CHAPTER 33

Solution of the Stokes Problem on Distributed-Memory Multiprocessors

Lois Mansfield*

Abstract

A combination of damped Jacobi preconditioning and coarse grid deflation is proposed to precondition the conjugate gradient solution of the Stokes problem on distributed-memory multiprocessors. Experimental results are given for the driven cavity problem on the iPSC/2 and iPSC/860 with both 4 and 16 processors.

1 Introduction

We consider the finite element solution of the Stokes problem

$$-\Delta \mathbf{u} + \nabla p = f \quad in\,\Omega, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad in\,\Omega, \tag{2}$$

on distributed-memory multiprocessors. Here Ω is a bounded domain in R^2, $\mathbf{u} = (u_1, u_2)^T$ is the velocity, and p is the pressure. In addition, one has boundary conditions for $\mathbf{u}$. The simplest is

$$\mathbf{u} = \mathbf{g} \quad on\,\partial\Omega, \tag{3}$$

but at outflow it is often better to specify boundary conditions which can be expressed as natural boundary conditions associated with the variational formulation used in the finite element approximation.

Besides being important in its own right, the solution of the Stokes problem is important in the solution of the time dependent Navier-Stokes equations when time splitting is used to separate the incompressibility condition $div\,\mathbf{u} = 0$ from the nonlinearity of the convection term $(\mathbf{u} \cdot \nabla)\mathbf{u}$.

To discretize by finite elements, one introduces finite dimensional spaces $\mathbf{V}^h$ and P^h. These spaces should satisfy the compatibility condition

$$\sup_{\mathbf{v} \epsilon \mathbf{V}^h} \frac{\int_\Omega div\,\mathbf{v}\,p\,dx}{||\mathbf{v}||_{H^1}||p||_{L^2}} \geq C, \tag{4}$$

for all $p \epsilon P^h$.

The Stokes problem has the algebraic form

*Department of Applied Mathematics, University of Virginia, Charlottesville, Va. 22903. This work was supported by NSF Grant ASC 8922539.

$$\begin{pmatrix} K & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}. \tag{5}$$

If the first system in (5) is solved for u and substituted into the second system, one obtains

$$Cp = B^T K^{-1} B p = -B^T K^{-1} f. \tag{6}$$

The matrix C is unattractive to set up regardless of what type of computing machine is being used, so it is natural to use an iterative method like conjugate gradient which only requires that one be able to compute Cq for any vector q. To do an evaluation of C times q, it is necessary to solve a system

$$Kx = b. \tag{7}$$

where K is a block Poisson matrix. On a conventional machine, one would probably factor K so that only a forward and backward solve is required to solve (7).

On a distributed-memory multiprocessor computer, triangular solves are very inefficient. Instead, the region Ω can be partitioned into as many subregions as processors and ordered as

$$\begin{pmatrix} G & V \\ V^T & \overline{F} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}, \tag{8}$$

where G contains all rows and columns associated with internal nodes on any processor and $\overline{F}$ contains all rows and columns associated with the separator nodes. One way to solve (8) is to solve the Schur complement system

$$Fx_1 = g_1, \tag{9}$$

where $F = \overline{F} - V^T G^{-1} V$ and $g_1 = b_1 - V^T G^{-1} b_0$, and then determine x_0 from

$$Gx_0 = b_0 - Vx_1. \tag{10}$$

One can solve (10) by direct methods and (9) by conjugate gradient.

In [5] we proposed a combination of damped Jacobi preconditioning and coarse grid deflation to improve the convergence of conjugate gradient iteration on distributed-memory multiprocesor computers. We showed that damped Jacobi preconditioning has much of the same effect that smoothing has in multigrid iteration in that it reduces the large eigenvalues of the matrix to nearly 1. We showed that coarse grid deflation has the effect of raising the lower bound on the eigenvalues of the system.

We propose as one way to solve (5), to use conjugate gradient iteration to solve (6). To solve the systems (7), we propose ordering (7) as (8) and solving (10) by direct methods and (9) by conjugate gradient using a combination of damped Jacobi preconditioning and coarse grid deflation. For C the diagonal is not readily available so damped Jacobi preconditioning can't be used. However, it is shown in [2] that the compatibility condition (4) implies that C is spectrally equivalent to the mass matrix M for the basis elements of P^h. This suggests that a good preconditioner for C is a diagonal matrix which is spectrally equivalent to M such as the lumped mass matrix L obtained by summing the elements in each row of M. We have found that in some cases the techniques of [5] are still useful here, where we replace damped Jacobi preconditioning by damped Richardson preconditioning applied to $L^{-1}C$.

One might hope to save work if the conjugate gradient solution of (9) isn't done to convergence. If we don't want to iterate the conjugate gradient solution of (9), and of (6), to convergence, we could treat the above procedure as a preconditioner for some iterative scheme applied to (5). The scheme which perhaps is simplest and easiest to analyze is

$$\begin{pmatrix} u_{i+1} \\ p_{i+1} \end{pmatrix} = \begin{pmatrix} u_i \\ p_i \end{pmatrix} + \begin{pmatrix} \hat{K} & -B \\ -B^T & \hat{T} \end{pmatrix}^{-1} [\begin{pmatrix} f \\ 0 \end{pmatrix} - \begin{pmatrix} K & -B \\ -B^T & 0 \end{pmatrix} \begin{pmatrix} u_i \\ p_i \end{pmatrix}] \tag{11}$$

introduced in [2], where $\hat{K}$ represents several steps of an iterative method applied to K and $\hat{T} = B^T \hat{K}^{-1} B - \hat{C}$, where $\hat{C}$ represents several steps of an iterative method applied to C.

In Section 3, we present experimental results when our combination of coarse grid deflation and either damped Jacobi or damped Richardson iteration is used to solve (9) and (6), or are used to obtain a preconditioner for (11). Our experiments are carried out on the iPSC/2 at the University of Virginia and the iPSC/860 at Oak Ridge National Laboratories with either 4 or 16 processors. The problem we solve is the driven cavity problem using both a uniform grid and an irregular grid with points concentrated at the corners. In Section 2, we give a brief description of damped Jacobi preconditioning and coarse grid deflation.

2 Damped Jacobi preconditioning and coarse grid deflation

As a preconditioner for conjugate gradient iteration applied to the system $Au = f$, damped or underrelaxed Jacobi preconditioning results from applying one or more steps of damped or underrelaxed Jacobi iteration to $A\hat{r} = r$ with initial approximation $\hat{r}^{(0)} = 0$. Damped Jacobi iteration may be described as

$$\hat{r}^{(m+1)} = (I - \alpha D^{-1}A)\hat{r}^{(m)} + \alpha D^{-1}r.$$

where D is the diagonal of A. Ordinary Jacobi iteration is obtained by setting $\alpha = 1$. If M^{-1} is chosen to represent m steps of damped Jacobi iteration

$$M^{-1} = (I + I - \alpha D^{-1}A + (I - \alpha D^{-1}A)^2 + \cdots + (I - \alpha D^{-1}A)^{m-1})\alpha D^{-1}.$$

Similarly to Theorem 1 of [1], it can be shown that for m odd, M is always positive definite since D is positive definite, and for m even, M is positive definite if and only if $2D - \alpha A$ is positive definite. Note that the requirement for damped Jacobi iteration to be convergent (see [9], Theorem 3.3) is that $2D - \alpha A$ and A be positive definite. Of course, $2D - \alpha A$ can be made to be positive definite by appropriate choice of α.

The choice $m = 1$ essentially amounts to diagonal preconditioning. As is shown in [1] the benefits of taking more steps of an m-step preconditioner decreases as m increases. We have used $m = 2$ and will limit discussion to this case. For $m = 2$

$$M^{-1}A = \alpha(2I - \alpha D^{-1}A)D^{-1}A.$$

The eigenvalues of $M^{-1}A$ are given by

$$\gamma_i = \alpha(2 - \alpha\lambda_i)\lambda_i, \quad i = 1, ..., n,$$

where λ_i, $i = 1, ..., n$, are the eigenvalues of $D^{-1}A$.

$$\gamma = \alpha(2 - \alpha\lambda)\lambda \tag{12}$$

is a parabolic curve in λ. If α is taken to be $2/\lambda_n$, where λ_n is the largest eigenvalue of $D^{-1}A$, (12) is zero at 0 and λ_n and reaches its maximum at $\lambda_n/2$. This corresponds to ordinary Jacobi iteration. By taking α smaller, the point at which (12) reaches its maximum and its right zero are pushed to the right. This likely increases the number and moves to the higher end of the spectrum, the eigenvalues λ_i of $D^{-1}A$ which have their corresponding γ_i close to 1. The best choice appears to be $\alpha = 4/(3\lambda_n)$. A graph illustrating this is given in [5].

If the diagonal is not available, as in the case of the Schur complement matrix C in (6), damped Jacobi preconditioning can't be used. Instead we propose to use damped Richardson preconditioning. If $\tilde{M}^{-1}$ represents two steps of damped Richardson iteration,

$$\tilde{M}^{-1}A = \alpha(2I - \alpha A)A$$

so that α should be chosen to be $\alpha = 4/(3\tilde{\lambda}_n)$, where $\tilde{\lambda}_n$ is the largest eigenvalue of A.

Thus we propose to use damped Jacobi preconditioning or damped Richardson preconditioning much as smoothing is used in multigrid iteration, to handle the large eigenvalues of $D^{-1}A$ or A. We propose to combine this with coarse grid deflation, which attacks the smaller eigenvalues. The deflation method of Nicolaides [6] to solve $Au = f$ where A is an nxn symmetric positive definite matrix may be described as follows. Let E be an nxm matrix whose columns are a set of linearly independent vectors. First solve $Au = f$ in the subspace $\mathcal{E}$ generated by the columns of E. This amounts to solving

$$E^T AEd = E^T f. \tag{13}$$

Then solve

$$A(I - P)v = f - AEd \tag{14}$$

in $span\{E\}^{\perp}$, where P is the projection operator

$$P = E(E^T AE)^{-1} E^T A, \tag{15}$$

which is the orthogonal projector in the norm

$$\|x\|_A = (x^T Ax)^{1/2}.$$

Then

$$u = Ed + (I - P)v. \tag{16}$$

The effect of (13) is to find the best approximation in the norm $\|\cdot\|_A$ to the solution u to $Au = f$ which can be found among linear combinations of the columns of E. Since m steps of conjugate gradient iteration does the same thing over the Krylov subspace (A, r^0, m), the effectiveness of deflation depends on the solution being better approximated by the columns of E than the Krylov subspace. In [3] we proposed defining the columns of E in certain situations to reflect known qualitative features of the solution such as being nearly constant along certain lines in the domain. More generally, we propose to define the columns of E using a nodal basis for a coarse grid over the domain. The matrix E then defines an interpolation operator mapping the coarse grid to the problem grid.

In [4] we showed that if the system (1) arises from the finite element discretization of a second order elliptic partial differential equation or system of second order elliptic partial differential equations satisfying standard ellipticity assumptions, and if the finite element grid is quasi-uniform, the effect of replacing the Schur complement matrix C by $C(I - P)$ is to improve the bound on the condition number by raising the lower bound on the smallest positive eigenvalue. Likewise the effect of replacing the matrix A by $A(I - P)$ is also to raise the lower bound on the smallest positive eigenvalue. The matrix $A(I - P)$ is also the Schur complement matrix obtained when the usual nodal basis is replaced by the union of a nodal basis over a coarse grid along with a space consisting of functions vanishing at the nodes of the coarse grid. Such a decomposition was proposed to solve indefinite problems in [8].

3 Experimental Results

We consider the driven cavity problem. The region Ω is the unit square. The boundary conditions are $\mathbf{u} = 0$ on the bottom and sides of the square and $u_1 = 1$, $u_2 = 0$ on the top of the square. A rectangular grid was used. We used piecewise biquadratic polynomials to approximate the velocity and piecewise bilinear polynomials to approximate the pressure. It was shown in [7] that these finite element spaces satisfy the compatibility condition (4). For 4 processors the region Ω was divided into 4 rectangles each half as long and wide as Ω. For 16 processors the region Ω was divided into 16 rectangles each one fourth as long and wide as Ω.

We consider both a regular grid over Ω containing 16^2 or 32^2 rectangles and an irregular grid containing 20^2 rectangles. For the irregular grid the rectangles are obtained from x_i, $1 - x_i$, and y_i, $1 - y_i$, where the x_i and y_i are given by 0, .02, .04, .08, .12, .16, .20, .25, .30, .40, .50.

We first solve the Stokes problem by solving (6) by conjugate gradient iteration, where the Poisson systems (7) are solved to convergence by ordering them as (8) and solving (10) by direct methods and (9) by conjugate gradient iteration. Our results are given in Tables 1 and 2. SDEF9M indicates that deflation with 9 coarse grid basis functions along with damped Jacobi preconditioning was used in the solution of (9) and that (6) was preconditioned with the lumped mass matrix L. SDEF9SM uses this same preconditioning for the solution of (9) but preconditions (6) with damped Richardson preconditioning applied to $L^{-1}C$. SDEF33SM uses a finer coarse grid containing 33 basis functions. SDEF33NDSM orders the columns of E in this coarse grid by nested dissection to decrease the number of arithmetic operations required. In the last two entries in Table 2, no preconditioning was used in the solution of (9) but deflation with a coarse grid containing 9 basis functions was used to precondition either $L^{-1}C$ or the matrix obtained from damped Richardson preconditioning applied to $L^{-1}C$. The convergence criteria were that the l^2 norm of the original residual be reduced by a factor of 10^6 for (6) and that the l^2 norm of the residual be less than 10^{-9} for (9).

In all cases we set up the Schur complement matrix F explicitly. To do the deflation applied to F, we first orthogonalized the columns of E with respect to the inner product x^TFy to obtain V so that V^TFV is the identity matrix. The vectors in FV were retained in memory to facilitate the evaluation of $F(I-P)p$. This eliminated the need to set up E^TFE. Deflation applied to $L^{-1}C$ was done in the same way. The number of operations required to orthogonalize the columns of E and evaluate $(I-P)p$ can be reduced if the nodes in the coarse grid are ordered by nested dissection. To determine α we estimated the largest eigenvalue of $D^{-1}F$ or $L^{-1}C$ by the power method.

The timings we give are the solution times. Not included is the time to set up the matrix K, reorder it as in (8), and set up F. Also the time to orthogonalize the columns of E is not included. On 16 processors this setup time varied from about 4% to 20% of the solution time, depending on the size of the problem and the machine used. On 4 processors it was as high as 30% of the solution time on the iPSC/860. If the Stokes problem is to be solved repeatedly in a time splitting for the time dependent Navier-Stokes equations, the setup time is even less significant.

The results seem to indicate that damped Richardson preconditioning for $L^{-1}C$ is worthwhile for 16 processors. The effect of damped Jacobi or damped Richardson preconditioning and deflation overall, however, was less than when applied to the cantilevered beam problem, where their combined use would typically reduce the number of iterations required by about a factor of four. Experimental results for the cantilevered beam problem are given in [5]. This greater success for the cantilevered beam problem may be because it is quite ill-conditioned. Deflation did not seem to be very effective when applied to $L^{-1}C$, perhaps because it already is a well-conditioned matrix.

The timings in Table 2 indicate that it is hardly worthwhile increasing from 4 to 16 processors, particularly on the iPSC/860. We think this is due to the fact that the work each processor has to do in each iteration to solve (9) is proportional to the number of separators coincident with its subregion. Each subrectangle in the 4 processor decomposition has 2 sides while interior subrectangles in the 16 processor decomposition have 4 sides. This means that for the same size problem the amount of work per processor for that phase of the computation remains the same. We expect to see reasonable speedup when the number of processors is increased beyond 16, but haven't had a chance to do the experimentation to confirm this yet. The values given for the numbers of iterations to solve (9) are totals summed over the whole solution process. We have found that the number of iterations required to solve (9) decreases significantly as one nears convergence of (6).

We now consider solving the Stokes problem using (11). Here (9) is not solved to convergence in each iteration and the previous procedure then is a preconditioner for a Richardson type iteration on (5). In our experiments, we defined $\hat{C}$ in (11) by stopping the iteration after N_1 iterations or after our previous convergence criterion was passed. Likewise we defined $\hat{K}$ in (11) by stopping the iteration after N_2 iterations or after our previous convergence criterion was passed. We picked N_1 and N_2 using our experience gained from our previous experiments. In [2], the more sensitive number N_1 was picked by estimating $\|I-\hat{K}^{-1}K\|_2$ and using the theory developed.

We found that if N_1 and N_2 were not chosen sufficiently large, the algorithm would report convergence prematurely. Thus unless the iteration (11) is used carefully it doesn't seem to be robust. For the preconditionings of SDEF9M and $N_1 = 20$, $N_2 = 5$ with the regular 16x16 grid on 16 processors, we obtained convergence on the iPSC/2 in 58.18 seconds. The total numbers of iterations of (6) and (9) were 43 and 463. This is better than the result reported in Table 1. For the regular 32x32 grid the best we were able to achieve was convergence in 388.91 seconds with

	16x16 289 eqns in (6) 177 eqns in (9)	32x32 1089 eqns in (6) 369 eqns in (9)
no precond.	57,1742 (66.75,10.36)	63,1646 (251.11,25.54)
SDEF9M	37,633 (65.99,9.91)	42,965 (278.88,30.90)
SDEF9SM	23,530 (62.41,9.01)	24,710 (246.86,26.14)
SDEF33SM	23,319 (76.19,9.88)	24,465 (273.67,27.31)
SDEF33NDSM	23,319 (54.60,7.87)	24,465 (220.12,22.72)

Table 1: Iterations and Timings for 16 processors on regular grid. Numbers are number of iterations required to solve (6) and total number of iterations required to solve (9). Numbers in parentheses are timings in seconds on the iPSC/2 and iPSC/860.

	4 processors 441 eqns in (6) 77 eqns in (9)	16 processors 441 eqns in (6) 225 eqns in (9)
no precond.	no conv.	no conv.
SDEF9M	38,688 (169.50,14.51)	38,785 (105.81,13.74))
SDEF9SM	22,525 (177.24,14.66)	23,612 (94.20,12.21)
SDEF33SM	-	22,432 (115.81,13.60)
SDEF33NDSM	-	23,432 (88.08,11.47)
MD	36,2011 (218.23,19.49)	-
SMD	22,1578 (230.08,19.71)	-

Table 2: Iterations and Timings for 4 and 16 processors on irregular grid. Numbers are number of iterations required to solve (6) and total number of iterations required to solve (9). Numbers in parentheses are timings in seconds on the iPSC/2 and iPSC/860.

$N_1 = 25$, $N_2 = 8$. The total numbers of iterations of (6) and (9) were 78 and 1071. This was far worse than the result reported in Table 1.

References

[1] L. Adams. *m-step preconditioned conjugate gradient methods,* SIAM J. Sci. Stat. Comput. 6 (1985) , 452-463.

[2] R. E. Bank, B. D. Welfert,and H. Yserentant. *A class of iterative methods for solving saddle point problems,* Numer. Math. 56 (1990), 645-666.

[3] L. Mansfield. *On the use of deflation to improve the convergence of conjugate gradient iteration,* Commun. Appl. Numer. Methods 4 (1988), 151-156.

[4] L. Mansfield. *On the conjugate gradient solution of the Schur complement system obtained from domain decomposition,* SIAM J. Num. Anal. 27 (1990), 1612-1620.

[5] L. Mansfield. *Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers,* SIAM J. Sci. Stat. Comput. (to appear).

[6] R. A. Nicolaides. *Deflation of conjugate gradients with applications to boundary value problems,* SIAM J. Num. Anal. 24 (1987), 355-365.

[7] R. Stenberg. *Analysis of mixed finite element methods for the Stokes problem: a unified approach,* Math. Comp. 38 (1984), 9-23.

[8] H. Yserentant. *On the multi-level splitting of finite element spaces for indefinite elliptic boundary value problems,* SIAM J. Num. Anal. 23 (1986), 581-595.

[9] D. M. Young. Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.