stopping parameter δ is small. In our situation when $\tau = 10^{-3}$, we find $\delta = 10^{-1}$ offers the best CPU results, and therefore we do not need large overlaps. In the rest of the tests, we use this set of τ and δ, and *ovlp* $= 1$.

We next look at the degree of preconditioning polynomial in each subdomain. We focus on the 8 subdomain cases with GMRES/ILU(0) as local subdomain solvers. The partitionings used for Mesh12k and Mesh48k are different, as shown in Fig. 1. The subdomains are numbered as in Fig. 1. The results obtained for one hundred time steps starting at $t = 1.0$ are summarized in Table 2. It turns out the required degrees of local preconditioning polynomials are quite different. For the subsonic case, subdomains Ω_6 and Ω_8 need more iterations (4 and 6 respectively) than other subdomains. The left picture of Fig. 1 shows that these two subdomains cover the top portion of the airfoil.

Figure 1 Left figure shows the partitioning of Mesh12k into 8 subdomains and right one shows that for Mesh48k. The airfoil is at the center of the domain.



Only two iterations are needed for subdomains that are far away from the airfoil, such as Ω_1, Ω_2 and Ω_3. The number of iterations reflects the conditioning of the subdomain matrix. For the transonic case, all subdomains need either one or two iterations. Tables 1 and 2 also show that the number of global and local iterations are surprisingly small. This indicates that the linear systems of equations are in fact not too ill-conditioned. We believe that this is due to the use of relatively small time steps, which is necessary in order to obtain time accurate solutions.

Table 3 Global GMRES/(multiplicative VDS) with CFL=50, $\tau = 10^{-3}$, $\delta = 10^{-1}$, *ovlp* = 1. The number of subdomains is 8. For the FreezeIt=200 case, the numbers are taken for 200 time steps divided by 2.

FreezeIt=	1	5	10	50	100	200
Test 1, EMatVec	1613	1611	1615	1618	1629	1875
TotalIt	292	292	291	290	291	321
Test 2, EMatVec	832	829	830	842	868	1243
TotalIt	300	300	300	305	315	469

Finally, we examine the effect of using the same preconditioner, or part of the preconditioner, for several time steps without doing the factorization at every time step. In Table 3, we summarize the results for using different numbers of frozen steps, namely FreezeIt $= 1, 5, \ldots$. There is a range of optimal FreezeIt one can choose from; similar numbers of EMatVec are obtained in our implementation for FreezeIt ranging from 5 to 50. For the subsonic case, we can go a bit further, e.g., take FreezeIt = 100.

4 Conclusions

We proposed and tested a family of variable degree Schwarz(VDS) preconditioned GMRES methods for solving linear systems that arise from the discretization of unsteady, compressible N.-S. equations on 2D unstructured meshes for both subsonic and transonic flows past a single element NACA0012 airfoil. In VDS, the level of preconditioning in each subdomain varies according to the local flow condition, therefore extra preconditioning is performed only when and where it is needed. For subsonic problems, we found that the conditioning of the subdomain matrices changes quite a bit from one flow region to another, and extra local preconditioning in subdomains in which the flow changes drastically can significantly reduce the total number of global linear iterations. This is somewhat less obvious for transonic flow, which needs a nearly uniformly small global and local number of iterations. When using VDS, the best results are obtained with small overlap. For the multiplicative version, the convergence rate depends very mildly on the number of subdomains (up to 128 subdomains have been tested), and for the additive version, a slight dependence is observed for the subsonic test problem and therefore a coarse space might be useful.

Acknowledgement

The first author was supported in part by NSF ASC-9457534, NSF ASC-9217394 and NASA NAG5-2218. The second author was supported in part by NSF ASC-9217394, NSF ASC-9217394 and NASA NAG5-2218. The third author was supported in part by NSF ASC-9406582, NSF ASC-9217394 and NASA NAG5-2218.

REFERENCES

[CFS96] Cai X.-C., Farhat C., and Sarkis M. (1996) Variable-degree Schwarz methods for the implicit solution of unsteady compressible Navier-Stokes equations on two-dimensional unstructured meshes. Technical Report ICASE Report No. 96-48, ICASE, NASA Langley Research Center.

[CM94] Chan T. F. and Mathew T. P. (1994) Domain decomposition algorithms. *Acta Numerica* (61-143).

[CS96] Cai X.-C. and Saad Y. (1996) Overlapping domain decomposition algorithms for general sparse matrices. *Numer. Lin. Alg. Applics* 3: 221–237.

[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.

[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.

[GO93] Golub G. and Ortega J. M. (1993) *Scientific Computing: An Introduction with Parallel Computing.* Academic Press, Inc.

[Saa93] Saad Y. (1993) A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Stat. Comput.* 14: 461–469.

[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.