Technical Report

TR-2007-002

Block preconditioning for saddle point systems with indefinite (1,1) block

by

Michele Benzi, Jia Liu

Mathematics and Computer Science

EMORY UNIVERSITY

International Journal of Computer Mathematics
Vol. 00, No. 05, May 2007, 1–16

Block preconditioning for saddle point systems with indefinite $(1,1)$ *block*

MICHELE BENZI*† AND JIA LIU‡

†Department of Mathematics and Computer Science,
Emory University, Atlanta, GA 30322, USA

‡Department of Mathematics and Statistics,
University of West Florida, Pensacola, FL 32514, USA
(Received 00 Month 200x; In final form 00 Month 200x)

We investigate the solution of linear systems of saddle point type with an indefinite $(1,1)$ block by preconditioned iterative methods. Our main focus is on block matrices arising from eigenvalue problems in incompressible fluid dynamics. A block triangular preconditioner based on an augmented Lagrangian formulation is shown to result in fast convergence of the GMRES iteration for a wide range of problem and algorithm parameters. Some theoretical estimates for the eigenvalues of the preconditioned matrices are given. Inexact variants of the preconditioner are also considered.

1 Introduction

In this paper we begin a study of preconditioning techniques for generalized saddle point systems of the form

$$\begin{pmatrix} A-\beta M & B^T \\ B & O \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}, \qquad \text{or} \qquad \mathcal{A}\mathbf{x} = \mathbf{b}\,, \tag{1}$$

where $A, M \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times n}$ with $m \le n$ and $\beta \in \mathbb{R}$. We make the following assumptions on A, M, B and $\mathcal{A}$:

- A is positive real; that is, the matrix $H = \frac{1}{2}(A + A^T)$, the symmetric part of A, is positive definite.
- M is symmetric positive definite (it can be thought of as a mass matrix, or in same cases as the $n \times n$ identity matrix).
- B has full row rank.

*Corresponding author. Email: benzi@mathcs.emory.edu

- $A - \beta M$ is indefinite, in the sense that it has eigenvalues on either side of the imaginary axis (this implies $\beta > 0$ and "sufficiently large").
- $A - \beta M$ and $\mathcal{A}$ are both invertible.

We note that the matrix A may itself be symmetric in some applications. However, in this paper we do not assume $A = A^T$. Linear systems of the type (1) will be referred to as (generalized) "saddle point systems with indefinite $(1,1)$ block." Such linear systems arise in various areas of scientific computing, including the solution of eigenvalue problems in fluid mechanics [8,13] and electromagnetics [2] by shift-and-invert algorithms, and in certain time-harmonic wave propagation problems [12, 15]. We emphasize that while numerous effective solution algorithms exist for the case of a positive definite or semidefinite $(1,1)$ block (corresponding to either $\beta \leq 0$ or $\beta > 0$ but smaller than the real part of the eigenvalue of A of smallest magnitude), see [3, 7, 9], relatively little has been done for the case where the $(1,1)$ block is indefinite. Generally speaking, this is a rather challenging problem, which gets harder as the matrix $A - \beta M$ becomes more indefinite.

It should be noted that for linear systems arising in the solution of the generalized eigenvalue problem

$$\begin{pmatrix} A & B^T \\ B & O \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} M & O \\ O & O \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{2}$$

the parameter β in (1) approximates a generalized eigenvalue λ (that is, $\beta \approx \lambda$), making the coefficient matrix $\mathcal{A}$ close to singular and therefore highly ill-conditioned. In most cases β will be an approximation to an eigenvalue of $\mathcal{A}$ close to the imaginary axis (in other words, to one of the eigenvalues of $\mathcal{A}$ with smaller real part), making the $(1,1)$ block only mildly indefinite. When β approximates eigenvalues of $\mathcal{A}$ that are closer to the middle of the spectrum, however, the $(1,1)$ block becomes more indefinite and problem (1) becomes harder to solve.

In this paper we experiment with a block triangular preconditioner based on an augmented Lagrangian formulation of (1), focusing on matrices arising from the discretization of incompressible fluid flow problems. Our experiments show that the preconditioner results in fast convergence of the preconditioned GMRES method for a wide range of problem parameters including the viscosity, the mesh size, and the value of the shift β. As the exact application of the preconditioner can be expensive we also experiment with the inexact case, in which the preconditioner solves are performed iteratively and terminated when some prescribed accuracy is reached. Our tests indicate that for most cases no significant degradation of the rate of convergence results from the inexact application of the preconditioner, and that the robustness with respect to problem parameters is generally preserved. The question of how to approximately apply the preconditioning operator in an efficient and robust manner, however, remains open and necessitates further study.

The remainder of the paper is organized as follows. In section 2 we introduce the augmented Lagrangian formulation and the corresponding block triangular preconditioner. In section 3 we provide some theoretical analysis of the exact variant of the preconditioner. Numerical experiments (for both the exact and inexact form of the preconditioner) are presented in section 4, and conclusions are given in section 5.

2 The augmented Lagrangian preconditioner

Let us define $A_\beta := A - \beta M$. The original linear system (1) is equivalent to the following *augmented Lagrangian* formulation [11]:

$$\begin{pmatrix} A_\beta + \gamma B^T B & B^T \\ -B & O \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c + \gamma B^T d \\ -d \end{pmatrix}, \qquad \text{or} \qquad \mathcal{A}_\gamma \mathbf{x} = \mathbf{b}_\gamma \,, \tag{3}$$

where γ is a positive scalar. The minus sign that now appears in the second block row is not essential, but it will be used henceforth. Note that for $\beta \leq 0$ or $\beta > 0$ and sufficiently small, the spectrum of the coefficient matrix of (3) is entirely contained in the right half-plane; see, e.g., [3, 6].

For the case $\beta = 0$, it was shown in [5] that a block triangular preconditioner of the type

$$\mathcal{P}_\gamma = \begin{pmatrix} A_\beta + \gamma B^T B & B^T \\ O & \frac{1}{\gamma} I \end{pmatrix} \tag{4}$$

(with $A_\beta = A_0 = A$) results in very fast convergence of preconditioned Krylov iterations applied to linear systems of the form (3) arising from stable finite element discretizations of the Oseen problem (linearized Navier–Stokes equations). The preconditioner was shown to be very robust with respect to both the mesh size h and the viscosity ν. Moreover, the quality of the preconditioner was not significantly affected when linear solves with the $(1, 1)$ block of (4) were performed inexactly via a single W-cycle of a specially developed multigrid method. It was also shown in [5] that $\gamma \approx 1$ gave sufficiently good results in many cases, although in a few situations the best overall results were obtained using smaller values of γ (up to about $\gamma \approx 0.02$). We note that in the Oseen problem, the matrix A is nonsymmetric. Similar ideas have been independently investigated by other researchers in order to develop block preconditioners for *symmetric* problems in other application areas; see, e.g., [12, 15].

In this paper, we study the performance of a similar augmented block triangular preconditioner on the indefinite system (3) for *nonzero* values of β. The case $\beta < 0$ arises in the solution of unsteady problems: in this case the block triangular preconditioner performs extremely well, even when applied inexactly. Since this is a relatively easy case, here we focus instead on the more challenging case where $\beta > 0$, and sufficiently large so as to make A_β indefinite.

It follows from the identity

$$\mathcal{P}_\gamma^{-1} = \begin{pmatrix} (A_\beta + \gamma B^T B)^{-1} & O \\ O & I_m \end{pmatrix} \begin{pmatrix} I_n & B^T \\ O & -I_m \end{pmatrix} \begin{pmatrix} I_n & O \\ O & -\gamma I_m \end{pmatrix} \tag{5}$$

that the action of the preconditioner on a given vector requires one application of $(A_\beta + \gamma B^T B)^{-1}$ and one sparse matrix-vector product with B^T. Clearly, the main issue is how to solve linear systems with coefficient matrix $A_\beta + \gamma B^T B$. For large problems these have to be solved by an inner iterative method. Even though the inner solves need not be performed to high accuracy, developing a robust and efficient iterative method for such problems is a non-trivial task. Note that in the case of incompressible flow problems (discretized Stokes and Oseen equations), the introduction of the additional term $\gamma B^T B$ in the $(1, 1)$ block of the saddle point matrix results in a coupling between the components of the velocity vector. For the *definite* case $\beta = 0$, an effective multigrid methods has been developed in [5]. The applicability of such method in the indefinite case $\beta > 0$ is questionable, unless β is sufficiently small. We will further discuss the issue of inexact solves in the section on numerical experiments.

To conclude this section, note that the augmented Lagrangian formulation (3) with γ taken sufficiently large makes the $(1, 1)$ block $A_\beta + \gamma B^T B$ less asymmetric and indefinite; indeed, in the limit as $\gamma \to \infty$ the symmetric positive semidefinite contribution $\gamma B^T B$ will dominate the $(1, 1)$ block. We will also show in the next section that convergence of preconditioned Krylov iterations can be expected to be fast for large values of γ, since in this case all the eigenvalues of the preconditioned matrix $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$ are tightly clustered around 1. On the other hand, a very large value of γ is likely to make the block $A_\beta + \gamma B^T B$ very ill-conditioned and therefore difficult to invert; see the discussion in [11, Remark 2.4]. Hence, the choice of the algorithmic parameter γ involves a trade-off.

3 Spectral properties of the preconditioned matrices

Characterizing the rate of convergence of nonsymmetric preconditioned iterations can be a difficult task. In particular, eigenvalue information alone may not be sufficient to give meaningful estimates of the convergence rate of a method like preconditioned GMRES. The situation is even more complicated for a method like BiCGStab, for which virtually no convergence theory exists. Nevertheless, experience shows that for many linear systems arising in practice, a well-clustered spectrum (away from zero) usually results in rapid convergence of the preconditioned iteration.

Here we develop some estimates for the eigenvalues of the preconditioned matrix $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$, assuming exact solves for the $(1,1)$ block. We show that for this "ideal" version of the preconditioner, under some fairly mild assumptions the eigenvalues of the preconditioned matrix become tightly clustered around 1 as $\gamma \to \infty$. Our analysis makes use of the following simple Lemma, which is a straightforward consequence of [10, Exercise 12.12]; see also [5, 14].

LEMMA 3.1 *Let the matrices $BA_\beta^{-1}B^T$ and $B(A_\beta + \gamma B^T B)^{-1} B^T$ be well-defined and invertible. Then*

$$\left[B(A_\beta + \gamma B^T B)^{-1} B^T\right]^{-1} = \left(BA_\beta^{-1}B^T\right)^{-1} + \gamma I_m \,. \tag{6}$$

It is worth noting that because the matrix A_β is not in general positive real, the invertibility of the various Schur complements must be assumed. In practical computations, however, the invertibility of the relevant matrices was never an issue.

A straightforward calculation shows that

$$\mathcal{A}_\gamma \, \mathcal{P}_\gamma^{-1} = \begin{pmatrix} I_n & O \\ -B(A_\beta + \gamma B^T B)^{-1} & \gamma S_\gamma \end{pmatrix}, \tag{7}$$

where $S_\gamma = B(A_\beta + \gamma B^T B)^{-1} B^T$. Hence, the preconditioned matrix has the eigenvalue 1 of multiplicity n and, by Lemma 3.1, the remaining m eigenvalues λ_i $(1 \le i \le m)$ are of the form $\lambda_i = \frac{\gamma}{\mu_i^{-1} + \gamma}$ where $\mu_i \in \mathbb{C}$ is the ith eigenvalue of $BA_\beta^{-1}B^T$. Since the μ_i's are independent of γ, it follows that the non-unit eigenvalues of $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$ tend to 1 for $\gamma \to \infty$. Notice that this requires that $\gamma \ne -\mu_i$, for $1 \le i \le m$, however, this condition is automatically satisfied under the assumptions of Lemma 3.1. Thus, we have the following result.

PROPOSITION 3.2 *Under the assumptions of Lemma 3.1 the spectrum of the preconditioned matrix $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$ consists of the eigenvalue 1 with multiplicity n, with the remaining m eigenvalues satisfying $\lambda_i(\gamma) \to 1$ for $\gamma \to \infty$. Therefore,* *the spectrum of $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$ is tightly clustered around 1 for large values of γ.*

In practice, as already mentioned, it is desirable to use only moderately large values of γ. From the expression $\lambda_i = \frac{\gamma}{\mu_i^{-1}+\gamma}$ we see that if an eigenvalue μ_i of $BA_\beta^{-1}B^T$ is large in absolute value, the corresponding eigenvalue λ_i of $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$ is close to 1 even for small values of γ; on the other hand, if $\mu_i \approx 0$ then γ must be taken large in order to have $\lambda_i \approx 1$. Thus, it would be useful to have some idea of when the matrix $BA_\beta^{-1}B^T$ is likely to have small eigenvalues.

Some insight for the case of the *Stokes problem*, corresponding to $\nu = 1$ and $\mathbf{w} = \mathbf{0}$ in equation (9) below, can be gained as follows. For a stable discretization of a 2D problem we can write

$$A = \begin{pmatrix} L & O \\ O & L \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} B_1 & B_2 \end{pmatrix}$$

where L is a discrete (scalar) Laplacian, B_1 represents discretization of ∂_x, and B_2 represents discretization of ∂_y. If we assume that L, B_1 and B_2 are pairwise commuting matrices and that $L = B_1B_1^T + B_2B_2^T$ we have, for $M = I$:

$$B(A-\beta I)^{-1}B^T = B_1(L-\beta I)^{-1}B_1^T + B_2(L-\beta I)^{-1}B_2^T = L(L-\beta I)^{-1}.$$

Therefore, the eigenvalues of $BA_\beta^{-1}B^T = B(A-\beta I)^{-1}B^T$ are given by $\mu_i = \frac{\zeta_i}{\zeta_i - \beta}$ where the ζ_i's are the eigenvalues of the discrete negative Laplacian. It is easy to see that the smallest value (in magnitude) is achieved for $\zeta_i = \zeta_{\min}$, since by hypothesis $\beta > \zeta_{\min}$. For the Dirichlet Laplacian on the unit square, the smallest eigenvalue is given by $\zeta_{\min} \approx 2\pi^2 \approx 19.74$. This yields the expression

$$\lambda = \frac{\gamma}{\frac{2\pi^2-\beta}{2\pi^2} + \gamma}. \tag{8}$$

For instance, plugging $\beta = 300$ and $\gamma = 100$ in (8) yields the value $\lambda \approx 1.17$. Hence, taking $\gamma = 100$ results in the entire spectrum of the preconditioned matrix $\mathcal{A}_\gamma \mathcal{P}_\gamma^{-1}$ being clustered near 1.

The foregoing argument is of course not entirely rigorous, since in practice the discrete operators A, B_1 and B_2 do not commute except in very special situations; see the discussion in [9, Section 8.2]. Nevertheless, the results of our numerical experiments in the following section suggest that the argument is not completely without heuristic value.

4 Numerical experiments

In this section we present the results of numerical tests using the augmented Lagrangian-based block triangular preconditioner. The block matrices used for the experiments arise from discretizations of incompressible fluid flow problems using the Marker-and-Cell (MAC) scheme. Both exact and inexact preconditioner solves are considered, using GMRES [19] and FGMRES [18] as the respective accelerators. Similar results have been observed using different (LBB-stable) discretizations and Krylov subspace solvers.

The matrices arise from consideration of the following model problem. Let $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ be a bounded domain with a Lipschitz boundary $\partial\Omega$. We consider the steady *Oseen-type problem*

$$-\nu\Delta\mathbf{u} + \mathbf{w} \times \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in} \quad \Omega \tag{9}$$

$$\operatorname{div}\mathbf{u} = 0 \quad \text{in} \quad \Omega \tag{10}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on} \quad \partial\Omega \tag{11}$$

arising from Picard linearization of the *rotation form* of the steady-state Navier–Stokes equations. Here $\mathbf{u}$ represents the velocity field, p the Bernoulli pressure, $\nu > 0$ the kinematic viscosity coefficient, $\mathbf{f}$ an external force field, and $\mathbf{w}$ a known coefficient computed from the curl of the velocity field obtained from the previous Picard iteration; see, e.g., [4, 17] and the references therein. For $\mathbf{w} = \mathbf{0}$ we have as a special case the Stokes problem. In this case we can rescale p and $\mathbf{f}$ and let $\nu = 1$. For the Oseen problem this rescaling is not useful, and the viscosity parameter ν controls the difficulty of the problem: the smaller ν, the harder the problem.

Discretization of problem (9)-(11) by MAC or any LBB-stable finite element method leads to the solution of linear systems in saddle point form; see, e.g., [3, 5, 9, 11, 13, 16]. The diagonal blocks of the $n \times n$ matrix A consist of d discrete Laplace operators. These blocks are coupled through the components of the vector field $\mathbf{w}$; for the Stokes problem, $\mathbf{w} = \mathbf{0}$ and the diagonal blocks are uncoupled. The $m \times n$ matrix B represents a discrete divergence operator and its transpose, B^T, the discrete (negative) gradient. We note that usually B has rank $m - 1$ since the kernel of B^T contains the constant vectors. This makes the saddle point system singular; while it is possible to remove this singularity by including an additional condition on the pressure, in practice the (simple) eigenvalue $\lambda = 0$ does not cause any problem to the convergence of preconditioned Krylov methods and there is no need for additional conditions; see [9, Section 2.3].

Investigation of the stability of steady flows, on the other hand, leads to generalized eigenproblems of the form (2), where M is the velocity mass matrix;

see, e.g., [8, 13]. Solution of (2) by shift-and-invert methods, in turn, requires the repeated solution of linear systems of the form (1) for several different values of β and different right-hand sides. Here we experiment with linear systems of such type. Equations (9)-(11) are discretized with MAC on a uniform grid. We use the augmented Lagrangian formulation (3) together with the preconditioner $\mathcal{P}_\gamma$ given by (4). The use of MAC as the discretization scheme leads to $M = I_n$ in (1) for a suitable scaling of the discrete equations. Both symmetric (Stokes-type) and nonsymmetric (Oseen-type) problems are considered. In all cases we impose homogeneous Dirichlet boundary conditions on the velocity ($\mathbf{g} = \mathbf{0}$ in (11)). For the Oseen-type problem (in dimension $d = 2$) we use the coefficient $\mathbf{w} = (8x(x-1)(1-2y), 8(2x-1)y(y-1))$. Experiments with different wind functions and with 3D problems resulted in very similar conclusions to those reported here.

We first study the effectiveness of the preconditioner with "exact" solves: i.e., linear systems with coefficient matrix $A_\beta + \gamma B^T B$ are solved by a direct sparse LU factorization in combination with appropriate sparsity-preserving orderings. We also investigate the effect of inexact preconditioner solves, obtained by an inner preconditioned Krylov iteration carried out to some prescribed accuracy. As the Krylov method of choice we use GMRES for the exact case, and FGMRES for the inexact case. For the inner iterations we use again GMRES preconditioned by incomplete LU factorizations. For the initial guess we always start from the zero vector. The (outer) iteration is stopped when the 2-norm of the initial residual has been reduced by at least six orders of magnitude. All results were computed in MATLAB 7.1.0 on one processor of an AMD Opteron with 32 GB of memory.

4.1 *Exact solves*

In the first set of experiments with the block triangular preconditioner (4) we generate linear systems corresponding to MAC discretizations of the Stokes problem on grids of different sizes. Our aim is to assess the dependence of the preconditioner $\mathcal{P}_\gamma$ on the mesh size h, for different values of γ. The experiments are repeated for different values of β. In the case of the Stokes problem on the unit square $\Omega = [0,1] \times [0,1]$, the smallest eigenvalue of A is $\lambda_{\min}(A) \approx 2\pi^2 \approx 19.74$ and the largest one is $\lambda_{\max}(A) \approx 8h^{-2}$. Therefore, β must be taken between these two values for the $(1,1)$ block of (1) to be indefinite.

Iteration counts (in terms of matrix-vector products) for full GMRES are given in Tables 1-3. The first conclusion to be drawn from this tables is that for the Stokes-type problem, the rate of convergence of the preconditioned iteration is, for fixed γ and β, independent of the mesh size h. Furthermore, the rate of convergence rapidly improves as γ goes from small to large (for β fixed), and tends to slowly deteriorate as β increases (for γ large). These

Table 1. Iteration counts of preconditioned GMRES for the 2D Stokes problem, $\beta = 100$, exact solves.

grid	$\gamma = 100$	$\gamma = 10$	$\gamma = 2$	$\gamma = 1$	$\gamma = 0.2$	$\gamma = 0.1$
16×16	3	6	12	14	22	23
32×32	3	6	12	15	23	24
64×64	3	6	13	15	24	25
128×128	3	6	13	15	25	26

Table 2. Iteration counts of preconditioned GMRES for the 2D Stokes problem, $\beta = 300$, exact solves.

grid	$\gamma = 100$	$\gamma = 10$	$\gamma = 2$	$\gamma = 1$	$\gamma = 0.2$	$\gamma = 0.1$
16×16	4	12	25	32	51	55
32×32	4	10	23	31	48	52
64×64	4	11	23	33	49	53
128×128	4	11	23	33	50	53

Table 3. Iteration counts of preconditioned GMRES for the 2D Stokes problem, $\beta = 1000$, exact solves.

grid	$\gamma = 100$	$\gamma = 10$	$\gamma = 2$	$\gamma = 1$	$\gamma = 0.2$	$\gamma = 0.1$
16×16	8	26	69	100	182	199
32×32	6	21	59	88	142	154
64×64	6	23	60	84	138	149
128×128	6	24	60	82	135	141

results are not surprising in view of the theoretical analysis in section 3.

Next, we fix $\gamma = 100$ and vary the mesh size and the parameter β, using values between 20 and 300. The underlying problem is again a Stokes-type problem on the unit square. We note that the matrix $A_\beta = A - \beta I_n$ corresponding to the 32×32 grid (for which $n = 1984$) has two negative eigenvalues for $\beta = 20$, six for $\beta = 50$, twelve for $\beta = 100$, and thirty-eight for $\beta = 300$. For these test runs we also provide some CPU timings. Results are shown in Table 4 for meshes up to size 256×256. The total number of unknowns for the finest grid is $n + m = 196,096$. We report the number of GMRES iterations (under 'Its'), the time for the sparse LU factorization of $A_\beta + \gamma B^T B$ (under 'Time LU'), the time for the iterative solve phase (under 'Time its'), and the total time. Prior to factorization, an approximate minimum degree ordering [1] was applied: the time to compute the permutation is negligible compared to overall solution time. Because we are using a direct solver, the time scales superlinearly in the problem size; note that at each refinement of the mesh the number of unknowns grows by a factor of four.

In Table 5 we present a few results on a 3D Stokes-like problem on the unit cube $\Omega = [0, 1] \times [0, 1] \times [0, 1]$. Note that halving the mesh size now increases the total number of unknowns by a factor of eight.

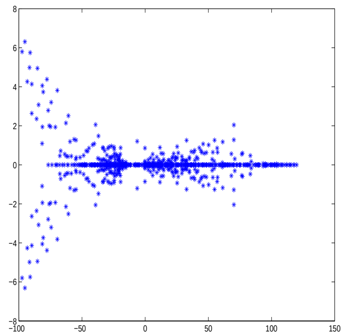
Table 4. Iterations and CPU times for preconditioned GMRES; 2D Stokes problem, $\gamma = 100$, exact solves.

grid	β	Its	Time LU	Time its	Total time
16×16	20	3	0.01	0.01	0.01
	50	3	0.01	0.01	0.01
	100	3	0.01	0.01	0.01
	300	4	0.01	0.01	0.02
32×32	20	3	0.05	0.01	0.06
	50	3	0.04	0.03	0.07
	100	3	0.04	0.04	0.08
	300	4	0.04	0.04	0.08
64×64	20	3	0.34	0.09	0.43
	50	3	0.33	0.09	0.42
	100	3	0.35	0.17	0.52
	300	4	0.33	0.25	0.58
128×128	20	3	2.68	0.47	3.15
	50	3	2.68	0.46	3.14
	100	3	2.67	0.89	3.56
	300	5	2.80	1.56	4.36
256×256	20	3	22.23	2.12	24.35
	50	3	22.59	2.12	24.71
	100	3	22.57	4.56	27.13
	300	5	22.66	7.75	30.41

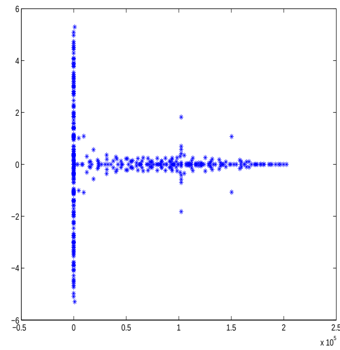
Table 5. Iterations and CPU times for preconditioned GMRES; 3D Stokes problem, $\gamma = 100$, exact solves.

grid	β	Its	Time LU	Time its	Total time
8×8	20	3	0.05	0.08	0.13
	50	4	0.03	0.21	0.24
	100	4	0.03	0.20	0.23
	300	7	0.12	0.32	0.44
16×16	20	3	0.59	1.06	1.65
	50	4	0.59	2.85	3.44
	100	4	0.60	2.79	3.39
	300	6	7.00	3.83	10.83
32×32	20	3	15.01	20.75	35.76
	50	4	15.88	45.12	61.00
	100	4	15.02	44.95	59.97
	300	6	16.78	59.91	76.69

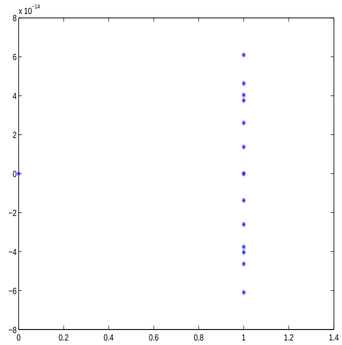
The next set of experiments is concerned with nonsymmetric (Oseen-type) problems on the unit square. Now we have an additional parameter, namely,



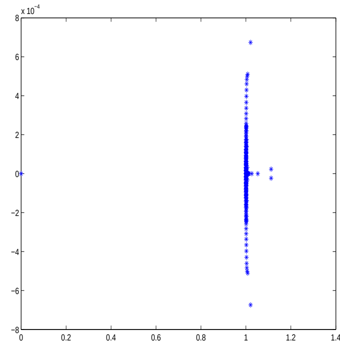
(a) Indefinite system.



(b) Augmented Lagrangian system.



(c) Case $\beta = 20$.



(d) Case $\beta = 100$.

Figure 1. Eigenvalue distribution for the 2D Oseen-type problem with $\nu = 10^{-2}$.

the viscosity ν. We found again that $\gamma = 100$ results in fast convergence of the preconditioned iteration, therefore we use this value for all the remaining experiments. Results for a sequence of grids (from 8×8 to 256×256) with values of ν ranging from 10^{-1} down to 10^{-6} show that the number of GMRES iterations is independent of both h and the viscosity ν. The rate of convergence is also independent of the parameter β, at least for values of β between 20 and 300. Similar results were obtained on 3D problems. Hence, the exact version of the preconditioner $\mathcal{P}_\gamma$ with $\gamma = 100$ is extremely robust with respect to all characteristic problem parameters. Figs. 1(a)-(b) display the eigenvalues of the systems (1) and (3), respectively, for the discrete Oseen-type operator on the unit square on a 16×16 grid, for $\nu = 10^{-2}$ and $\beta = 100$. Note the different scaling of the horizontal axis in the two figures. Fig. 1(c) displays the eigenvalues of the preconditioned matrix when $\beta = 20$, $\gamma = 100$ and Fig. 1(d)

Table 6. Iterations and CPU times for preconditioned GMRES; 2D Oseen problem, $\gamma = 100$, exact solves.

grid	ν	β	Its	Time LU	Time its	Total time
64×64	10^{-1}	20	3	1.12	0.47	1.59
	10^{-2}	20	3	1.14	0.73	1.87
	10^{-3}	20	3	1.43	0.90	2.33
64×64	10^{-1}	50	4	1.15	0.68	1.83
	10^{-2}	50	3	1.18	0.79	1.97
	10^{-3}	50	3	1.42	0.83	2.25
64×64	10^{-1}	100	4	1.16	0.79	1.95
	10^{-2}	100	4	1.29	1.08	2.37
	10^{-3}	100	4	1.38	1.06	2.44
64×64	10^{-1}	300	5	1.17	1.10	2.27
	10^{-2}	300	4	1.40	1.09	2.49
	10^{-3}	300	4	1.30	1.07	2.37
128×128	10^{-1}	20	3	15.36	3.41	18.77
	10^{-2}	20	3	15.49	5.15	20.64
	10^{-3}	20	3	16.48	6.66	23.14
128×128	10^{-1}	50	4	15.33	4.81	20.14
	10^{-2}	50	3	15.51	5.90	21.41
	10^{-3}	50	3	19.71	7.41	27.12
128×128	10^{-1}	100	4	15.34	5.55	20.89
	10^{-2}	100	4	15.57	7.69	23.26
	10^{-3}	100	4	20.36	9.17	29.53
128×128	10^{-1}	300	4	15.49	7.00	22.49
	10^{-2}	300	4	18.68	8.98	27.66
	10^{-3}	300	4	18.58	8.47	27.05
256×256	10^{-1}	20	3	244.35	27.20	271.55
	10^{-2}	20	3	243.87	36.64	280.51
	10^{-3}	20	3	246.02	50.11	296.13
256×256	10^{-1}	50	4	243.93	37.56	281.49
	10^{-2}	50	3	244.11	44.87	288.98
	10^{-3}	50	3	249.78	52.68	302.46
256×256	10^{-1}	100	4	244.15	41.31	285.46
	10^{-2}	100	4	244.62	60.13	304.75
	10^{-3}	100	4	267.94	71.58	339.52
256×256	10^{-1}	300	4	244.08	50.82	294.90
	10^{-2}	300	4	245.96	63.00	308.96
	10^{-3}	300	4	312.89	78.45	391.34

Table 7. Iteration count for preconditioned FGMRES, Stokes problem, $\gamma = 100$, inexact solves, $\tau = 10^{-5}$.

grid	8×8	16×16	32×32	64×64
β	outer/inner	outer/inner	outer/inner	outer/inner
20	4/4	5/5	5/16	15/109
50	5/6	6/10	8/22	19/152
100	4/4	5/7	6/26	15/128
300	4/4	8/10	8/19	16/183

displays the eigenvalues of the preconditioned system when $\beta = \gamma = 100$. Both spectra are tightly clustered around 1.

Computational results, including timings, are shown in Table 6 for the three finest grids and three values of ν. In these experiments, the (nonsymmetric) matrix $A_\beta + \gamma B^T B$ was reordered with a reverse Cuthill–McKee ordering, as the approximate minimum degree ordering occasionally resulted in very high fill-in in the LU factors. As it can be seen, the cost of the LU factorization dominates the total solution time.

4.2 *Inexact solves*

In practice, using exact solves in the application of the preconditioner may be too expensive, especially for three-dimensional problems. Here we consider replacing the exact solves with inexact ones, obtained via an inner preconditioned GMRES iteration. In this paper we explore the use of drop tolerance-based incomplete LU as the preconditioner for the inner iteration. Since the inner solves are based on a non-stationary method, we use flexible GMRES (FGMRES) for the outer iteration.

In the first set of experiments we consider the 2D Stokes-type problem. As before, we take $\gamma = 100$. The matrix $A_\beta + \gamma B^T B$ is reordered with the approximate minimum degree algorithm. We compute an incomplete LU factorization with a fixed value $\tau = 10^{-5}$ for the drop tolerance, and we stop the inner GMRES iteration when the corresponding (relative) residual norm has been reduced below $tol = 10^{-2}$, the residual norm tolerance for the outer FGMRES iteration being kept at 10^{-6}. The results are shown in Table 7 for different values of β and different grids. Under 'outer/inner' we report, respectively, the number of outer FGMRES iterations and the *total* number of inner GMRES iterations.

It is clear from these results that the quality of the inexact preconditioner deteriorates as the mesh is refined; only the behavior with respect to β is completely satisfactory. The problem stems from the fact that using a constant value of the drop tolerance does not work well, as the matrix entries actually grow unboundedly as $h \to 0$.

Table 8. Iteration count for preconditioned FGMRES, Stokes problem, $\gamma = 100$, inexact solves, $\tau = 10^{-p}$.

grid	8×8	16×16	32×32	64×64	128×128
β	outer/inner	outer/inner	outer/inner	outer/inner	outer/inner
20	8/9	10/12	8/15	13/18	11/17
50	12/22	10/16	10/15	12/17	14/28
100	9/11	11/16	11/23	12/22	13/30
300	11/19	14/19	9/10	9/10	11/12

In the next set of experiments we used an adaptive drop tolerance, namely, $\tau = 10^{-p}$ where $h = 2^{-p}$. Thus, for the 128×128 grid, we used $\tau = 10^{-7}$. We set the inner stopping criterion at $tol = 10^{-1}$; the results are shown in Table 8. Now the dependency of the convergence rate on h is rather mild and the average number of inner iterations is always less than three per outer iteration. The fact that the number of outer iterations is sometimes smaller than in the previous set of experiments (see the results for the 64×64 grid) can be explained by observing that with the adaptive drop tolerance now used, the actual inner residual is sometimes rather small after the last inner iteration. Unfortunately, the cost of the inexact preconditioner still scales superlinearly, due to the need to compute the incomplete factorization with smaller and smaller drop tolerances. As is well known, this is an inherent limitation of incomplete factorizations in PDE-type problems; better scalings may be possible if multilevel methods are used for the approximate inner solves.

Finally, we performed some experiments with inexact solves for Oseen-type problems for different values of the viscosity ν. In this case we found that reordering the matrix $A_\beta + \gamma B^T B$ can cause the incomplete factorization process to break down; therefore, no reordering was applied. Again, we found that using an adaptive drop tolerance is necessary in order to keep the number of iterations stable with respect to h and ν. Some results are shown in Table 9.

Although the total number of inner iterations is now quite reasonable, the cost of the incomplete LU factorization with adaptively chosen drop tolerance scales superlinearly, and the total computing times are generally no better than those obtained with the complete LU factorization. It should be kept in mind that the incomplete factorizations and inner iterations used are not very efficient. For the complete factorizations, the Matlab code makes use of highly optimized sparse direct solvers. In contrast, the implementations of the incomplete LU factorization function in Matlab is not very efficient. Because of this, the incomplete factors may actually be more expensive to compute than the complete factors, and the additional costs induced by the (few) inner Krylov subspace iterations needed to satisfy the convergence criteria for the inexact solves lead to increased overall solution costs compared to the case of exact solves. Clearly, better inner solution strategies need to be developed.

Table 9. Iteration count for FGMRES, Oseen-type problem, $\gamma = 100$, inexact solves, $\tau = 10^{-p-1}$.

	grid	8×8	16×16	32×32	64×64
ν	β	outer/inner	outer/inner	outer/inner	outer/inner
10^{-1}	20	11/14	12/21	10/15	8/11
	50	7/8	6/6	7/8	6/7
	100	5/5	9/9	6/6	6/6
	300	6/6	6/6	8/9	7/7
10^{-2}	20	5/5	7/7	9/9	8/8
	50	5/5	4/4	8/9	9/10
	100	5/5	4/4	5/5	6/6
	300	5/5	5/5	5/5	6/6
10^{-3}	20	5/5	4/4	5/5	18/26
	50	4/4	4/4	4/4	4/4
	100	4/4	5/5	4/4	4/4
	300	5/5	5/5	5/5	5/5

5 Conclusions

In this paper we have investigated the use of an augmented Lagrangian-based block triangular preconditioner for certain saddle point systems with an indefinite $(1, 1)$ block. Our numerical experiments on block matrices arising from incompressible fluid dynamics problems indicate that the preconditioner results in fast convergence independently of mesh size and viscosity. For large problems, the efficient implementation of the proposed method demands the use of inexact preconditioner solves. We have shown experimentally that fast convergence of the outer preconditioned flexible iteration is often observed even when the preconditioner solves are performed to a prescribed accuracy using an inner preconditioned Krylov iteration. However, we have found that for some of the more difficult problems even reaching a modest level of accuracy in the preconditioner solves may require considerable computational effort. Therefore, how to best perform the inexact inner solves for such problems remains an open question.

Acknowledgements. The first author was supported in part by National Science Foundation grant DMS-0511336. The authors would like to thank Prof. Ke Chen for his invitation to contribute the present paper.

References

[1] Amestoy, P. R., Davis, T. A. and Duff, I. S., 1996, An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, **17**, 886–905.

[2] Arbenz, P. and Geus, R., 2005, Multilevel preconditioned iterative eigensolvers for Maxwell eigenvalue problems. *Applied Numerical Mathematics*, **54**, 107–121.

[3] Benzi, M., Golub, G. H. and Liesen, J., 2005, Numerical solution of saddle point problems. *Acta Numerica*, **14**, 1–137.
[4] Benzi, M. and Liu, J., 2007, An efficient solver for the Navier–Stokes equations in rotation form. *SIAM Journal on Scientific Computing*, to appear.
[5] Benzi, M. and Olshanskii, M. A., 2006, An augmented Lagrangian-based approach to the Oseen problem. *SIAM Journal on Scientific Computing*, **28**, 2095–2113.
[6] Benzi, M. and Simoncini, V., 2006, On the eigenvalues of a class of saddle point matrices. *Numerische Mathematik*, **103**, 173–196.
[7] Chen, K., 2005, *Matrix Preconditioning Techniques and Applications* (Cambridge: Cambridge University Press).
[8] Cliffe, K. A., Garratt, T. J. and Spence, A., 1994, Eigenvalues of block matrices arising from problems in fluid mechanics. *SIAM Journal on Matrix Analysis and Applications*, **15**, 1310–1318.
[9] Elman, H., Silvester, D. and Wathen, A., 2005, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics* (Oxford: Oxford University Press).
[10] Fletcher, R., 1987, *Practical Methods of Optimization*, 2nd ed., (Chichester: Wiley & Sons).
[11] Fortin, M. and Glowinski, R., 1983, *Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problems* (Amsterdam, New York, Oxford: North-Holland).
[12] Gantner, A., Hoppe, R. H. W., Köster, D., Siebert, K. and Wixforth, A., 2007, Numerical simulation of piezoelectrically agitated surface acoustic waves on microfluidic biochips. *Computing and Visualization in Science*, to appear.
[13] Glowinski, R., 2003, Finite element methods for incompressible viscous flow. *Handbook of Numerical Analysis. Volume IX, Numerical Methods for Fluids (Part 3)*, (Amsterdam: North-Holland).
[14] Golub, G. H. and Greif, C., 2003, On solving block structured indefinite linear systems. *SIAM Journal on Scientific Computing*, **24**, 1076–2092.
[15] Greif, C. and Schötzau, D., 2007, Preconditioners for the discretized time-harmonic Maxwell equations in mixed form. *Numerical Linear Algebra with Applications*, to appear.
[16] Lube, G. and Olshanskii, M. A., 2002, Stable finite element calculation of incompressible flows using the rotation form of convection. *IMA Journal of Numerical Analysis*, **22**, 437–461.
[17] Olshanskii, M. A., 1999, An iterative solver for the Oseen problem and numerical solution of incompressible Navier–Stokes equations. *Numerical Linear Algebra with Applications*, **6**, 353–378.
[18] Saad, Y., 1993, A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, **14**, 461–469.
[19] Saad, Y. and Schultz, M. H., 1986, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, **7**, 856–869.