Journal of Computational and Applied Mathematics 235 (2010) 263–269

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

New choices of preconditioning matrices for generalized inexact parameterized iterative methods☆

Yang Cao [a], Mei-Qun Jiang [a,*], Lin-Quan Yao [a,b]

[a] School of Mathematical Science, Soochow University, Suzhou, Jiangsu, 215006, PR China
[b] School of Urban Transportation, Soochow University, Suzhou, Jiangsu, 215021, PR China

ARTICLE INFO

Article history:
Received 28 December 2009
Received in revised form 30 May 2010

MSC:
65H10
65W05

Keywords:
Saddle point problems
Iterative method
Preconditioning

ABSTRACT

For large sparse saddle point problems, Chen and Jiang recently studied a class of generalized inexact parameterized iterative methods (see [F. Chen, Y.-L. Jiang, A generalization of the inexact parameterized Uzawa methods for saddle point problems, Appl. Math. Comput. 206 (2008) 765–771]). In this paper, the methods are modified and some choices of preconditioning matrices are given. These preconditioning matrices have advantages in solving large sparse linear system. Numerical experiments of a model Stokes problem are presented.

1. Introduction

We consider the solution of systems of linear equations of the block 2×2 form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad \text{or} \quad \mathcal{A}u = \hat{b}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$, $x, f \in \mathbb{R}^n$, $y, g \in \mathbb{R}^m$, and $m \leq n$. We assume that the matrix A is symmetric positive definite and B has full rank. Under those conditions we know that the solution of (1.1) exists and is unique. The system (1.1) is called symmetric saddle point problem. We further assume that the matrices A, B, and C are large and sparse.

The system (1.1) arises in a variety of scientific and engineering applications, including computational fluid dynamics, mixed finite element methods for solving elliptic PDEs, constrained optimization, constrained least-squares problems and so on. Therefore, it is of great importance to develop efficient iterative methods for such problems.

When the matrix blocks $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times n}$ are large and sparse, iterative methods become more attractive than direct methods for solving the saddle point problems (1.1), but direct methods play an important role in the form of preconditioners embedded in an iterative framework. The best known and the oldest methods are the Uzawa and preconditioned Uzawa methods. In recent years, a large variety of methods for solving (1.1) have been studied. For example, the inexact Uzawa algorithms [1–7], the nonlinear Uzawa algorithms [8], the HSS algorithms [9] and so on. It should be mentioned that the HSS method is very useful for non-Hermitian positive definite systems of equations. Algorithmic variants and theoretical analyses of these HSS iteration methods for the saddle point problems have been extensively and deeply discussed in [10–15]

☆ This work is supported by The National Natural Science Foundation (No. 10672111), PR China.
* Corresponding author.
E-mail addresses: caoyangsqq@163.com (Y. Cao), mqjiang@suda.edu.cn (M.-Q. Jiang), lqyao@suda.edu.cn (L.-Q. Yao).

0377-0427/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2010.05.054

and so on. A large variety of other methods for solving linear systems of the form (1.1) can be found in the excellent survey paper [16] and references therein. In this paper, the important class of iterative methods that we focus on is the class of inexact parameterized Uzawa methods. These methods have been studied in many papers, such as [1,2,5,7] and so on. Recently, Chen and Jiang studied a class of generalized parameterized inexact Uzawa (GPIU) methods in [5]. The GPIU method outperforms the parameterized inexact iterative method [1]. The convergence results under suitable restrictions on the preconditioning matrices are given. In this paper, the GPIU method is modified and called the MGPIU method. Besides, some choices of preconditioning matrices are given. These preconditioning matrices have advantages in solving large sparse linear systems.

The remainder of this paper is arranged as follows. In Section 2, a class of generalized parameterized inexact Uzawa methods are modified for solving saddle point problems and the conditions for guaranteeing its convergence are derived. In Section 3, Numerical experiments of a model Stokes problem are presented. The numerical results show that the modified methods given in this paper have a better convergence rate than the generalized inexact parameterized iterative methods studied in [5]. Finally, we give some conclusions in Section 4.

2. Iterative method

In [5], the authors first rewrite (1.1) as its equivalent form

$$\begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ -g \end{bmatrix}, \quad \text{or} \quad \mathcal{A}u = b. \tag{2.1}$$

The coefficient matrix $\mathcal{A}$ in (2.1) has the following desirable properties.

Lemma 2.1 (*[14]*). *Let $\mathcal{A} \in \mathbb{C}^{(m+n)\times(m+n)}$ be the coefficient matrix defined in (2.1), where A is symmetric and positive definite and B has full row-rank. Let $\sigma(\mathcal{A})$ denote the spectrum of $\mathcal{A}$ and $\lambda \in \sigma(\mathcal{A})$ be an eigenvalue of $\mathcal{A}$. Then*

1. *$\mathcal{A}$ is nonsingular;*
2. *$\mathcal{A}$ is positive real, i.e., $\xi^* \mathcal{A}\xi > 0$ for all $\xi \in \mathbb{C}^{(m+n)}$ ($\xi \neq 0$);*
3. *$\mathcal{A}$ is positive stable, i.e., $\mathrm{Re}(\lambda) > 0$ holds for all $\lambda \in \sigma(\mathcal{A})$, where $\mathrm{Re}(\lambda)$ denotes the real part of the complex number λ.*

Thus by changing the sign of the last m equations in (1.1), we can gain the positive definiteness. The positive definiteness of the coefficient matrix is very important for many simple iterative solvers [14,7,15]. Then the following splitting is considered in [5]

$$\begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} = \begin{bmatrix} A+Q_1 & 0 \\ -B+Q_3 & Q_2 \end{bmatrix} - \begin{bmatrix} Q_1 & -B^T \\ Q_3 & Q_2 \end{bmatrix},$$

where $Q_1 \in \mathbb{R}^{n\times n}$ is symmetric positive semi-definite, $Q_2 \in \mathbb{R}^{m\times m}$ is symmetric positive definite, and $Q_3 \in \mathbb{R}^{m\times n}$ is arbitrary. The corresponding iterative method is described below.

Algorithm 2.1.

$$\begin{cases} x_{n+1} = x_n + (A+Q_1)^{-1}(f - Ax_n - B^T y_n), \\ y_{n+1} = y_n + Q_2^{-1}((B-Q_3)x_{n+1} + Q_3 x_n - g). \end{cases}$$

It should be mentioned that Algorithm 2.1 presents a general framework for the PIU methods for solving the saddle-point problems. Choosing different preconditioning matrices, we obtain different algorithms. If $Q_1 = \frac{1}{\delta}A - A$, $Q_2 = \frac{1}{\tau}Q$, $Q_3 = 0$ ($\delta > 0$) and Q is symmetric positive definite, then this gives the following well-known generalized successive overrelaxation (GSOR) method [1].

Algorithm 2.2. GSOR method

$$\begin{cases} x_{n+1} = (1-\delta)x_n + \delta A^{-1}(f - B^T y_n), \\ y_{n+1} = y_n + \tau Q^{-1}(Bx_{n+1} - g). \end{cases}$$

In particular, when $\delta = \tau$, the GSOR method becomes the SOR-like method [7]. For other algorithms with different choices of Q_1, Q_2 and Q_3, one can see [5, Algorithms 3.1–3.5].

The convergence analysis of Algorithm 2.1 is summarized as follows.

Theorem 2.1 (*[5]*). *Assume that A is symmetric positive definite and B has full row rank. If Q_1 is symmetric positive semidefinite, Q_2 is symmetric positive definite, and $Q_3 \in \mathbb{R}^{m\times n}$ is such that $B^T Q_2^{-1} Q_3$ is symmetric, then the generalized iterative method is convergent provided that*

$$\gamma - 4\alpha - 2\beta < 2\tau < 2\beta,$$

where

$$\alpha = \frac{u^*Q_1u}{u^*u} \geq 0, \qquad \beta = \frac{u^*Au}{u^*u} > 0, \qquad \gamma = \frac{u^*B^TQ_2^{-1}Bu}{u^*u} \geq 0, \qquad \tau = \frac{u^*B^TQ_2^{-1}Q_3u}{u^*u}$$

and $(u^*, v^*)^*$ *is an eigenvector of the iteration matrix with* $u \in \mathbb{C}^n$ *and* $v \in \mathbb{C}^m$.

In [5], the authors choose $Q_3 = tB$ (t is a parameter) such that $B^TQ_2^{-1}Q_3$ is symmetric. Then, the generalized iterative method becomes

Algorithm 2.3. GPIU method

$$\begin{cases} x_{n+1} = x_n + (A + Q_1)^{-1}(f - Ax_n - B^Ty_n), \\ y_{n+1} = y_n + Q_2^{-1}((1-t)Bx_{n+1} + tBx_n - g). \end{cases}$$

In fact, we can choose $Q_3 = -Q_2B$ such that $B^TQ_2^{-1}Q_3$ is symmetric. Then, the new iterative scheme is given.

Algorithm 2.4. MGPIU method

$$\begin{cases} x_{n+1} = x_n + (A + Q_1)^{-1}(f - Ax_n - B^Ty_n), \\ y_{n+1} = y_n + B(x_{n+1} - x_n) + Q_2^{-1}(Bx_{n+1} - g). \end{cases}$$

In the following, we deduce the convergence property for the MGPIU iteration. Note that the iteration matrix of the MGPIU iteration is

$$\Gamma = \begin{bmatrix} A + Q_1 & 0 \\ -(Q_2 + I)B & Q_2 \end{bmatrix}^{-1} \begin{bmatrix} Q_1 & -B^T \\ -Q_2B & Q_2 \end{bmatrix}.$$

Let $\rho(\Gamma)$ denote the spectral radius of Γ. Then the MGPIU iteration converges if and only if $\rho(\Gamma) < 1$ [17]. Let λ be an eigenvalue of Γ and $\begin{pmatrix} u \\ v \end{pmatrix}$ be the corresponding eigenvector. Then we have

$$\begin{bmatrix} Q_1 & -B^T \\ -Q_2B & Q_2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} A + Q_1 & 0 \\ -(Q_2 + I)B & Q_2 \end{bmatrix} \begin{bmatrix} \lambda u \\ \lambda v \end{bmatrix},$$

or equivalently,

$$Q_1u - B^Tv = \lambda(A + Q_1)u, \tag{2.2}$$

$$-Q_2Bu + Q_2v = -\lambda(Q_2 + I)Bu + \lambda Q_2v. \tag{2.3}$$

To get a convergence condition, we first give some lemmas.

Lemma 2.2 (*[5]*)**.** *If* λ *is an eigenvalue of the matrix* Γ*, then* $\lambda \neq 1$.

Lemma 2.3 (*[5]*)**.** *If* $\begin{pmatrix} u \\ v \end{pmatrix}$ *is an eigenvector of the matrix* Γ *corresponding to the eigenvalue* λ*, then* $u \neq 0$*. Moreover, if* $v = 0$*, then* $0 \leq \lambda < 1$.

Theorem 2.2. *Assume that A is symmetric positive definite and B has full row rank. Let* Q_1 *be symmetric positive semidefinite and* Q_2 *be symmetric positive definite, then the MGPIU method is convergent provided that*

$$\gamma < 4\alpha + 2\beta + 2\tau,$$

where

$$\alpha = \frac{u^*Q_1u}{u^*u} \geq 0, \qquad \beta = \frac{u^*Au}{u^*u} > 0, \qquad \gamma = \frac{u^*B^TQ_2^{-1}Bu}{u^*u} \geq 0, \qquad \tau = -\frac{u^*B^TBu}{u^*u} \leq 0.$$

Proof. By Lemma 2.2 we have $\lambda \neq 1$. As Q_2 is symmetric positive definite, $(1 - \lambda)Q_2$ is nonsingular. Hence, from (2.3) we obtain

$$v = Bu - \frac{\lambda}{1 - \lambda}Q_2^{-1}Bu.$$

By eliminating v from (2.2), we have

$$(1 - \lambda)Q_1u - \lambda Au - B^TBu + \frac{\lambda}{1 - \lambda}B^TQ_2^{-1}Bu = 0. \tag{2.4}$$

If $Bu = 0$, then from (2.3) we have $v = 0$. By the second part of Lemma 2.3 we know $0 \le \lambda < 1$. If $Bu \ne 0$, we obtain $\tau = -\frac{u^*B^TBu}{u^*u} < 0$. In this case, (2.4) can be rewritten as

$$\lambda^2(\alpha+\beta) + \lambda(\gamma - 2\alpha - \beta - \tau) + \alpha + \tau = 0.$$

From [1,2], we know that a sufficient and necessary condition for the roots of the real quadratic equation

$$x^2 + bx + c = 0$$

to satisfy $|x| < 1$ is that

$$|c| < 1 \quad \text{and} \quad |b| < 1 + c.$$

Therefore, it is easy to check that the MGPIU method is convergent provided that

$$\left|\frac{\alpha+\tau}{\alpha+\beta}\right| < 1 \quad \text{and} \quad \left|\frac{\gamma - 2\alpha - \beta - \tau}{\alpha+\beta}\right| < 1 + \frac{\alpha+\tau}{\alpha+\beta}. \tag{2.5}$$

By straightforwardly solving (2.5), we get

$$\gamma - 4\alpha - 2\beta < 2\tau < 2\beta, \tag{2.6}$$

where

$$\alpha = \frac{u^*Q_1u}{u^*u} \ge 0, \qquad \beta = \frac{u^*Au}{u^*u} > 0, \qquad \gamma = \frac{u^*B^TQ_2^{-1}Bu}{u^*u} \ge 0, \qquad \tau = -\frac{u^*B^TBu}{u^*u} \le 0.$$

The right-hand of the inequality (2.6) unconditionally holds. The left-hand of the inequality (2.6) can be written as

$$\gamma < 4\alpha + 2\beta + 2\tau.$$

Thus we complete the proof. □

Let $\lambda_{\max}(W)$ and $\lambda_{\min}(W)$ denote the maximum and the minimum eigenvalues of an symmetric matrix W, respectively. Under suitable restrictions on the preconditioning matrix, the convergence condition of the MGPIU method can be simplified below.

Corollary 2.1. *Assume that the preconditioning matrix Q_1 satisfies $\lambda_{\min}(Q_1) > \frac{1}{2}\sigma_1$, where σ_1 is the largest singular value of B. Then by the definition of α, β and τ, the new iterative method converges provided that*

$$2\lambda_{\max}(B^TB) + \lambda_{\max}(B^TQ_2^{-1}B) < 4\lambda_{\min}(Q_1) + 2\lambda_{\min}(A).$$

3. Numerical experiments

In this section, we use the GSOR method, the GPIU method and the MGPIU method to solve a model Stokes problem. To discretize the Stokes problem, we take two methods. One is the upwind scheme [13,1]. The second is the "marker and cell" (M.A.C.) finite difference scheme [18]. In our computations, we take $k_{\max} = 1000$. All runs with respect to each iteration scheme are started from the initial vector $(x_0^T, y_0^T)^T = 0$, and terminated if the current iterations satisfy ERR = $\|r_k\|_2/\|r_0\|_2 \le 10^{-5}$ (where r_k is the residual at the kth iteration) or if the numbers of the prescribed iteration $k_{\max}$ are exceeded. We use IT and CPU to represent the number of iteration steps and the elapsed CPU time in seconds. All runs are performed in MATLAB 6.5 on an Intel Pentium 4 (512 M RAM) Windows XP system.

Consider Stokes problem: find u and p such that

$$\begin{cases} -\nu\Delta\mathbf{u} + \nabla p = \tilde{f}, & \text{in } \Omega \\ \nabla\cdot\mathbf{u} = \tilde{g}, & \text{in } \Omega \\ \mathbf{u} = 0, & \text{on } \partial\Omega \\ \displaystyle\int_\Omega p(x)\mathrm{d}x = 0 \end{cases} \tag{3.1}$$

where $\Omega = (0,1)\times(0,1) \subset \mathbb{R}^2$, $\partial\Omega$ is the boundary of Ω, ν stands for the viscosity scalar, Δ is the componentwise Laplace operator, $\mathbf{u} = (u^T, v^T)^T$ is a vector-valued function representing the velocity, and p is a scalar function representing the pressure. By discretizing (3.1) with the upwind scheme and the M.A.C. finite difference scheme, we obtain the system of linear equations

$$\begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix}\begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} f \\ -g \end{bmatrix} \tag{3.2}$$

with different structure of submatrices in the coefficient matrix. We consider the two examples in the following.

Table 1
Choices of the matrices Q, Q_1 and Q_2.

Case no.	Matrix Q	Matrix Q_1	Matrix Q_2	Descripition
I	$BA^{-1}B^T$	δA	$BQ_1^{-1}B^T$	
II	$B\hat{A}^{-1}B^T$	$\delta\hat{A}$	$BQ_1^{-1}B^T$	$\hat{A} = \mathrm{diag}(A)$
III	$B\hat{A}^{-1}B^T$	$\delta\hat{A}$	$BQ_1^{-1}B^T$	$\hat{A} = \mathrm{tridiag}(A)$

Table 2
Results of Case I for Example 3.1.

	$q = 16$				$q = 24$			
		IT	CPU	ERR		IT	CPU	ERR
GSOR	$\delta = 1$ $\tau = 1$	2	0.031	4.0384	$\delta = 1$ $\tau = 1$	2	0.187	5.7294
GPIU	$\delta = 0.3$	22	0.390	9.5700	$\delta = 0.3$	22	2.094	8.9994
	$\delta = 0.4$	18	0.313	8.2943	$\delta = 0.4$	18	1.687	8.1071
MGPIU	$\delta = 0.3$	23	0.406	6.7981	$\delta = 0.3$	22	2.172	9.9898
	$\delta = 0.4$	18	0.312	6.6677	$\delta = 0.4$	18	1.735	6.9395
	$\delta = 0.5$	20	0.359	5.2022	$\delta = 0.5$	20	1.9690	5.4769

Table 3
Results of Case II for Example 3.1.

	$q = 16$				$q = 24$			
		IT	CPU	ERR		IT	CPU	ERR
GSOR	$\delta = 0.3419$ $\tau = 0.2066$	74	1.313	8.3469	$\delta = 0.2488$ $\tau = 0.1423$	96	9.927	8.2548
GPIU	$\delta = 0.1$	100	1.078	7.0302	$\delta = 0.1$	176	10.266	8.6183
	$\delta = 0.2$	185	2.109	6.5916	$\delta = 0.2$	258	15.031	9.3994
MGPIU	$\delta = 0.1$	86	0.922	6.8367	$\delta = 0.05$	93	5.531	6.4722
	$\delta = 0.15$	119	1.359	7.1338	$\delta = 0.1$	153	9.078	8.1611
	$\delta = 0.2$	149	1.641	7.3246	$\delta = 0.15$	213	12.671	7.0390

Example 3.1. By discretizing (3.1) with the upwind scheme, the submatrices in the coefficient matrix (3.2) have the following form

$$A = \begin{bmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{bmatrix} \in \mathbb{R}^{2q^2 \times 2q^2},$$

and

$$B^T = \begin{bmatrix} I \otimes F \\ F \otimes I \end{bmatrix} \in \mathbb{R}^{2q^2 \times q^2},$$

where

$$T = \frac{\nu}{h^2} \cdot \mathrm{tridiag}(-1, 2, -1) \in \mathbb{R}^{q\times q}, \qquad F = \frac{1}{h} \cdot \mathrm{tridiag}(-1, 1, 0) \in \mathbb{R}^{q\times q}$$

with $\otimes$ being the Kronecker product symbol and $h = \frac{1}{q+1}$ the discretization meshsize.

For this example, we have $n = 2q^2$ and $m = q^2$. Hence, the total number of variables is $m + n = 3q^2$. In computation, we choose the matrices Q, Q_1 and Q_2 according to the three cases listed in Table 1. The parameters in the GSOR method are taken from [1, Table 2]. Parameter t in the GPIU method is chosen as 0.02 [5]. The corresponding numerical results of each method in different cases are listed in Tables 2–4.

The numerical results in Tables 2–4 show that the MGPIU method has better convergence property than the GPIU method and is slightly less efficient than the GSOR method. Choosing appropriate parameter, the MGPIU method may be better than the GSOR method. From Tables 2–4, we can also see that using the tridiagonal matrix as preconditioning matrix is better than the diagonal matrix. It should be pointed out that Case I in Table 2 is not very useful in practice, because it needs to solve the inverse of $BA^{-1}B^T$. In fact, once the inverse of $BA^{-1}B^T$ can be solved exactly, the direct method can be applied to solve the original system.

Example 3.2. By discretizing (3.1) with the M.A.C. difference scheme, the coefficient matrix of (3.2) contains three block rows, the first two of which come from the momentum equations for the individual components of the discrete velocity

Table 4
Results of Case III for Example 3.1.

	$q = 16$				$q = 24$			
		IT	CPU	ERR		IT	CPU	ERR
GSOR	$\delta = 0.4429$ $\tau = 0.2854$	52	0.891	9.2195	$\delta = 0.3308$ $\tau = 0.1985$	90	8.734	9.6850
GPIU	$\delta = 0.15$	81	0.891	5.0512	$\delta = 0.15$	142	8.328	6.3785
	$\delta = 0.2$	102	1.078	5.9508	$\delta = 0.2$	191	11.265	6.2958
MGPIU	$\delta = 0.1$	77	0.844	9.9826	$\delta = 0.1$	89	5.406	6.9972
	$\delta = 0.15$	74	0.828	5.8852	$\delta = 0.15$	122	7.406	6.5512
	$\delta = 0.2$	92	0.988	7.6556	$\delta = 0.2$	152	9.234	8.8822

Table 5
Results of Case I for Example 3.2.

	$q = 16$				$q = 24$			
		IT	CPU	ERR		IT	CPU	ERR
GSOR	$\delta = 1$ $\tau = 1$	2	0.031	2.3866	$\delta = 1$ $\tau = 1$	2	0.172	3.4079
GPIU	$\delta = 0.8$	28	0.438	7.4230	$\delta = 0.8$	28	2.574	6.7762
	$\delta = 0.9$	31	0.453	6.6486	$\delta = 0.9$	31	2.782	6.0750
MGPIU	$\delta = 0.7$	26	0.406	8.4598	$\delta = 0.7$	26	2.344	9.2703
	$\delta = 0.8$	25	0.306	2.0358	$\delta = 0.8$	25	2.235	2.2170
	$\delta = 0.9$	24	0.359	1.7658	$\delta = 0.9$	24	2.156	1.6248

Table 6
Results of Case II for Example 3.2.

	$q = 16$				$q = 24$			
		IT	CPU	ERR		IT	CPU	ERR
GSOR	$\delta = 0.3427$ $\tau = 0.2071$	50	0.734	8.7297	$\delta = 0.2442$ $\tau = 0.1392$	125	11.156	9.8717
GPIU	$\delta = 0.3$	288	2.766	9.3569	$\delta = 0.3$	644	35.609	8.3031
	$\delta = 0.4$	388	3.734	8.3867	$\delta = 0.4$	963	54.359	8.4441
MGPIU	$\delta = 0.2$	116	1.140	7.1398	$\delta = 0.1$	176	12.125	9.9408
	$\delta = 0.3$	161	1.532	9.2489	$\delta = 0.3$	309	17.438	8.1422
	$\delta = 0.4$	209	1.984	7.4705	$\delta = 0.4$	396	22.251	8.5670

field and the last from the incompressibility constraint. A and B have the form

$$A = \nu \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \qquad B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}.$$

The submatrices are defined as follows. Assume that Ω is divided into a uniform $q \times q$ grid of cells of width $h = 1/q$. Let ϕ_{jk} denote the value of a mesh function ϕ at the point $(jh, kh) \in \Omega$. The form of the indices (j, k) depends on the mesh function to which they correspond. In particular, they need not be integers. The first block row of (3.2) is defined by

$$[-\Delta u]_{jk} \approx [A_1 u]_{jk} \equiv \frac{1}{h^2}(4u_{jk} - u_{j-1,k} - u_{j+1,k} - u_{j,k+1} - u_{j,k-1}),$$

and

$$[p_x]_{jk} \approx [B_1^T p]_{jk} \equiv \frac{1}{h}(p_{j+1/2,k} - p_{j-1/2,k}).$$

The second block row (associated with v) is defined analogously. The discrete incompressibility constraint is

$$[u_x + v_y]_{jk} \approx [-Bu]_{jk} \equiv \frac{1}{h}(u_{j+1/2,k} - u_{j-1/2,k}) + \frac{1}{h}(v_{j+1/2,k} - v_{j-1/2,k}).$$

For this example, we have $n = 2(q-1)q$ and $m = q^2$. Hence, the total number of variables is $m + n = 3q^2 - 2q$. In computation, we also choose the matrices Q, Q_1 and Q_2 according to the three cases listed in Table 1. The parameters in the GSOR method are taken as the optimal iteration parameters [1, Theorem 4.1]. Parameter t in the GPIU method is also chosen as 0.02 [5]. The corresponding numerical results of each method with different cases are listed in Tables 5–7.

Clearly, the above numerical results show that the GSOR method has a faster convergence rate than the MGPIU method and the GPIU method. The MGPIU method is also more efficient than the GPIU method. Using the tridiagonal matrix as

Table 7
Results of Case III for Example 3.2.

	$q = 16$				$q = 24$			
		IT	CPU	ERR		IT	CPU	ERR
GSOR	$\delta = 0.4437$ $\tau = 0.2856$	36	0.563	8.2055	$\delta = 0.3246$ $\tau = 0.1939$	77	6.922	9.2712
GPIU	$\delta = 0.3$	163	1.563	9.6676	$\delta = 0.3$	343	19.172	9.3652
	$\delta = 0.4$	214	2.047	8.9275	$\delta = 0.4$	467	26.062	9.6939
MGPIU	$\delta = 0.3$	97	0.922	9.6724	$\delta = 0.2$	143	7.984	9.9260
	$\delta = 0.4$	121	1.72	8.1661	$\delta = 0.3$	182	10.125	7.8671
	$\delta = 0.5$	148	1.453	8.6813	$\delta = 0.4$	233	13.063	9.6435

preconditioning matrix is also better than the diagonal matrix. Moreover, the results show that the methods presented in this paper are powerful solvers for the large sparse saddle point problems.

4. Conclusion

In this paper, a new iterative (MGPIU) method is studied and some new choices of preconditioning matrices are presented for solving saddle point problems. In fact, the new method belongs to a class of parameterized inexact Uzawa (PIU) methods, which have been studied in many papers, see [1,2,5] and references therein. The convergence results under suitable restrictions on the preconditioning matrices are discussed. Numerical experiments of a model Stokes problem are presented to show the effectiveness of the MGPIU method.

References

[1] Z.-Z. Bai, B.N. Parlett, Z.-Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, Numer. Math. 102 (2005) 1–38.
[2] Z.-Z. Bai, Z.-Q. Wang, On parameterized inexact Uzawa methods for generalized saddle point problems, Linear Algebra Appl. 428 (2008) 2900–2932.
[3] J.H. Bramble, J.E. Pasciak, A.T. Vassilev, Analysis of the inexact Uzawa algorithm for saddle point problems, SIAM J. Numer. Anal. 34 (1997) 1072–1092.
[4] J.H. Bramble, J.E. Pasciak, A.T. Vassilev, Uzawa type algorithm for nonsymmetric saddle point problems, Math. Comp. 69 (1999) 667–689.
[5] F. Chen, Y.-L. Jiang, A generalization of the inexact parameterized Uzawa methods for saddle point problems, Appl. Math. Comput. 206 (2008) 765–771.
[6] H. Elman, G.H. Golub, Inexact and preconditioned Uzawa algorithms for saddle point problems, SIAM J. Numer. Anal. 31 (1994) 1645–1661.
[7] G.H. Golub, X. Wu, J.-Y. Yuan, SOR-like methods for augmented systems, BIT 41 (2001) 71–85.
[8] Z.-H. Cao, Fast Uzawa algorithm for generalized saddle point problems, Appl. Numer. Math. 46 (2003) 157–171.
[9] Z.-Z. Bai, G.H. Golub, M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl. 24 (2003) 603–626.
[10] Z.-Z. Bai, Optimal parameters in the HSS-like methods for saddle point problems, Numer. Linear Algebra Appl. 16 (2009) 447–479.
[11] Z.-Z. Bai, G.H. Golub, C.-K. Li, Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices, SIAM J. Sci. Comput. 28 (2006) 583–603.
[12] Z.-Z. Bai, G.H. Golub, L.-Z. Lu, J.-F. Yin, Block triangular and skew-Hermitian splitting methods for positive-definite linear systems, SIAM J. Sci. Comput. 23 (2005) 844–863.
[13] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, Numer. Math. 98 (2004) 1–32.
[14] M. Benzi, G.H. Golub, A preconditioner for generalized saddle point problems, SIAM J. Matrix Anal. Appl. 26 (2004) 20–41.
[15] M.-Q. Jiang, Y. Cao, On local Hermitian and skew-Hermitian splitting iteration methods for generalized saddle point problems, J. Comput. Appl. Math. 231 (2009) 973–982.
[16] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[17] Y. Saad, Iterative Methods for Sparse Linear Systems, Publishing Company, Boston, 1996.
[18] H.C. Elman, Preconditioning for the steady-state Navier–Stokes equations with low viscosity, SIAM J. Sci. Comput. 20 (1999) 157–171.