East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.190716.311216a

Vol. 7, No. 1, pp. 192-210
February 2017

A Modified Relaxed Positive-Semidefinite and Skew-Hermitian Splitting Preconditioner for Generalized Saddle Point Problems

Yang Cao[1,2,*], An Wang[3] and Yu-Juan Chen[3]

[1] *School of Transportation, Nantong University, Nantong 226019, P.R. China.*
[2] *Jiangsu Key Laboratory for Numerical Simulation of Large Scale Complex Systems, Nanjing 210023, P.R. China.*
[3] *School of Sciences, Nantong University, Nantong 226019, P.R. China.*

Received 19 July 2016; Accepted (in revised version) 31 December 2016.

Abstract. Based on the relaxed factorization techniques studied recently and the idea of the simple-like preconditioner, a modified relaxed positive-semidefinite and skew-Hermitian splitting (MRPSS) preconditioner is proposed for generalized saddle point problems. Some properties, including the eigenvalue distribution, the eigenvector distribution and the minimal polynomial of the preconditioned matrix are studied. Numerical examples arising from the mixed finite element discretization of the Oseen equation are illustrated to show the efficiency of the new preconditioner.

AMS subject classifications: 65F10, 65F50

Key words: Generalized saddle point problems, positive-semidefinite and skew-Hermitian splitting, preconditioning, Krylov subspace method.

1. Introduction

Recently, a large amount of work has been devoted to the problem of solving large linear systems in saddle point form. Such systems arise in a wide variety of scientific computing and engineering applications, such as geomechanics [15], mixed finite element approximation of elliptic partial differential equations [16], piezoelectric structures [17], meshfree approximation of elastic mechanics [21], computational fluid dynamics [22], optimization problems [25] and so on. For more background information on the applications of saddle point problems, please see [1, 13, 29] and references therein.

This work is concerned with the iterative solution of the following large sparse generalized saddle point linear system

$$\mathcal{A}\mathrm{x} \equiv \begin{bmatrix} A & B^* \\ -B & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \equiv b, \tag{1.1}$$

*Corresponding author. *Email address:* caoyangnt@ntu.edu.cn (Y. Cao)

 ©2017 Global-Science Press

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

where $A \in \mathbb{C}^{n\times n}$ is non-Hermitian positive-definite (i.e., its Hermitian part $\frac{1}{2}(A+A^*)$ is positive-definite), $B \in \mathbb{C}^{m\times n}$ $(m \le n)$ is a rectangular matrix of full row rank, B^* is the conjugate transpose of B, $C \in \mathbb{C}^{m\times m}$ is Hermitian positive-semidefinite, $f \in \mathbb{C}^n$ and $g \in \mathbb{C}^m$ are two given vectors. In particular, when $A \in \mathbb{C}^{n\times n}$ is Hermitian positive-definite and $C = 0$, the linear system (1.1) is often called the standard saddle point problem [13]. The above assumptions ensure that the block two-by-two matrix $\mathscr{A}$ is nonsingular [13, Theorem 3.4]. For the nonsingularity of a general block two-by-two matrix, please see [4, Lemma 2.1]. Thus, the solution of (1.1) exists and is unique.

In many cases, the matrices A, B and C are large sparse and iterative techniques are preferable for solving (1.1). Since the generalized saddle point matrix $\mathscr{A}$ is non-Hermitian positive-semidefinite and often ill-conditioned, preconditioning is in most cases indispensable for iterative solution of (1.1) [29]. Let

$$\mathscr{A} = \begin{bmatrix} A & B^* \\ -B & C \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & C \end{bmatrix} + \begin{bmatrix} S & B^* \\ -B & 0 \end{bmatrix} = \hat{\mathscr{H}} + \hat{\mathscr{S}}$$

be the splitting of $\mathscr{A}$ into its Hermitian and skew-Hermitian parts, where $H = \frac{1}{2}(A+A^*)$ and $S = \frac{1}{2}(A-A^*)$ are the Hermitian part and the skew-Hermitian part of the (1,1) block matrix A, respectively. Applying the Hermitian and skew-Hermitian splitting (HSS) iteration method

$$\begin{cases} (\alpha I + \hat{\mathscr{H}})\mathrm{x}^{k+\frac{1}{2}} = (\alpha I - \hat{\mathscr{S}})\mathrm{x}^k + b, \\ (\alpha I + \hat{\mathscr{S}})\mathrm{x}^{k+1} = (\alpha I - \hat{\mathscr{H}})\mathrm{x}^{k+\frac{1}{2}} + b, \end{cases} \quad (k = 0, 1, 2, \cdots .) \tag{1.2}$$

proposed by Bai, Golub and Ng in [8], Benzi and Golub constructed a class of HSS preconditioners

$$\hat{\mathscr{P}}_{HSS} = \frac{1}{2\alpha}\begin{bmatrix} \alpha I + H & 0 \\ 0 & \alpha I + C \end{bmatrix}\begin{bmatrix} \alpha I + S & B^* \\ -B & \alpha I \end{bmatrix} \tag{1.3}$$

for generalized saddle point problems (1.1), where α is a given positive parameter and I is the identity matrix with suitable dimensions. The HSS iteration method is a very promising method since it is convergent unconditionally for solving non-Hermitian positive-definite linear systems [8]. In addition, the unconditional convergence property can be extended to the generalized saddle point problems [12] and the general non-Hermitian positive-semidefinite linear systems [6]. As a preconditioner, the pre-factor has no effect on the preconditioned system. So, in many cases, we can use the following one

$$\begin{aligned} \mathscr{P}_{HSS} &= \frac{1}{\alpha}\begin{bmatrix} \alpha I + H & 0 \\ 0 & \alpha I + C \end{bmatrix}\begin{bmatrix} \alpha I + S & B^* \\ -B & \alpha I \end{bmatrix} \\ &= \begin{bmatrix} \alpha I + A + \frac{1}{\alpha}HS & B^* + \frac{1}{\alpha}HB^* \\ -B - \frac{1}{\alpha}CB & \alpha I + C \end{bmatrix} \end{aligned} \tag{1.4}$$

to replace the original HSS preconditioner (1.3). Although $\mathscr{P}_{HSS}$ no longer relates to an alternating direction iteration method (1.2), but it is of no consequence when $\mathscr{P}_{HSS}$ is used as a preconditioner for the Krylov subspace method like GMRES [18, 21]. To improve the preconditioning effects of the HSS preconditioner and accelerate the convergence rate of

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

the preconditioned iteration methods, several variants of the HSS preconditioner as well as the optimal parameters discussion can be found in [2,3,5,31].

From (1.3) or (1.4), we can see that the HSS preconditioner is a product of a block diagonal Hermitian positive-definite matrix and a normal matrix. Since the (1,1) block matrix A is non-Hermitian, it is often difficult to implement the HSS preconditioner. To remedy this, Pan, Ng and Bai [30] proposed a class of deteriorated positive-definite and skew-Hermitian splitting (DPSS) preconditioners

$$\hat{\mathscr{P}}_{DPSS} = \frac{1}{2\alpha}\begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I + C \end{bmatrix}\begin{bmatrix} \alpha I & B^* \\ -B & \alpha I \end{bmatrix} \tag{1.5}$$

for the case $C = 0$. Then it was extended by Shen [33] when C is Hermitian positive-semidefinite. In fact, the DPSS preconditioner can be induced by a matrix splitting iteration method [7, 30]. If A is Hermitian, the DPSS preconditioner $\hat{\mathscr{P}}_{DPSS}$ reduces to the HSS preconditioner $\hat{\mathscr{P}}_{HSS}$. Similarly, we can also use the following one

$$\begin{aligned} \mathscr{P}_{DPSS} =& \frac{1}{\alpha}\begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I + C \end{bmatrix}\begin{bmatrix} \alpha I & B^* \\ -B & \alpha I \end{bmatrix} \\ =& \begin{bmatrix} \alpha I + A & B^* + \frac{1}{\alpha}AB^* \\ -B - \frac{1}{\alpha}CB & \alpha I + C \end{bmatrix} \end{aligned} \tag{1.6}$$

to replace $\hat{\mathscr{P}}_{DPSS}$ (1.5) since the pre-factor has on effect on the preconditioned linear system. Numerical results indicated that the DPSS preconditioner presented better preconditioning effects than the HSS preconditioner if the experimental optimal parameters were used [30, 33]. To further improve the DPSS preconditioner, based on the relaxed techniques studied in [14, 18, 19], Fan et al. presented a class of relaxed positive-semidefinite and skew-Hermitian splitting (RPSS) preconditioners

$$\begin{aligned} \mathscr{P}_{RPSS} =& \frac{1}{\alpha}\begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & B^* \\ -B & C \end{bmatrix} \\ =& \begin{bmatrix} A & \frac{1}{\alpha}AB^* \\ -B & C \end{bmatrix} \end{aligned} \tag{1.7}$$

for the generalized saddle point problems (1.1) and studied spectrum properties of the RPSS preconditioned matrix. It directly follows from (1.4), (1.6) and (1.7) that the RPSS preconditioner is a better approximation to the generalized saddle point matrix than the HSS and the DPSS preconditioners. It is shown in [24] that the RPSS preconditioner is more efficient than the HSS and the DPSS preconditioners. In fact, if A is Hermitian, the RPSS preconditioner $\mathscr{P}_{RPSS}$ (1.7) is the simplified HSS (SHSS) preconditioner studied by Cao, Ren and Shi [20]. If $C = 0$, the RPSS preconditioner is a special case of the simple-like preconditioners studied in [27, 28].

In this paper, based on the idea of the simple-like preconditioners, by introducing a preconditioning matrix Q we propose a new modified RPSS (MRPSS) preconditioner for

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

generalized saddle point problems (1.1). Theoretical analyses show that the MRPSS preconditioned matrix has a clustered eigenvalue distribution and the corresponding preconditioned GMRES method converges within finite iteration steps. Compared with the RPSS preconditioner, the MRPSS preconditioner can result in more rapid convergence rate with suitable choices of the preconditioning matrix Q.

The remainder of this paper is as follows. The MRPSS preconditioner and the implementation aspects are presented in Section 2. In Section 3, some properties, including the eigenvalue distribution, the eigenvector distribution and the minimal polynomial of the preconditioned matrix are studied. In Section 4, numerical examples arising from the mixed finite element discretization of the Oseen equation are illustrated to show the effectiveness of the new preconditioner. Finally, our brief concluding remarks are made in Section 5.

2. The MRPSS Preconditioner

Based on the idea of the simple-like preconditioners, we propose an improved variant of the RPSS preconditioner. Let $Q \in \mathbb{C}^{n\times n}$ be a nonsingular matrix. The new preconditioner is defined as follows

$$\mathscr{P}_{MRPSS} = \begin{bmatrix} A & \frac{1}{\alpha}AQ^{-1}B^* \\ -B & C \end{bmatrix} \tag{2.1}$$

and is called the modified RPSS (MRPSS) preconditioner. By comparing the MRPSS preconditioner $\mathscr{P}_{MRPSS}$ (2.1) with the HSS preconditioner $\mathscr{P}_{HSS}$ (1.4) and the DPSS preconditioner $\mathscr{P}_{DPSS}$ (1.6), we can see that the MRPSS preconditioner is a better approximation to the generalized saddle point matrix A. By comparing the MRPSS preconditioner with the RPSS preconditioner, we can find that an additional preconditioning matrix Q is presented. Obviously, the MRPSS preconditioner reduces to the RPSS preconditioner when $Q = I$, and reduces to the generalized saddle point matrix when $Q = \frac{1}{\alpha}A$. Thus, we can choose a suitable preconditioning matrix Q for the MRPSS preconditioner to get faster convergence than the RPSS preconditioner.

In fact, the MRPSS preconditioner can be decomposed into

$$\mathscr{P}_{MRPSS} = \begin{bmatrix} A & 0 \\ -B & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & C+\frac{1}{\alpha}BQ^{-1}B^* \end{bmatrix}\begin{bmatrix} I & \frac{1}{\alpha}Q^{-1}B^* \\ 0 & I \end{bmatrix}. \tag{2.2}$$

From (2.2), we can see that the MRPSS preconditioner is nonsingular if and only if $C + \frac{1}{\alpha}BQ^{-1}B^*$ is nonsingular. Since C is Hermitian positive-semidefinite and B has full rank, in the following we always assume that Q is a positive-definite matrix, which can guarantee the nonsingularity of the MRPSS preconditioner.

Now, we consider the implementation aspects of the MRPSS preconditioner. At each iteration step of the MRPSS preconditioned Krylov subspace method, it is necessary to solve a generalized residual equation of the form

$$\mathscr{P}_{MRPSS}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} A & 0 \\ -B & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & C+\frac{1}{\alpha}BQ^{-1}B^* \end{bmatrix}\begin{bmatrix} I & \frac{1}{\alpha}Q^{-1}B^* \\ 0 & I \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \tag{2.3}$$

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

where $[z_1^*,\ z_2^*]^*$ ($z_1 \in C^n$, $z_2 \in C^m$) and $[r_1^*, r_2^*]^*$ ($r_1 \in C^n$, $r_2 \in C^m$) are the current and the generalized residual vectors, respectively. Analogous to the algorithms presented in [18,20,24], we can solve the residual equation (2.3) according to the following algorithm.

Algorithm 2.1. For a given vector $[r_1^*, r_2^*]^*$, we can compute the vector $[z_1^*,\ z_2^*]^*$ in (2.3) from the following steps:

(1) Solve $At_1 = r_1$;

(2) Solve $(C + \frac{1}{\alpha}BQ^{-1}B^*)z_2 = Bt_1 + r_2$;

(3) Solve $Qt_2 = \frac{1}{\alpha}B^*z_2$;

(4) Compute $z_1 = t_1 - t_2$.

Compared with [24, Algorithm 1], an additional linear sub-system with the coefficient matrix Q need to be solved in implementing the MRPSS preconditioner. However, this is not an issue. In practical computation, we can choose some simple preconditioning matrix Q. For example, Q can be chosen as the diagonal or the tridiagonal approximations of A. On one hand, the third step in Algorithm 2.1 can be solved easily. On the other hand, the convergence rate of the MRPSS preconditioned iteration method may be faster than that of the RPSS preconditioned iteration method.

3. Properties of the MRPSS Preconditioned Matrices

In this section, we study the spectral properties of the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$. Bounds on the eigenvalues are further studied provided that both the matrices A and Q are Hermitian positive-definite. Besides, the eigenvector distribution and the minimal polynomial of the preconditioned matrix are discussed, which are instructive for the implementation of the Krylov subspace acceleration. These theoretical results extend those in [20].

Theorem 3.1. *Let $A \in \mathbb{C}^{n\times n}$ and $Q \in \mathbb{C}^{n\times n}$ be positive-definite matrices, $B \in \mathbb{C}^{m\times n}$ have full row rank, $C \in \mathbb{C}^{m\times m}$ be Hermitian positive-semidefinite and α be a positive constant. Let the MRPSS preconditioner $\mathscr{P}_{MRPSS}$ be defined as in* (2.1). *Then the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ has an eigenvalue 1 with multiplicity n, and the remaining m eigenvalues are the eigenvalues of the matrix $(C + \frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C + BA^{-1}B^*)$. Furthermore, if $Q = \frac{1}{\alpha}A$, then all eigenvalues of the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ are 1.*

Proof. Let

$$\mathscr{Q}_{MRPSS} = \mathscr{P}_{MRPSS} - \mathscr{A} = \begin{bmatrix} 0 & (\frac{1}{\alpha}AQ^{-1} - I)B^* \\ 0 & 0 \end{bmatrix}. \tag{3.1}$$

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Then from (2.2) and (3.1), we have

$$\begin{aligned}
&\mathscr{P}_{MRPSS}^{-1}\mathscr{A}\\
=&\mathscr{P}_{MRPSS}^{-1}(\mathscr{P}_{MRPSS}-\mathscr{Q}_{MRPSS})\\
=&I-\mathscr{P}_{MRPSS}^{-1}\mathscr{Q}_{MRPSS}\\
=&I-\begin{bmatrix} I & \frac{1}{\alpha}Q^{-1}B^* \\ 0 & I \end{bmatrix}^{-1}\begin{bmatrix} I & 0 \\ 0 & C+\frac{1}{\alpha}BQ^{-1}B^* \end{bmatrix}^{-1}\begin{bmatrix} A & 0 \\ -B & I \end{bmatrix}^{-1}\begin{bmatrix} 0 & (\frac{1}{\alpha}AQ^{-1}-I)B^* \\ 0 & 0 \end{bmatrix}\\
=&\begin{bmatrix} I & A^{-1}B^*-\frac{1}{\alpha}Q^{-1}B^*(C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*) \\ 0 & (C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*) \end{bmatrix}.
\end{aligned} \tag{3.2}$$

From (3.2), we immediately obtain that the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ has an eigenvalue 1 with multiplicity n, and the remaining m eigenvalues are the eigenvalues of the matrix $(C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*)$. Furthermore, if $Q=\frac{1}{\alpha}A$, then the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ is an $(n+m)$ by $(n+m)$ identity matrix. Thus, its eigenvalues are 1. □

Theorem 3.2. *Assume that the conditions in Theorem* 3.1 *are satisfied. Further assume that the matrices A and Q are Hermitian. Let the smallest and the largest singular values of the matrix B be σ_1 and σ_m, respectively. Let the smallest eigenvalues of the A and Q be θ_1 and τ_1, respectively, and let the largest eigenvalues of A, Q, and C be θ_n, τ_n and μ_m, respectively. Then the nonunit eigenvalues of preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ are real and located in a positive interval*

$$\left[\frac{\alpha\tau_1\sigma_1^2}{(\alpha\tau_1\mu_m+\sigma_m^2)\theta_n},\ \frac{\alpha\tau_n(\mu_m\theta_1+\sigma_m^2)}{\theta_1\sigma_1^2}\right]. \tag{3.3}$$

Proof. From (3.2), we know that the nonunit eigenvalues of preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ are the eigenvalues of the matrix $(C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*)$. Since all the matrices A, Q and C are Hermitian, the eigenvalues of the matrix $(C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*)$ are real.

Now, we study the bounds of the eigenvalues of the matrix $(C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*)$. Assume that θ be an eigenvalue of $(C+\frac{1}{\alpha}BQ^{-1}B^*)^{-1}(C+BA^{-1}B^*)$ and x be the normalized eigenvector, i.e. $\|x\|_2=1$. Then we have

$$\theta=\frac{x^*Cx+x^*BA^{-1}B^*x}{x^*Cx+\frac{1}{\alpha}x^*BQ^{-1}B^*x}.$$

From [32, Theorem 1.22], we have

$$x^*Cx+x^*BA^{-1}B^*x\le x^*Cx+\frac{1}{\theta_1}x^*BB^*x\le\mu_m+\frac{\sigma_m^2}{\theta_1}$$

and

$$x^*Cx+x^*BA^{-1}B^*x\ge\frac{1}{\theta_n}x^*BB^*x\ge\frac{\sigma_1^2}{\theta_n}.$$

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Similarly,

$$\frac{\sigma_1^2}{\alpha\tau_n} \le x^*Cx + \frac{1}{\alpha}x^*BQ^{-1}B^*x \le \mu_m + \frac{\sigma_m^2}{\alpha\tau_1}.$$

Thus, the remaining m eigenvalues are located in the positive interval (3.3). □

The termination of a preconditioned Krylov subspace method is not only related to the location of the eigenvalues, but also the number of the corresponding linearly independent eigenvectors. Let $null(W)$ and $range(W)$ denote the null space and the range space of the matrix W, respectively. Let $dim(\cdot)$ denote the dimension of the corresponding space. We present the eigenvector distribution of the MRPSS preconditioned matrix in the following theorem.

Theorem 3.3. *Let the MRPSS preconditioner $\mathscr{P}_{MRPSS}$ be defined as in* (2.1)*, then the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ has $n+i+j$ $(i+j \le m)$ linearly independent eigenvectors. There are*

(1) n eigenvectors of the form $\begin{bmatrix} u_p^1 \\ 0 \end{bmatrix}$ that correspond to the eigenvalue 1, where $u_p^1 \in \mathbb{C}^n$ $(p = 1, \cdots, n)$ are arbitrary linearly independent vectors.

(2) i $(0 \le i \le m)$ eigenvectors of the form $\begin{bmatrix} u_p^2 \\ v_p^2 \end{bmatrix}$ that correspond to the eigenvalue 1, where u_p^2 are arbitrary vectors, the vectors $v_p^2 \ne 0$ satisfy $(\frac{1}{\alpha}AQ^{-1} - I)B^*v_p^2 = 0$ and $i = dim\{null(\frac{1}{\alpha}AQ^{-1} - I) \bigcap range(B^*)\}$.

(3) j $(0 \le j \le m)$ eigenvectors of the form $\begin{bmatrix} u_p^3 \\ v_p^3 \end{bmatrix}$ that correspond to eigenvalues $\lambda \ne 1$, where the vectors $v_p^3 \ne 0$ satisfy the generalized eigenvalue problem $(C + BA^{-1}B^*)v_p^3 = \lambda(C + \frac{1}{\alpha}BQ^{-1}B^*)v_p^3$ and $u_p^3 = \frac{1}{1-\lambda}(\frac{\lambda}{\alpha}Q^{-1} - A^{-1})B^*v_p^3$. In addition, if $v_p^3 \in null(C)$, then $u_p^3 \in null(B)$, and vice versa.

Proof. The eigenvector distribution of the MRPSS preconditioned matrix can be derived by considering the following generalized eigenvalue problem

$$\begin{bmatrix} A & B^* \\ -B & C \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \lambda\begin{bmatrix} A & \frac{1}{\alpha}AQ^{-1}B^* \\ -B & C \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}, \tag{3.4}$$

where λ is an eigenvalue of the MRPSS preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ and $\begin{bmatrix} u \\ v \end{bmatrix}$ is the corresponding eigenvector. Expanding (3.4) out, we have

$$(1-\lambda)Au = \left(\frac{\lambda}{\alpha}AQ^{-1} - I\right)B^*v, \tag{3.5}$$

and

$$(1-\lambda)Bu = (1-\lambda)Cv. \tag{3.6}$$

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

According to the eigenvalue distribution of the preconditioned matrix, we first consider the case $\lambda = 1$. Thus, (3.6) holds naturally and (3.5) becomes

$$\left(\frac{1}{\alpha}AQ^{-1} - I\right)B^*v = 0. \tag{3.7}$$

If $v = 0$, then the equation (3.7) satisfies naturally. Thus, there are n eigenvectors of the form $\begin{bmatrix} u_p^1 \\ 0 \end{bmatrix}$ that correspond to the eigenvalue 1, where $u_p^1 \in \mathbb{C}^n$ $(p = 1, \cdots, n)$ are arbitrary linearly independent vectors. If there exists any $v \neq 0$ which satisfies (3.7), then there will be i $(0 \leq i \leq m)$ eigenvectors of the form $\begin{bmatrix} u_p^2 \\ v_p^2 \end{bmatrix}$ that correspond to the eigenvalue 1, where u_p^2 are arbitrary vectors, the vectors $v_p^2 \neq 0$ satisfy $(\frac{1}{\alpha}AQ^{-1} - I)B^*v_p^2 = 0$ and $i = dim\{null(\frac{1}{\alpha}AQ^{-1} - I) \bigcap range(B^*)\}$.

Next, we consider the case $\lambda \neq 1$. From (3.5) we have

$$u = \frac{1}{1-\lambda}\left(\frac{\lambda}{\alpha}Q^{-1} - A^{-1}\right)B^*v. \tag{3.8}$$

Substituting the above equation into (3.6), we obtain that the vectors v satisfy

$$(C + BA^{-1}B^*)v = \lambda\left(C + \frac{1}{\alpha}BQ^{-1}B^*\right)v. \tag{3.9}$$

Equation (3.9) is trivially satisfied by $v = 0$. However, this can not happen. Otherwise, from (3.8) we have $u = 0$, which contradicts with $[u^* \ v^*]^*$ being an eigenvector. If there exists any $v \neq 0$ satisfies (3.9), then there will be j $(0 \leq j \leq m)$ linearly independent eigenvectors of the form $\begin{bmatrix} u_p^3 \\ v_p^3 \end{bmatrix}$ that correspond to eigenvalues $\lambda \neq 1$, where the vectors $v_p^3 \neq 0$ satisfy the generalized eigenvalue problem (3.9) and $u_p^3 = \frac{1}{1-\lambda}(\frac{\lambda}{\alpha}Q^{-1} - A^{-1})B^*v_p^3$. For such case, if $v_p^3 \in null(C)$, then from (3.6) we have $Bu_p^3 = 0$, i.e. $u_p^3 \in null(B)$. Conversely, if $u_p^3 \in null(B)$, then from (3.6) we have $Cv_p^3 = 0$, i.e. $v_p^3 \in null(C)$.

The linearly independence of $n + i + j$ eigenvectors can be proved by the same method studied in [20, Theorem 3.2] with only technical modifications. Hence, we omit the rest of the proof here. □

Based on the block structure (3.2), an upper bound of degree of the minimal polynomial of the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ can be obtained. We show this result in the following theorem.

Theorem 3.4. *Let the MRPSS preconditioner $\mathscr{P}_{MRPSS}$ be defined as in* (2.1)*. Then, both the degree of the minimal polynomial of the preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ and the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{MRPSS}^{-1}\mathscr{A}, b)$ are at most $m + 1$.*

Proof. The MRPSS preconditioned matrix $\mathscr{P}_{MRPSS}^{-1}\mathscr{A}$ has the same block structure with the SHSS preconditioned matrix studied in [20, Theorem 3.3]. So, the proof of the theorem is essentially analogous to the proof of Theorem 3.3 in [20], with only technical modifications. □

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Remark 3.1. The theoretical results presented in Theorem 3.4 is very important, since it determines the convergence behavior of a Krylov subspace method, such as GMRES [32]. From Theorem 3.4, we know that any Krylov subspace iterative method with an optimality or Galerkin property (for example GMRES) will terminate in at most $m+1$ iterations with the solution to a linear system of the form (1.1) if the MRPSS preconditioner is used.

4. Numerical Results

In this section, our goal is to compare the performance of the MRPSS preconditioner with the HSS, the DPSS and the RPSS preconditioners. To do this, we provide numerical experiments on generalized saddle point linear systems from the Oseen equation

$$\begin{cases} -\nu\Delta u + w\cdot\bigtriangledown u + \bigtriangledown p = f, \\ \bigtriangledown\cdot u = 0, \end{cases} \qquad in\ \Omega \tag{4.1}$$

which is obtained when the steady-state Navier-Stokes equation is linearized by the Picard iteration. In (4.1), $\Omega \subset \mathbb{R}^2$ is a bounded domain, $\nu > 0$ is the viscosity parameter, u and p stand for the unknown velocity and the unknown pressure, respectively, and w is the known velocity from the previous Picard iteration. To generate the test problems, the leaky-lid driven cavity problem in a square domain ($0 \le x \le 1, 0 \le y \le 1$) is considered and the IFISS software [23] written by Elman, Ramage and Silvester is used. To discrete the Oseen equation (4.1), the stabilized Q1-P0 mixed finite element methods on both uniform and stretched grids are adopted and the stabilization parameter is taken as 0.25. For the case of stretched grids, the stretching is done in both the horizontal and vertical directions, resulting in rather fine grids near the boundaries. In actual computations, we take three viscosity values $\nu = 1, 0.1, 0.01$. For each viscosity value, four increasing grids, i.e, 8×8, 16×16, 32×32 and 64×64 grids are considered.

In actual implementations, we solve the discretized generalized saddle point linear systems by the left-preconditioned GMRES methods incorporated with the discussed four preconditioners and compare the performance of different preconditioners from aspects of the number of iteration steps (denoted by 'IT') and the elapsed CPU times in seconds (denoted by 'CPU'). The preconditioned GMRES methods are started from the zero initial guess and terminated once the current residual satisfies

$$\text{RES} = \frac{\|b - \mathcal{A}\mathrm{x}^k\|_2}{\|b\|_2} \le 10^{-6}$$

or if the iteration step exceeds the largest prescribed iteration step $k_{max} = 1500$. The subsystems of linear equations arising in the applications of the discussed preconditioners are solved by the sparse Cholesky factorization when the coefficient matrix is symmetric and by the sparse LU factorization when the coefficient matrix is nonsymmetric. All codes are run in MATLAB (version R2010b) in double precision and all experiments are performed on an Intel Core (i5-4288u CPU, 8G RAM) Windows 7 system.

To implement these preconditioners efficiently and obtain fast convergence rate of the corresponding preconditioned GMRES methods, the parameters and the preconditioning

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 1: The parameters used in different preconditioners (uniform grids).

ν	α	Grids			
		8×8	16×16	32×32	64×64
$\nu=1$	α_{HSS}	1.3898	1.4964	1.5607	1.5958
	α_{DPSS}	1.8916	2.0155	2.0866	2.1248
	α_{RPSS}	2.2639	2.5263	2.6720	2.7488
	α_{MRPSS1}	1.0446	1.0528	1.0568	1.0587
	α_{MRPSS2}	1.0320	1.0384	1.0415	1.0430
$\nu=0.1$	α_{HSS}	0.4568	0.3359	0.2612	0.2166
	α_{DPSS}	1.0737	0.9421	0.8456	0.7792
	α_{RPSS}	0.6653	0.5322	0.4334	0.3682
	α_{MRPSS1}	1.0052	1.0115	1.0206	1.0316
	α_{MRPSS2}	1.0036	1.0084	1.0152	1.0233
$\nu=0.01$	α_{HSS}	0.4232	0.2943	0.2068	0.1460
	α_{DPSS}	1.0477	0.8892	0.7548	0.6398
	α_{RPSS}	0.6290	0.4713	0.3439	0.2477
	α_{MRPSS1}	1.0002	1.0003	1.0005	1.0008
	α_{MRPSS2}	1.0001	1.0002	1.0003	1.0005

matrices involved in these preconditioners should be chosen appropriately. The experimental optimal parameters, which lead to the least iteration number of the corresponding preconditioned iteration methods, are often used to check the efficiency of the discussed preconditioners. However, this is not practical in actual computation. In this paper, the practical formula proposed by Huang [26] for computing the optimal parameters of the HSS iteration method is used for computing the quasi-optimal parameters of the HSS preconditioner. For other three preconditioners, by making use the techniques studied in [11, 31], the following estimates

$$\alpha_{DPSS}=\sqrt{\frac{\|A\|_F\cdot\|B\|_F}{\|I_n\|_F\cdot\|I_m\|_F}},\qquad \alpha_{RPSS}=\frac{\|A\|_F}{\|I_n\|_F},\qquad \alpha_{MRPSS}=\frac{\|A\|_F}{\|Q\|_F} \tag{4.2}$$

are used for the DPSS preconditioner, the RPSS preconditioner and the MRPSS preconditioner, respectively. In the above expressions, $\|A\|_F$ denotes the Frobenius norm of the matrix A, and I_n and I_m denote the $n\times n$ and $m\times m$ identity matrices, respectively. As for the preconditioning matrix Q, we take the following two choices

Case I: Q=diag(A),

Case II: Q=tridiag(A).

The corresponding MRPSS preconditioners are denoted by the MRPSS1 preconditioner and the MRPSS2 preconditioner, respectively.

We first present the numerical results of different preconditioned GMRES methods for solving the Oseen equation discretized by uniform grids. Specifically, in Table 1, the parameters used in the discussed preconditioners are listed. In Table 2, we present the numerical

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 2: Numerical results for the Oseen equation with $\nu = 1$ (uniform grids).

Preconditioner		Grids			
		8×8	16×16	32×32	64×64
I	IT	77	171	368	404
	CPU	0.0380	0.2376	3.1823	10.7807
	RES	7.9934e-7	9.8576e-7	9.4717e-7	9.9803e-7
HSS	IT	45	96	202	219
	CPU	0.0252	0.0895	0.9615	4.6796
	RES	9.5089e-7	9.6949e-7	8.7427e-7	9.8694e-7
DPSS	IT	50	107	222	241
	CPU	0.0275	0.0856	0.9959	4.6410
	RES	9.6937e-7	9.7116e-7	8.6706e-7	9.8520e-7
RPSS	IT	11	17	26	39
	CPU	0.0055	0.0088	0.0395	0.2944
	RES	7.2096e-7	6.6169e-7	6.4856e-7	8.8749e-7
MRPSS1	IT	11	18	26	39
	CPU	0.0086	0.0076	0.0462	0.2926
	RES	8.7736e-7	3.6183e-7	8.7005e-7	9.2790e-7
MRPSS2	IT	12	17	25	38
	CPU	0.0069	0.0228	0.0894	0.6298
	RES	3.5822e-7	8.7911e-7	8.7209e-7	7.3029e-7

results of different preconditioned GMRES methods for the discretized Oseen equation with the viscosity parameter $\nu = 1$. In Table 3 and Table 4, the same items are listed for $\nu = 0.1$ and $\nu = 0.01$, respectively. From numerical results listed in Tables 2-4, we can find that

- Both the iteration counts and the elapsed CPU times show that the GMRES method converges slowly if no preconditioner is used. All the discussed preconditioners can accelerate the convergence rate of the GMRES method efficiently.
- With the practical formulas studied in [11, 31], the DPSS preconditioner is less efficient than the HSS preconditioner. This means that the parameter estimate method presented in [11,31] may not be a good choice for the DPSS preconditioner. It should be noted that the quasi-optimal parameters of the PSS iteration method (preconditioner) have been studied in [9, 10] when the coefficient matrix is non-Hermitian positive-definite. However, there is no discussion on the quasi-optimal parameters of the DPSS preconditioners when the coefficient matrix has the saddle point structure.
- The RPSS preconditioner and the MRPSS preconditioners are more efficient than the HSS preconditioner and the DPSS preconditioner. When $\nu = 1$ and 0.1, the iteration counts of the RPSS preconditioned GMRES method are almost the same as those of the MRPSS preconditioned GMRES methods. But when $\nu = 0.01$, the iteration counts of the MRPSS preconditioned GMRES methods are less than those of the RPSS preconditioned GMRES method.

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 3: Numerical results for the Oseen equation with $\nu = 0.1$ (uniform grids).

Preconditioner		Grids			
		8×8	16×16	32×32	64×64
I	IT	84	157	275	539
	CPU	0.0736	0.2097	1.8744	18.3102
	RES	8.9902e-7	9.0207e-7	9.5096e-7	9.7511e-7
HSS	IT	36	78	155	300
	CPU	0.0178	0.0752	0.6436	7.8936
	RES	8.6606e-7	7.7381e-7	8.9792e-7	9.6708e-7
DPSS	IT	56	115	213	416
	CPU	0.0287	0.1066	0.9422	12.3943
	RES	7.1115e-7	8.7010e-7	8.6323e-7	9.4510e-7
RPSS	IT	16	23	33	46
	CPU	0.0086	0.0132	0.0611	0.3728
	RES	6.4401e-7	7.2706e-7	8.7081e-7	9.0767e-7
MRPSS1	IT	14	22	32	46
	CPU	0.0140	0.0188	0.0506	0.3765
	RES	5.9803e-7	4.9164e-7	6.84172e-7	8.2645e-7
MRPSS2	IT	14	21	31	45
	CPU	0.0058	0.0208	0.0960	0.7310
	RES	7.3469e-7	8.1662e-7	9.8837e-7	9.7574e-7

In Tables 5-8, we list the numerical results for the Oseen equation discretized by stretched grids. More specifically, the parameters used in different preconditioners are listed in Table 5. The computational results of different preconditioend GMRES methods for solving discretized Oseen equation with $\nu = 1$, 0.1 and 0.01 are listed in Table 6, Table 7 and Table 8, respectively. In Table 8, '-' means that the corresponding iteration method does not converge to the solution within the prescribed maximal iteration number k_{max}.

From these tables, we can also find that all the discussed preconditioners can accelerate the convergence rate of the GMRES method efficiently. The RPSS preconditioner studied in [24] and the MRPSS preconditioner proposed in this paper are much efficient than the HSS preconditioner and the DPSS preconditioner, too. No matter whether the viscosity parameter is large or small, the iteration counts of the MRPSS preconditioned GMRES method are less than those of the RPSS preconditioned GMRES method. But the elapsed CPU times show that the MRPSS1 preconditioner is more efficient than the MRPSS2 preconditioner although the iteration steps of the MRPSS2 preconditioned GMRES method are less than those of the MRPSS1 preconditioned GMRES method. This is because an additional subsystem of linear equations with the tridiagonal coefficient matrix Q need to be solved.

To further show the convergence behavior of the preconditioned GMRES methods, we plot the residual curves of the preconditioned GMRES methods for solving the Oseen equation discretized by uniform 32×32 grids in Fig. 1. The residual curves for stretched grid

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 4: Numerical results for the Oseen equation with $\nu = 0.01$ (uniform grids).

Preconditioner		Grids			
		8×8	16×16	32×32	64×64
I	IT	141	343	725	1336
	CPU	0.1349	0.9135	12.1683	108.5875
	RES	7.7991e-7	9.4375e-7	9.9915e-7	9.9270e-7
HSS	IT	79	162	382	803
	CPU	0.0347	0.1754	2.9168	47.0472
	RES	8.9961e-7	8.8868e-7	9.6785e-7	9.9664e-7
DPSS	IT	118	256	573	1116
	CPU	0.0577	0.3683	6.3451	85.3692
	RES	7.7519e-7	9.8584e-7	9.9255e-7	9.9576e-7
RPSS	IT	35	51	69	90
	CPU	0.0116	0.0450	0.1460	1.0214
	RES	7.9899e-7	8.8142e-7	9.8534e-7	9.2677e-7
MRPSS1	IT	23	36	50	70
	CPU	0.0098	0.0328	0.0976	0.6739
	RES	8.8716e-7	9.0631e-7	8.7477e-7	9.9234e-7
MRPSS2	IT	17	27	45	66
	CPU	0.0092	0.0287	0.1219	1.1477
	RES	3.8672e-7	9.9900e-7	7.8674e-7	8.8298e-7

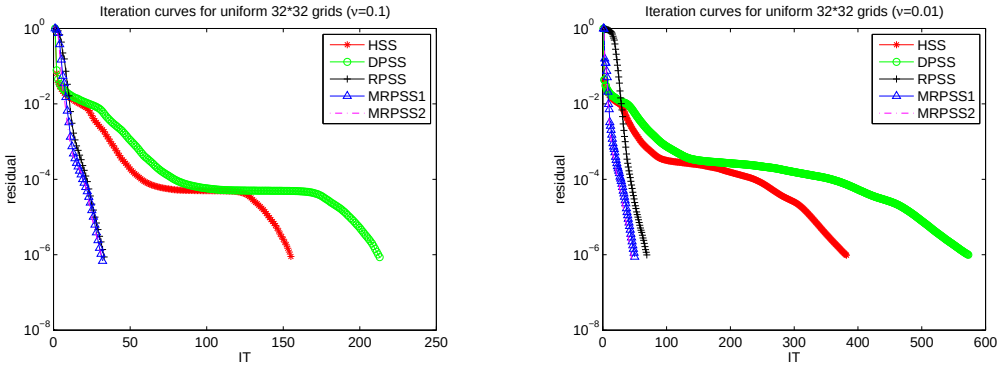


Figure 1: Residual curves for uniform 32×32 grids with $\nu = 0.1$ (left) and $\nu = 0.01$ (right).

cases are plotted in Fig. 2. These figures clearly show that the RPSS preconditioner and the MRPSS preconditioners are better than the HSS preconditioner and the DPSS preconditioner. And the proposed MRPSS preconditioners are slightly better than the RPSS preconditioner.

In order to investigate the dependence of the MRPSS and the RPSS preconditioned GMRES methods on the iteration parameter α, we illustrate the changing of their iteration steps with respect to α from 0.1 to 1.5 for uniform 32×32 grid cases in Fig. 3. The stretched

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 5: The parameters used in different preconditioners (stretched grids).

ν	α	Grids			
		8×8	16×16	32×32	64×64
$\nu=1$	α_{HSS}	1.5527	2.1519	3.0683	4.4122
	α_{DPSS}	2.1080	2.6126	3.2416	4.0196
	α_{RPSS}	2.5287	3.6391	5.2573	7.6021
	α_{MRPSS1}	1.0593	1.1317	1.1965	1.2410
	α_{MRPSS2}	1.0381	1.0712	1.0973	1.1138
$\nu=0.1$	α_{HSS}	0.4778	0.3817	0.3766	0.4661
	α_{DPSS}	1.1468	1.0782	1.1267	1.3049
	α_{RPSS}	0.6748	0.5931	0.6267	0.7987
	α_{MRPSS1}	1.0080	1.0442	1.1267	1.2106
	α_{MRPSS2}	1.0052	1.0252	1.0656	1.1014
$\nu=0.01$	α_{HSS}	0.4350	0.3016	0.2123	0.1538
	α_{DPSS}	1.1127	0.9683	0.8432	0.7438
	α_{RPSS}	0.6291	0.4721	0.3469	0.2576
	α_{MRPSS1}	1.0003	1.0008	1.0036	1.0157
	α_{MRPSS2}	1.0010	1.0005	1.0020	1.0086

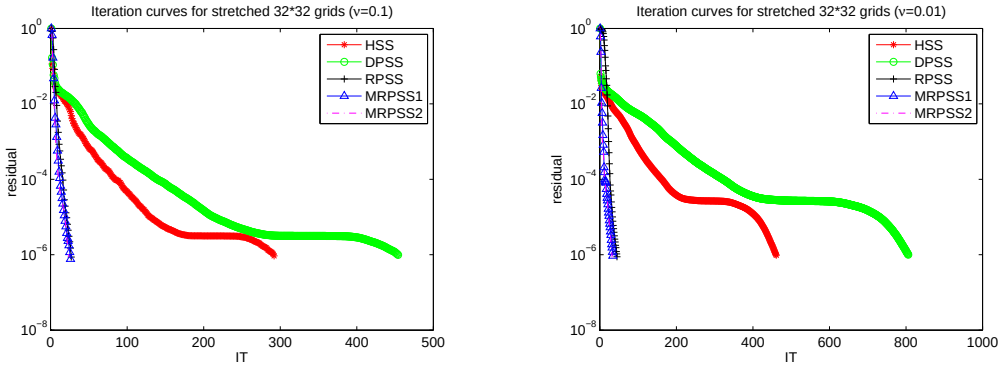


Figure 2: Residual curves for stretched 32×32 grids with $\nu=0.1$ (left) and $\nu=0.01$ (right).

32×32 grid cases are illustrated in Fig. 4. From Figs. 3 and 4, we see that the MRPSS preconditioned GMRES methods need less iteration counts than the RPSS preconditioned GMRES method. What's more, the MRPSS preconditioner is not sensitive to the parameter α, in the sense that the iteration count does not change dramatically, while the RPSS preconditioner is very sensitive to the parameter α. So, another advantage of the MRPSS preconditioner is that a fairly wide range of values of the parameter α can produce similar fast convergence results.

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 6: Numerical results for the Oseen equation with $\nu = 1$ (stretched grids).

Preconditioner		Grids			
		8×8	16×16	32×32	64×64
I	IT	100	358	469	807
	CPU	0.0759	0.9725	5.1236	40.1721
	RES	8.9481e-7	9.3131e-7	9.7681e-7	9.9628e-7
HSS	IT	62	211	218	315
	CPU	0.0394	0.2715	1.0842	8.4585
	RES	5.3165e-7	9.6384e-7	9.9268e-7	9.9557e-7
DPSS	IT	68	226	222	301
	CPU	0.0413	0.2760	0.9993	6.8542
	RES	5.1337e-7	9.1037e-7	9.9807e-7	9.9581e-7
RPSS	IT	11	13	20	31
	CPU	0.0079	0.0140	0.0334	0.2103
	RES	2.9117e-7	7.8714e-7	9.4336e-7	9.7330e-7
MRPSS1	IT	11	15	20	27
	CPU	0.0074	0.0102	0.0346	0.1881
	RES	4.6847e-7	4.4218e-7	6.8368e-7	8.6749e-7
MRPSS2	IT	10	14	19	22
	CPU	0.0065	0.0265	0.0767	0.4916
	RES	8.6903e-7	4.7249e-7	5.9542e-7	8.0976e-7

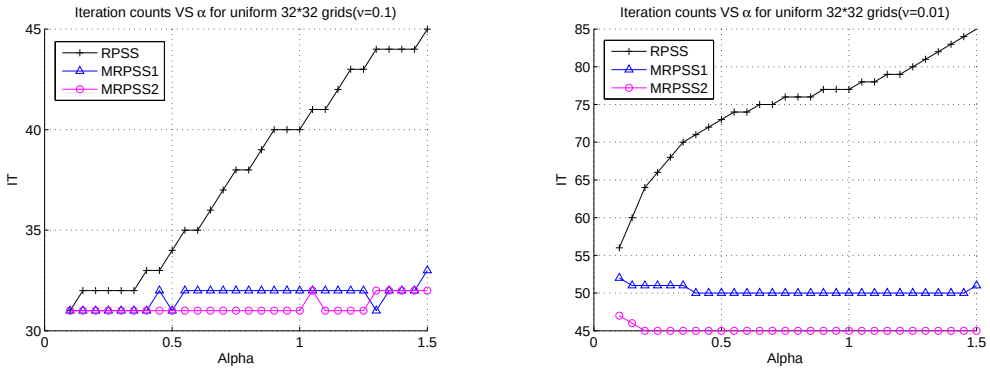


Figure 3: Iteration counts VS α for uniform 32×32 grids with $\nu = 0.1$ (left) and $\nu = 0.01$ (right).

5. Conclusion

In this paper, based on the idea of the simple-like preconditioners, by introducing a preconditioning matrix we proposed a modified relaxed positive-semidefinite and skew-Hermitian splitting (MRPSS) preconditioner for generalized saddle point problems. The new preconditioner can include the simple-like preconditioners [27,28], the SHSS preconditioner [20] and the RPSS preconditioner [24] as special cases. The eigenvalue distribu-

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 7: Numerical results for the Oseen equation with $\nu = 0.1$ (stretched grids).

Preconditioner		Grids			
		8×8	16×16	32×32	64×64
I	IT	91	230	632	961
	CPU	0.0708	0.4189	9.2485	56.6934
	RES	9.0230e-7	9.7764e-7	9.6484e-7	9.9282e-7
HSS	IT	37	103	292	381
	CPU	0.0330	0.0934	1.8062	11.8118
	RES	6.3465e-7	9.2779e-7	9.3226e-7	9.8711e-7
DPSS	IT	59	164	454	581
	CPU	0.0307	0.1620	3.8916	23.4390
	RES	9.5384e-7	9.7319e-7	9.8890e-7	9.9610e-7
RPSS	IT	15	19	27	40
	CPU	0.0092	0.0171	0.0379	0.3002
	RES	5.6414e-7	8.6680e-7	8.6545e-7	8.8590e-7
MRPSS1	IT	14	19	26	34
	CPU	0.0101	0.0168	0.0449	0.2406
	RES	5.8934e-7	9.9800e-7	7.6689e-7	9.4299e-7
MRPSS2	IT	13	18	24	34
	CPU	0.0075	0.0227	0.0875	0.5891
	RES	6.5247e-7	6.3295e-7	8.6466e-7	8.2642e-7

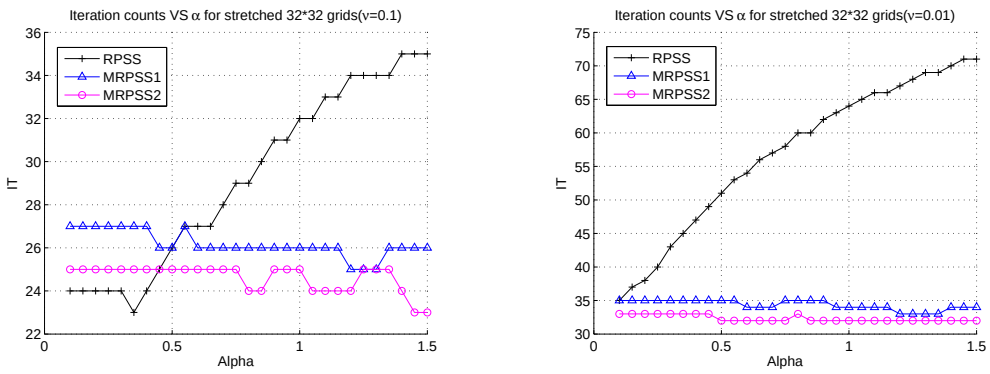


Figure 4: Iteration counts VS α for stretched 32×32 grids with $\nu = 0.1$ (left) and $\nu = 0.01$ (right).

tion, the eigenvector distribution and the minimal polynomial of the MRPSS preconditioned matrix are studied. In particular, we presented eigenvalue bounds of the MRPSS preconditioned matrix when both the (1,1) block matrix A and the preconditioning matrix Q are Hermitian. Numerical examples arising from the mixed finite element discretization of the Oseen equation show that the new MRPSS preconditioner is better than some existing preconditioners.

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

Table 8: Numerical results for the Oseen equation with $\nu = 0.01$ (stretched grids).

Preconditioner		Grids			
		8×8	16×16	32×32	64×64
I	IT	139	407	1017	-
	CPU	0.1166	1.2805	23.9332	-
	RES	9.6873e-7	9.8399e-7	9.9704e-7	-
HSS	IT	81	193	461	1118
	CPU	0.0525	0.2316	4.1700	87.1386
	RES	8.9100e-7	9.5386e-7	9.6618e-7	9.9921e-7
DPSS	IT	123	321	806	-
	CPU	0.0701	0.5661	12.3913	-
	RES	6.9132e-7	9.9705e-7	9.9739e-7	-
RPSS	IT	34	41	45	56
	CPU	0.0197	0.0297	0.0739	0.4833
	RES	5.4886e-7	8.0892e-7	8.8264e-7	9.7544e-7
MRPSS1	IT	22	29	34	47
	CPU	0.0132	0.0133	0.0637	0.3879
	RES	7.9839e-7	7.9382e-7	9.3305e-7	8.9379e-7
MRPSS2	IT	16	25	32	45
	CPU	0.0054	0.0251	0.0860	0.6890
	RES	5.8207e-7	9.0554e-7	8.5690e-7	8.1981e-7

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 11301290, 11572210), the Open Project of Jiangsu Key Lab for Numerical Simulation of Large Scale Complex Systems (No. 201605) and the Qing Lan Project of Excellent-Young-Backbone Teacher of Jiangsu Province.

References

[1] Z.-Z. Bai, *Structured preconditioners for nonsingular matrices of block two-by-two structures*, Math. Comput., **75**, 791-815 (2006).

[2] Z.-Z. Bai, *Optimal parameters in the HSS-like methods for saddle-point problems*, Numer. Linear Algebra Appl., **16**, 447-479 (2009).

[3] Z.-Z. Bai, *Block alternating splitting implicit iteration methods for saddle-point problems from time-harmonic eddy current models*, Numer. Linear Algebra Appl., **19**, 914-936 (2012).

[4] Z.-Z. Bai, *Eigenvalue estimates for saddle point matrices of Hermitian and indefinite leading blocks*, J. Comput. Appl. Math., **237**, 295-306 (2013).

[5] Z.-Z. Bai and G.H. Golub, *Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems*, IMA J. Numer. Anal., **27**, 1-23 (2007).

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

[6] Z.-Z. Bai, G.H. Golub and C.-K. Li, *Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices*, Math. Comput., **76**, 287-298 (2007).
[7] Z.-Z. Bai, G.H. Golub, L.-Z. Lu and J.-F. Yin, *Block triangular and skew-Hermitian splitting methods for positive-definite linear systems*, SIAM J. Sci. Comput., **26**, 844-863 (2005).
[8] Z.-Z. Bai, G.H. Golub and M.K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl. **24**, 603-626 (2003).
[9] Z.-Z. Bai and A. Hadjidimos, *Optimization of extrapolated Cayley transform with non-Hermitian positive definite matrix*, Linear Algebra Appl., **463**, 322-339 (2014).
[10] Z.-Z. Bai, J.-F. Yin and Y.-F. Su, *A shift-splitting preconditioner for non-Hermitian positive definite matrices*, J. Comput. Math., **24**, 539-552 (2006).
[11] M. Benzi, S. Deparis, G. Grandperrin and A. Quarteroni, *Parameter estimates for the relaxed dimensional factorization preconditioner and application to hemodynamics*, Comput. Methods Appl. Mech. Engrg., **300**, 129-145 (2016).
[12] M. Benzi and G.H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl. **26**, 20-41 (2004).
[13] M. Benzi, G.H. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., **14**, 1-137 (2005).
[14] M. Benzi, M.K. Ng, Q. Niu and Z. Wang, *A relaxed dimensional fractorization preconditioner for the incompressible Navier-Stokes equations*, J. Comput. Phys., **230**, 6185-6202 (2011).
[15] L. Bergamaschi and Á. Martínez, *RMCP: Relaxed Mixed Constraint Preconditioners for saddle point linear systems arising in geomechanics*, Comput. Methods Appl. Mech. Engrg., **221-222**, 54-62 (2012).
[16] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York (1991).
[17] T.-Q. Bui, M.N. Nguyen, C.-Z. Zhang and D.A.K. Pham, *An efficient meshfree method for analysis of two-dimensional piezoelectric structures*, Smart. Mater. Struct., **20**, 065016, 11 pages (2011).
[18] Y. Cao, J.-L. Dong and Y.-M. Wang, *A relaxed deteriorated PSS preconditioner for nonsymmetric saddle point problems from the steady Navier-Stokes equation*, J. Comput. Appl. Math., **273**, 41-60 (2015).
[19] Y. Cao, S.-X. Miao and Y.-S. Cui, *A relaxed splitting preconditioner for generalized saddle point problems*, Comput. Appl. Math., **34**, 865-879 (2015).
[20] Y. Cao, Z.-R. Ren and Q. Shi, *A simplified HSS preconditioner for generalized saddle point problems*, BIT Numer. Math., **56**, 423-439 (2016).
[21] Y. Cao, L.-Q. Yao, M.-Q. Jiang and Q. Niu, *A relaxed HSS preconditioner for saddle point problems from meshfree discretization*, J. Comput. Math., **31**, 398-421 (2013).
[22] H.C. Elman, D.J. Silvester and A.J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics (2nd edn)*, Oxford University Press, Oxford (2014).
[23] H.C. Elman, A. Ramage and D.J. Silvester, *IFISS: a Matlab toolbox for modelling incompressible flow*, ACM Trans. Math. Softare, **33**, Article 14 (2007).
[24] H.-T. Fan, B. Zheng and X.-Y. Zhu, *A relaxed positive semi-definite and skew-Hermitian splitting preconditioner for non-Hermitian generalized saddle point problems*, Numer. Algor., **72**, 813-834 (2016).
[25] C. Greif, E. Moulding and D. Orban, *Bounds on eigenvalues of matrices arising from interior-point methods*, SIAM, J. Optim., **24**, 49-83 (2014).
[26] Y.-M. Huang, *A practical formula for computing optimal parameters in the HSS iteration methods*, J. Comput. Appl. Math., **255**, 142-149 (2014).

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press

[27] C. Li and C. Vuik, *Eigenvalue analysis of the SIMPLE preconditioning for incompressible flow*, Numer. Linear Algebra Appl., **11**, 511-523 (2004).
[28] Z.-Z. Liang and G.-F. Zhang, *SIMPLE-like preconditioners for saddle point problems from the steady Navier-Stokes equations*, J. Comput. Appl. Math., **302**, 211-223 (2016).
[29] J. Pestana and A.J. Wathen, *Natural preconditioning and iterative methods for saddle point problems*, SIAM Review, **57**, 71-91 (2015).
[30] J.-Y. Pan, M.K. Ng and Z.-Z. Bai, *New preconditioners for saddle point problems*, Appl. Math. Comput., **172**, 762-771 (2006).
[31] Z.-R. Ren and Y. Cao, *An alternating positive-semidefinite splitting preconditioner for saddle point problems from time-harmonic eddy current models*, IMA J. Numer. Anal., **36**, 922-946 (2016).
[32] Y. Saad, *Iterative Methods for Sparse Linear Systems (2nd edn)*, SIAM: Philadelphia (2003).
[33] S.-Q. Shen, *A note on PSS preconditioners for generalized saddle point problems*, Appl. Math. Comput., **237**, 723-729 (2014).

https://doi.org/10.4208/eajam.190716.311216a Published online by Cambridge University Press