Algebraic Multilevel Preconditioners for non-symmetric PDEs on Stretched Grids

Marzio Sala[1], Paul T. Lin[1], John N. Shadid[1], and Ray S. Tuminaro[1]

1. Sandia National Laboratories, Albuquerque, 87185 New Mexico **

Summary. We report on algebraic multilevel preconditioners for the parallel solution of linear systems arising from a Newton procedure applied to the finite-element (FE) discretization of the incompressible Navier-Stokes equations. We focus on the issue of how to coarsen FE operators produced from high aspect ratio elements. The method uses on each level ℓ an auxiliary matrix B_ℓ, which contains inter-nodal distance information of the underlying initial FE grid. Then, a standard coarsening procedure is performed on B_ℓ and non-smoothed transfer operators are defined. Preliminary numerical results obtained on distributed memory parallel computers show that the use of the auxiliary matrix can greatly improve the convergence rate of the resulting multilevel preconditioner.

1 Introduction

We consider linear systems of type

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

where $A \in \mathbb{R}^{n \times n}$ is a real square (sparse) matrix, arising from a stabilized FE discretization of the incompressible Navier-Stokes equations, possibly with heat and mass transfer, and $x, b \in \mathbb{R}^n$ are the solution vector and the right-hand side, respectively. The elements of A are defined by a Newton procedure (see for instance [10]), since the original problem is nonlinear.

The linear problem (1) is usually solved using a Krylov accelerator; therefore a preconditioner is mandatory. Several solution strategies have been presented in the literature; in this paper we will focus on multilevel methods.

** This work was partially supported by the ASC program and the DOE Office of Science MICS program at Sandia National Laboratory. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The basic idea of multilevel methods is to capture errors by utilizing multiple resolutions in the iterative scheme. Oscillatory components are effectively reduced through simple relaxation procedure. In these methods the smooth components are handled using an auxiliary, lower-resolution version of the problem. The idea is applied recursively on the next coarser level.

The first and best known example of a multilevel preconditioner is multigrid (see, for example, [5]). Although extremely successful for certain classes of problems, multigrid methods have the notable disadvantage of requiring the generation of a set of coarser grids, which can be difficult to generate for problems defined on complex geometries and unstructured grids. For this reason, we consider algebraic methods of the aggregation type; see [12].

Aggregation provides an automatic way of generating coarse levels and transfer functions to move solutions between the levels. The method has been thoroughly developed for symmetric systems and relies on the idea of generating low energy (or smooth) basis functions that capture the kernel (or near kernel) of the discrete system being solved. In [6] we have shown that aggregation methods can deliver convergence rates comparable to that of geometric multigrid, while being more flexible, for problems defined on structured non-stretched grids.

For problems with anisotropies, a procedure equivalent to the so-called semi-coarsening is required; see [7]. The basic idea of semi-coarsening is that the mesh is only coarsened in directions where smoothing is easily accomplished. Thus, for a problem which has weak coupling in the x direction, coarsening would only be performed in the y and z directions. Algebraic methods can mimic this approach by ignoring connections which are "weak" in the graph coarsening phase. That is, if the coupling between unknowns i and j is ignored, they will not be agglomerated together to define a coarse unknown. However, this strategy fails to deliver the required semi-coarsening if applied to matrices arising from bilinear FE on stretched grids, since all the entries in the computational stencil have comparable value. Without a proper semi-coarsening, the resulting multilevel preconditioner performs poorly on anisotropic problems.

In order to recover semi-coarsening, we proceed as follows. On each level ℓ, we introduce an auxiliary matrix, B_ℓ, defined using some information about the grid, so that the magnitude of the elements of B_ℓ reflects weak and strong connections in the FE problem. Anisotropic aggregates can be constructed using B_ℓ, adopting a conventional dropping technique. B_ℓ is defined using additional information that is usually available in standard finite-element codes. The use of an auxiliary matrix is certainly not new in the geometric multigrid community, although to the best of our knowledge no paper reports on its use with aggregation-based preconditioners.

Several other approaches have been proposed in the literature to improve the coarsening of algebraic multilevel methods. Chow [3] suggested to compute algebraically smoothed vectors, Broker *et al.* [2] proposed to take advantage of SPAI smoothers, and Brezina *et al.* [1] introduced the adaptive smoothed

aggregation technique. Although promising, these techniques all rely on the computation of either a set of slowly converging vectors or SPAI smoothers, which are usually expensive operations.

This paper is organized as follows. Section 2 introduces the multilevel preconditioning algorithm we have adopted. Section 3 describes the proposed procedure to obtain semi-coarsening. Section 4 presents the numerical results, obtained on a distributed parallel computer. Finally, Section 5 outlines the conclusions.

2 Aggregation Multilevel Preconditioner

In this paper we focus on non-smoothed aggregation only, since no general theory is available to define a proper prolongator smoother for non-symmetric equations. The construction of the multilevel hierarchy in non-smoothed aggregation is performed by the following five steps. For each level ℓ (except the coarsest), do:

1. Extract from A_ℓ the graph $\mathcal{G}_\ell$ to coarsen.
2. Coarsen $\mathcal{G}_\ell$ to define a set of aggregates. Each aggregate defines a "grid point" on the coarser level.
3. Define the prolongator P_ℓ and restriction R_ℓ.
4. Compute the next-level matrix $A_{\ell+1}$ as $R_\ell\ A_\ell\ P_\ell$.

We now focus in more details on steps 1 and 2. For systems of equations, we define $\mathcal{G}_\ell$ by "condensing" all the physical unknowns corresponding to the same grid point, thus defining the "block" structure of A_ℓ. Each block has size $m \times m$, m being the number of physical unknowns. The graph coarsening is defined as follows:

$$e_{ij} \text{ is an edge of } \mathcal{G}_\ell \text{ iff} \quad |a_{ij}| \geq \theta \sqrt{|a_{ii}| \cdot |a_{jj}|}. \tag{2}$$

θ is the *threshold*, and $|\cdot|$ is an appropriate matrix norm. The $m \times m$ block elements $a_{i,j}$ that do not fulfill (2) are *dropped* in the construction of $\mathcal{G}_\ell$. A graph decomposition algorithm (such as those in METIS) is then applied to $\mathcal{G}_\ell$. The goal of this algorithm is to define groups of vertices (aggregates) such that each aggregate contains a tightly connected subgraph and so that each vertex is included in just one subgraph. Each aggregate will effectively become an unknown on the coarse mesh. Once the aggregates are defined, the prolongator matrix P_ℓ is constructed such that each row corresponds to a grid point and each column corresponds to an aggregate.

Once the multilevel hierarchy has been establish, an iteration (V-cycle) of the recursive algorithm is as follows. Starting from $\ell = 0$, on each level do:

1. If on the coarsest level, solve with a direct solver and return.
2. Do ν_1 iterations of pre-smoothing S_ℓ^{pre}.
3. Compute the restricted residual $r_{\ell+1} = R_\ell\ r_\ell$.

4. Recursively solve $A_{\ell+1}\, e_{\ell+1} = r_{\ell+1}$.
5. Interpolate error, $e_\ell = P_\ell\, e_{\ell+1}$.
6. Add the correction e_ℓ to the current iterate.
7. Do ν_2 iterations of post-smoothing S_ℓ^{post}.

3 Definition of the Auxiliary Matrix

For problems defined on stretched grids, the distribution of nodes in the stretched direction will correctly represent the low frequencies, whereas, in the direction perpendicular to the stretching, it will represent the high frequencies. The closer two nodes are, the better they will represent the high frequency components of the error. For the problems considered in this paper, the matrix coefficients do not properly reflect the strength of connection between points, while the geometric information does (i.e., points that are geometrically distant from each other have a weak connection between them compared to points that are close to each other). Therefore, we want to form a matrix which captures this geometric information that can be used in the coarsening stage of the algorithm. The basic idea is to create a discrete Laplacian matrix where the size of the off-diagonal entries is related to the distance between points. In particular, we define

$$b_{i,j} = -\frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}, \quad i \neq j, \quad b_{i,i} = \sum_{i \neq j} -b_{i,j},$$

where $\mathbf{x}_k$ represents the coordinates of node k.

B represents the finite-element mesh in the following sense: if a grid node i is "far" from j, then $b_{i,j}$ is "small", while if i is "close" to j, then $b_{i,j}$ is "large". The dropping technique (2) can now be straightforwardly applied to B to produce anisotropic aggregates.

The resulting algorithm for the definition of the multilevel preconditioner reads as follows:

1. Build the auxiliary matrix B_0 using the nodal coordinates $\mathbf{x}$
2. For each level ℓ, do
3. Define the dropping value
4. Build graph $\mathcal{G}_\ell$ based on B_ℓ
5. Create the aggregates using $\mathcal{G}_\ell$
6. Create the tentative prolongators P_ℓ and R_ℓ
7. $B_{\ell+1} = R_\ell B_\ell P_\ell$
8. Destroy B_ℓ
9. EndFor
10. Build a new hierarchy using A and the P_ℓ, R_ℓ previously computed

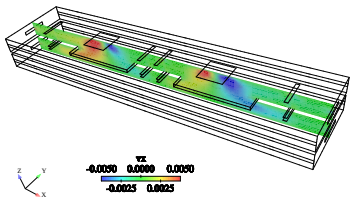

Fig. 1. Steady-state x-component of velocity for model 3D building

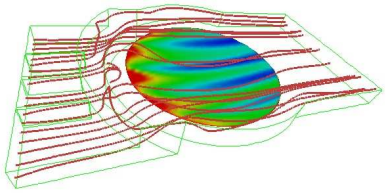

Fig. 2. 3D horizontal CVD reactor with a rotating disk

4 Numerical Results

We apply the algorithm described in Section 3 to the solution of the linear system arising from a stabilized FE discretization of the incompressible Navier-Stokes equations with energy and mass transport; see for instance [11]. The equations in residual form are:

$$R_P = \frac{\delta \rho}{\delta t} + \nabla \cdot (\rho \mathbf{u}) \tag{3}$$

$$\mathbf{R_m} = \rho \frac{\delta \mathbf{u}}{\delta t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) - \nabla \cdot \mathbf{T} - \rho \mathbf{g} \tag{4}$$

$$R_T = \rho C_p \left[\frac{\delta T}{\delta t} + \mathbf{u} \cdot \nabla T \right] + \nabla \cdot \mathbf{q} - \phi - \sum_{k=1}^{N_s} h_k \nabla \cdot \mathbf{j}_k \tag{5}$$

$$R_{Y_k} = \rho \left[\frac{\delta Y_k}{\delta t} + \mathbf{u} \cdot \nabla Y_k \right] + \nabla \cdot \mathbf{j}_k \qquad k = 1, 2, ..., N_s - 1 \tag{6}$$

The FE code used for this work is MPSalsa [11], which uses a parallel Newton-Krylov solver on unstructured meshes. The calculations were performed on the Sandia Cplant machine, composed of nodes with one 500-MHz Dec Alpha processor and 1 GB of RAM, connected together by Myrinet. A classical aggregation procedure has been used to define the aggregates [9]. The smoother is one sweep of Gauss-Seidel (with damping parameter of 0.67) for either the first level or the first two levels, while Aztec's incomplete factorization were adopted for the other levels. The KLU solver of the Amesos [8] library was used to solve the coarse problem. The threshold used in (2) was 0.05.

The first example involves the calculation of fluid flow, without thermal effects, in a simple prototype model of a building. This model represents a two-story building with the floors separated by two atria. Figure 1 shows a typical laminar steady-state solution. The centerline cutting plane shows the x-component of velocity. The worst aspect ratio hexahedral elements have largest dimension that is five times larger than the smallest dimension. We consider laminar steady-state calculations to allow direct-to-steady-state solutions. Seven nonlinear iterations were required to reach convergence.

Table 1 shows an algorithmic scaling study for the steady-state calculations on hexahedral meshes and shows the reduction in iteration count provided by the auxiliary matrix as compared to without the auxiliary matrix. The larger meshes are generated by uniform refinement of previous meshes, with the number of processors being increased to maintain a roughly constant number of unknowns per processor. After each level of uniform refinement of the building geometry, the fine mesh is load-balanced using the ParMETIS graph partitioner through Zoltan [4]. The first three columns present the number of processors and unknowns and nonzeros in the fine level matrix. For both the case with and without auxiliary matrix, the table presents the complexity of the hierarchy (sum of nonzeros of matrices in all levels divided by the nonzeros of the finest matrix), the setup time in seconds, the average linear iterations per Newton step, and the average time per Newton step in seconds.

In the results, the number of unknowns per processor is kept roughly constant, and therefore a perfectly scalable preconditioner would converge in the same number of iterations as the number of processors used in the computation is increased. From Table 1, one can note that using isotropic aggregates with 16 and 128 processors the iterations increase from 57 to 90 (an increment of about 57%), while using anisotropic aggregates the difference is modest (of about 25%). This makes the preconditioner based on the auxiliary matrix nearly scalable in terms of iterations to convergence, but still unsatisfactory from the point of view of CPU time. The large CPU times are due to one of the drawbacks of semi-coarsening: higher complexity. For this example, while isotropic aggregates reduces grid complexity between two consecutive levels by a factor of 27 in 3D, semi-coarsening only achieves a grid complexity reduction of 9 in 3D. This increases the setup and application cost of the resulting multilevel cycle, as well as the time required to compute the ILU factorizations.

proc	fine		5-level (GS/GS/ILU/ILU/KLU)							
	unks	nonzero	no auxiliary matrix				auxiliary matrix			
			complex	setup time (sec)	avg its/ Newt step	time/ Newt (sec)	complex	setup time (sec)	avg its/ Newt step	time/ Newt (sec)
2	227K	22.4M	1.02	3.5	41	120	1.16	6.4	25	92
16	1.70M	175M	1.02	4.3	57	164	1.18	13.7	27	112
128	13.1M	1390M	1.02	8.0	90	434	1.21	30.4	34	264

Table 1. Comparison of five-level preconditioner (GS/GS/ILU/ILU/KLU) with and without auxiliary matrix for 3D model building; uncoupled aggregation; Cplant machine.

The second example involves the deposition of poly-Silicon in a rotating disk chemical vapor deposition (CVD) reactor. A mixture of trichlorosilane

($SiCl_3H$), HCl, and H_2 enters from the four inlets on the left, flows over a forward facing step, and over an inset rotating disk, depositing silicon on the wafer. Chemical reactions occur on the surface of the disk only and not in the flow. Figure 2 shows a schematic of the CVD reactor. A contour plot of poly-silicon deposition rate on the disk is shown, along with representative streamlines of the flow through the reactor.

Table 2 shows a scaling study of a simple continuation step where the thermodynamic pressure was increased from 0.6 to 0.85 atmospheres and the inlet flow velocity from 30 cm/sec to 35 cm/sec. The worst aspect ratio hexahedral element has largest dimension that is about a factor of ten larger than the smallest dimension. This table shows a comparison of the 1-level DD ILU preconditioner with the 5-level preconditioners with and without the auxiliary matrix. The smoothers for the 5-level preconditioner were one sweep of Gauss-Seidel on the finest level with damping parameter of 0.67 and ILU on the next three levels. Non-restarted GMRES was used with a linear solve convergence criterion of 3×10^{-4}. From the table, one can see that the auxiliary matrix has improved the iteration count, while the CPU time is only marginally reduced. This situation might be improved by using GS on the second level as in the previous example.

proc	fine unks	1-level		5-level (GS/ILU/ILU/ILU/KLU)					
				no auxiliary matrix			auxiliary matrix		
		avg its/ Newt step	time (sec)	complex	avg its/ Newt step	time/ Newt (sec)	complex	avg its/ Newt step	time/ Newt (sec)
2	87400	49	125	1.01	75	118	1.06	48	103
16	636K	95	183	1.02	107	168	1.12	66	141
128	4.85M	221	409	1.02	164	319	1.19	97	313

Table 2. Comparison of five-level preconditioner (GS/ILU/ILU/ILU/KLU) with and without auxiliary matrix for CVD reactor; uncoupled aggregation; Cplant machine.

5 Conclusions

In this paper we presented the application of a multilevel preconditioner for the parallel solution of large, sparse linear systems for FE discretizations on stretched grids. We concentrated on the coarsening process. In order to improve the performance of our preconditioner, we introduced an auxiliary matrix, which contains information about the underlying finite-element grid. The coarsening is performed on an auxiliary matrix, then the final hierarchy is rebuilt on the linear system matrix.

By resorting to an auxiliary matrix, anisotropic aggregation can be constructed at a negligible computational cost. The reported preliminary numerical results, obtained on a distributed parallel computer, show that the proposed approach can significantly improve the performance of the algebraic multilevel preconditioner in terms of iterations to convergence. Although more effective, the preconditioner, of higher complexity, is more expensive to construct and to apply.

Using anisotropic aggregates, the CPU time is significantly reduced for linear systems arising from the discretization of the incompressible Navier-Stokes equations, while for chemically reacting flows the results are less satisfactory. These preliminary results are encouraging although much more work on a broader range of numerical tests is required.

References

1. M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation (αSA). *SIAM Journal on Scientific Computing*, 25(6):1896–1920, 2004.
2. O. Bröker, M. J. Grote, C. Mayer, and A. Reusken. Robust parallel smoothing for multigrid via sparse approximate inverses. *SIAM Journal on Scientific Computing*, 23(4):1396–1417, 2001.
3. E. Chow. An unstructured multigrid method based on geometric smoothness. *Numer. Linear Algebra Appl.*, 10:401–421, 2003.
4. Karen Devine, Erik Boman, Robert Heaphy, Bruce Hendrickson, and Courtenay Vaughan. Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, 4(2):90–97, 2002.
5. W. Hackbusch. *Multigrid methods and applications*, volume 4 of *Computational Mathematics*. Springer–Verlag, Berlin, 1985.
6. P.T. Lin, M. Sala, J.N. Shadid, and R.S. Tuminaro. Performance of fully-coupled algebraic multilevel preconditioners for incompressible flow and transport. *submitted to International Journal for Numerical Methods in Engineering*, 2004.
7. E. Morano, D.J. Mavriplis, and V. Venkatakrishnan. Coarsening strategies for unstructured multigrid techniques with applications to anisotropic problems. *SIAM Journal on Scientific Computing*, 20(2):393–415, 1998.
8. M. Sala. Amesos 2.0 reference guide. Technical Report SAND2004-4820, Sandia National Laboratories, September 2004.
9. M. Sala, J. Hu, and R. Tuminaro. ML 3.1 smoothed aggregation user's guide. Technical Report SAND2004-4819, Sandia National Laboratories, September 2004.
10. J.N. Shadid. A fully-coupled Newton-Krylov solution method for parallel unstructured finite element fluid flow, heat and mass transfer simulations. *Int. J. CFD*, 12:199–211, 1999.
11. J.N. Shadid, S.A. Hutchinson, G.L. Hennigan, H.K. Moffet, K.D. Devine, and A.G. Salinger. Efficient parallel computation of unstructured finite element reacting flow solutions. *Parallel Computing*, 23:1307–1325, 1997.
12. P. Vaněk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik*, 88:559–579, 2001.