

IMPROVED RELAXED POSITIVE-DEFINITE AND SKEW-HERMITIAN SPLITTING PRECONDITIONERS FOR SADDLE POINT PROBLEMS*

Yang Cao

School of Transportation, Nantong University, Nantong 226019, China

Email: caoyangnt@ntu.edu.cn

Zhiru Ren¹⁾

School of Statistics and Mathematics, Central University of Finance and Economics,

Beijing 100081, China

Email: renzr@lsec.cc.ac.cn

Linquan Yao

School of Urban Rail Transportation, Soochow University, Suzhou, 215006, China

Email: lqyao@suda.edu.cn

Abstract

We establish a class of improved relaxed positive-definite and skew-Hermitian splitting (IRPSS) preconditioners for saddle point problems. These preconditioners are easier to be implemented than the relaxed positive-definite and skew-Hermitian splitting (RPSS) preconditioner at each step for solving the saddle point problem. We study spectral properties and the minimal polynomial of the IRPSS preconditioned saddle point matrix. A theoretical optimal IRPSS preconditioner is also obtained. Numerical results show that our proposed IRPSS preconditioners are superior to the existing ones in accelerating the convergence rate of the GMRES method for solving saddle point problems.

Mathematics subject classification: 65F10, 65F50

Key words: Saddle point problems, Preconditioning, RPSS preconditioner, Eigenvalues, Krylov subspace method.

1. Introduction

We consider the iterative solution of large sparse saddle point problems of the form

$$\mathcal{A}u \equiv \begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \equiv b, \quad (1.1)$$

where $A \in \mathbb{R}^{n \times n}$ is a positive definite matrix, $B \in \mathbb{R}^{m \times n}$ ($m \leq n$) has full row rank, $x, f \in \mathbb{R}^n$ and $y, g \in \mathbb{R}^m$. Under these assumptions we know that the saddle point matrix \mathcal{A} is nonsingular and the linear system (1.1) has a unique solution; see [1] for a general discussion about the nonsingularity of block two-by-two matrices. The saddle point problem (1.1) arises from many scientific computing and engineering applications [2], such as constrained optimization and constrained least-squares problem [3], computational fluid dynamics [4,5], data interpolation [6], element-free Galerkin discretization of elasticity problem [7–9]. The linear system (1.1) is also termed as a *Karsh-Kahn-Tucker* (KKT) system, an augmented system or an equilibrium system.

* Received March 20, 2017 / Revised version received September 25, 2017 / Accepted October 23, 2017 /
Published online August 14, 2018 /

¹⁾ Corresponding author

There exists many efficient methods for solving saddle point problems, such as the null space method, the coupled direct solver, the stationary iterative method, the Krylov subspace method, and so on; see [2] for more details. The Krylov subspace method is one of the most effective methods for solving large sparse systems of linear equations [10]. However, when the Krylov subspace method is applied to solve the saddle point problem (1.1), it often converges very slowly and an efficient preconditioner is needed to achieve rapid convergence. One way to construct preconditioner is by matrix splitting iterative methods. For solving the saddle point problem (1.1), the Uzawa-like iteration methods [2] and the *Hermitian and skew-Hermitian splitting* (HSS)-like iteration methods [11–13] are two classes of efficient iterative methods, which lead to the block diagonal and block triangular preconditioners [14–16] and the HSS-like preconditioners [12, 13], respectively.

Let

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} S & B^T \\ -B & 0 \end{bmatrix} = \mathcal{H} + \mathcal{S}$$

be the Hermitian and skew-Hermitian splitting of the matrix \mathcal{A} , where $H = \frac{1}{2}(A + A^T)$ and $S = \frac{1}{2}(A - A^T)$ are the symmetric and the skew-symmetric parts of the (1,1) block matrix A , respectively. Applying the HSS iteration method proposed in [11], Benzi and Golub [13] proposed the HSS preconditioner

$$\mathcal{P}_{HSS} = \frac{1}{2\alpha} \begin{bmatrix} \alpha I + H & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I + S & B^T \\ -B & \alpha I \end{bmatrix} \quad (1.2)$$

for the saddle point problem (1.1), where α is a real positive parameter and I is the identity matrix of suitable dimension. The HSS preconditioner (1.2) is induced by the stationary HSS iteration method

$$\begin{cases} (\alpha I + \mathcal{H})u^{k+\frac{1}{2}} = (\alpha I - \mathcal{S})u^k + b, \\ (\alpha I + \mathcal{S})u^{k+1} = (\alpha I - \mathcal{H})u^{k+\frac{1}{2}} + b, \end{cases} \quad k = 0, 1, 2, \dots$$

It is noted that Bai et al. [11] first proposed the HSS iteration method for solving non-Hermitian positive definite linear systems and they proved the unconditionally convergent property of this method. Then Benzi and Golub [13] applied the HSS iteration method to the saddle point problem (1.1) and proved that it is also unconditionally convergent for (1.1). There are several variants of the HSS iteration method in recent years; see [17–21]. Spectral properties of the HSS preconditioned matrices as well as the optimal parameters can be found in [22–27].

The HSS iteration method is a two-half steps iteration method. The first step is easy to solve since $\alpha I + \mathcal{H}$ is symmetric positive definite. However, the second step is difficult to solve since the coefficient matrix $\alpha I + \mathcal{S}$ has the same structure as the original saddle point matrix \mathcal{A} and the (1,1) block of the matrix $\alpha I + \mathcal{S}$ is also nonsymmetric. Based on the idea of the *positive-definite and skew-Hermitian splitting* (PSS) iteration method [28], Pan et al. [29] proposed a *deteriorated PSS* (DPSS) preconditioner

$$\tilde{\mathcal{P}}_{DPSS} = \frac{1}{2\alpha} \begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix}$$

for the saddle point problem (1.1). When A is Hermitian, the DPSS preconditioner is the same as the HSS preconditioner. When A is non-Hermitian, the DPSS preconditioner is easier

to be implemented than the HSS preconditioner. Theoretical analyses in [20, 29, 30] indicate that the corresponding DPSS iteration method is unconditionally convergent and the DPSS preconditioned matrix has a clustered eigenvalue distribution. When the matrix $\tilde{\mathcal{P}}_{DPSS}$ is used as a preconditioner, the pre-factor has no effect on the preconditioned system. So we can replace $\frac{1}{2\alpha}$ by $\frac{1}{\alpha}$. Now, the DPSS preconditioner becomes

$$\mathcal{P}_{DPSS} = \frac{1}{\alpha} \begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix} = \begin{bmatrix} \alpha I + A & (I + \frac{1}{\alpha}A)B^T \\ -B & \alpha I \end{bmatrix}. \quad (1.3)$$

As a preconditioner, it is expected to be close to the coefficient matrix as much as possible. From (1.1) and (1.3), we can see that the difference between the saddle point matrix \mathcal{A} and the DPSS preconditioner \mathcal{P}_{DPSS} is given by

$$\mathcal{R}_{DPSS} = \mathcal{P}_{DPSS} - \mathcal{A} = \begin{bmatrix} \alpha I & \frac{1}{\alpha}AB^T \\ 0 & \alpha I \end{bmatrix}.$$

We find that only the (2,1) block of \mathcal{R}_{DPSS} is zero and an ideal parameter α should be chosen to balance the diagonal and the off-diagonal parts. To get a better approximation, by removing the (1,1) block of \mathcal{R}_{DPSS} , Zhang et al. [31] proposed a *relaxed positive-definite and skew-Hermitian splitting* (RPSS) preconditioner

$$\mathcal{P}_{RPSS} = \frac{1}{\alpha} \begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & (I + \alpha A^{-1})B^T \\ -B & \alpha I \end{bmatrix} = \begin{bmatrix} A & (I + \frac{1}{\alpha}A)B^T \\ -B & \alpha I \end{bmatrix} \quad (1.4)$$

for the saddle point problem (1.1). The relaxed technique has also been studied for other HSS-like preconditioners [9, 32, 33]. Numerical results in [31] show that the RPSS preconditioned matrix $\mathcal{P}_{RPSS}^{-1}\mathcal{A}$ has more clustered eigenvalue distribution than the DPSS preconditioned matrix $\mathcal{P}_{DPSS}^{-1}\mathcal{A}$ and the RPSS preconditioned GMRES method needs less iteration steps than the DPSS preconditioned GMRES method. Recently, a generalized RPSS preconditioner is presented for the saddle point problem (1.1) [34]. However, when these RPSS preconditioners are used to accelerate convergence of Krylov subspace methods, it is necessary to solve a linear subsystem with the coefficient matrix like $\alpha I + \frac{1}{\alpha}BB^T + BA^{-1}B^T$ at each iteration. This is expensive due to the existing of the Schur complement matrix.

In this paper, based on ideas of the RPSS preconditioner and the inexact preconditioners studied in [15], we propose an *improved relaxed positive-definite and skew-Hermitian splitting* (IRPSS) preconditioner for the saddle point problem (1.1), which is much easier to be implemented than the RPSS preconditioner. Some properties including the eigenvalue distribution, the eigenvector distribution and the minimal polynomial of the IRPSS preconditioned matrix are studied. By choosing an appropriate preconditioning matrix, we can obtain a more clustered eigenvalue distribution of the IRPSS preconditioned saddle point matrix and a theoretical optimal IRPSS preconditioner. Numerical results show that the IRPSS preconditioners are more effective than the RPSS preconditioner when they are used to accelerate Krylov subspace methods for solving the saddle point problem (1.1).

The outline of this paper is as follows. In Section 2, we propose an IRPSS preconditioner for saddle point problem (1.1). In Section 3, some properties of the IRPSS preconditioned matrix are studied in detail and a theoretical optimal IRPSS preconditioner is obtained. In Section 4, numerical experiments are given to show the effectiveness of the proposed IRPSS preconditioners. Finally, we end this paper with a few concluding remarks in Section 5.

2. The Improved RPSS Preconditioner

When the RPSS preconditioner \mathcal{P}_{RPSS} in (1.4) is used to accelerate Krylov subspace methods such as GMRES for solving the saddle point problem (1.1), the following generalized residual equation needs to be solved at each iteration

$$\mathcal{P}_{RPSS}z = \begin{bmatrix} A & (I + \frac{1}{\alpha}A)B^T \\ -B & \alpha I \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = r, \quad (2.1)$$

where $z_1, r_1 \in \mathbb{R}^n$ and $z_2, r_2 \in \mathbb{R}^m$. According to the procedure in [31], we can first do the matrix factorization for the RPSS preconditioner

$$\mathcal{P}_{RPSS} = \frac{1}{\alpha} \begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{\alpha}B & I \end{bmatrix} \begin{bmatrix} \alpha I & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} I & (\frac{1}{\alpha}I + A^{-1})B^T \\ 0 & I \end{bmatrix},$$

where $C = \alpha I + \frac{1}{\alpha}BB^T + BA^{-1}B^T$. Then we obtain the following algorithm to solve (2.1).

Algorithm 2.1. Let $z = [z_1^T, z_2^T]^T$ and $r = [r_1^T, r_2^T]^T$ be the current and the generalized residual vectors, respectively, with $r_1, z_1 \in \mathbb{R}^n$ and $r_2, z_2 \in \mathbb{R}^m$. We can solve the linear system $\mathcal{P}_{RPSS}z = r$ (2.1) by the following procedure:

- (1) solve $At_1 = r_1$;
- (2) solve $Cz_2 = Bt_1 + r_2$;
- (3) compute $t_2 = B^T z_2$ and solve $A\tilde{z}_1 = t_2$;
- (4) compute $z_1 = t_1 - \frac{1}{\alpha}t_2 - \tilde{z}_1$.

From Algorithm 2.1, we see that there are two linear systems with the same coefficient matrix A and one linear system with the coefficient matrix $C = \alpha I + \frac{1}{\alpha}BB^T + BA^{-1}B^T$ needed to be solved. If the matrix A is symmetric positive definite, then we can use the sparse Cholesky decomposition or the conjugate gradient method to solve the linear system. If the matrix A is nonsymmetric positive definite, then the sparse LU decomposition can be used. However, it is difficult to solve the linear system with the coefficient matrix $C = \alpha I + \frac{1}{\alpha}BB^T + BA^{-1}B^T$. The computation of the coefficient matrix $\alpha I + \frac{1}{\alpha}BB^T + BA^{-1}B^T$ is expensive due to existing of the Schur complement matrix $BA^{-1}B^T$.

In order to effectively implement the RPSS preconditioner, we propose an *improved relaxed positive-definite and skew-Hermitian splitting* (IRPSS) preconditioner as follows

$$\begin{aligned} \mathcal{P}_{IRPSS} &= \frac{1}{\alpha} \begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{\alpha}B & I \end{bmatrix} \begin{bmatrix} \alpha I & 0 \\ 0 & \widehat{C} \end{bmatrix} \begin{bmatrix} I & (\frac{1}{\alpha}I + A^{-1})B^T \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} A & (I + \frac{1}{\alpha}A)B^T \\ -B & \widehat{C} - B(\frac{1}{\alpha}I + A^{-1})B^T \end{bmatrix}, \end{aligned} \quad (2.2)$$

where $\widehat{C} \in \mathbb{R}^{m \times m}$ is a nonsingular matrix. Here, we expect that the matrix \widehat{C} is chosen such that the linear system with the coefficient matrix \widehat{C} is easy to be solved and the IRPSS preconditioned saddle point matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ has more clustered eigenvalue distribution. Similar

to Algorithm 2.1, we obtain the following algorithm to implement the IRPSS preconditioner, which is equivalent to solving a sequence of generalized residual equations

$$\mathcal{P}_{IRPSS}z = \begin{bmatrix} A & (I + \frac{1}{\alpha}A)B^T \\ -B & \widehat{C} - B(\frac{1}{\alpha}I + A^{-1})B^T \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = r, \quad (2.3)$$

where $z_1, r_1 \in \mathbb{R}^n$ and $z_2, r_2 \in \mathbb{R}^m$.

Algorithm 2.2. Let $z = [z_1^T, z_2^T]^T$ and $r = [r_1^T, r_2^T]^T$ be the current and the generalized residual vectors, respectively, with $r_1, z_1 \in \mathbb{R}^n$ and $r_2, z_2 \in \mathbb{R}^m$. We can solve the linear system $\mathcal{P}_{IRPSS}z = r$ in (2.3) by the following procedure:

- (1) solve $At_1 = r_1$;
- (2) solve $\widehat{C}z_2 = Bt_1 + r_2$;
- (3) compute $t_2 = B^T z_2$ and solve $A\tilde{z}_1 = t_2$;
- (4) compute $z_1 = t_1 - \frac{1}{\alpha}t_2 - \tilde{z}_1$.

Comparing Algorithm 2.2 with Algorithm 2.1, we find that only the second step is different. Note that the matrix \widehat{C} here is not an approximation to the matrix $C = \alpha I + \frac{1}{\alpha}BB^T + BA^{-1}B^T$. We can choose an appropriate matrix \widehat{C} such that the IRPSS preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ has clustered eigenvalue distribution and at the same time the IRPSS preconditioner is easy to be implemented. For example, let \widehat{C} be an approximate matrix of $BA^{-1}B^T$ according to theoretical analysis in Section 3. Hence, the IRPSS preconditioner is not an inexact variant of the RPSS preconditioner.

3. Properties of the IRPSS Preconditioned Matrix

In this section, we derive some properties of the IRPSS preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$. The following theorem describes the eigenvalue distribution of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$.

Theorem 3.1. Let $A \in \mathbb{R}^{n \times n}$ be a positive definite matrix, $B \in \mathbb{R}^{m \times n}$ have full row rank and α be a positive constant. Let the IRPSS preconditioner \mathcal{P}_{IRPSS} be defined as in (2.2). Then the IRPSS preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ has an eigenvalue 1 with multiplicity n , and the remaining m eigenvalues are the eigenvalues of the matrix $\widehat{C}^{-1}BA^{-1}B^T$. Furthermore, if $\widehat{C} = BA^{-1}B^T$, then all eigenvalues of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ are 1.

Proof. By the matrix factorization of the IRPSS preconditioner in (2.2), we get

$$\mathcal{P}_{IRPSS}^{-1} = \begin{bmatrix} A^{-1} - (\frac{1}{\alpha}I + A^{-1})B^T\widehat{C}^{-1}BA^{-1} & -(\frac{1}{\alpha}I + A^{-1})B^T\widehat{C}^{-1} \\ \widehat{C}^{-1}BA^{-1} & \widehat{C}^{-1} \end{bmatrix}.$$

Then it holds that

$$\mathcal{P}_{IRPSS}^{-1}\mathcal{A} = \begin{bmatrix} I & A^{-1}B^T - (\frac{1}{\alpha}I + A^{-1})B^T\widehat{C}^{-1}BA^{-1}B^T \\ 0 & \widehat{C}^{-1}BA^{-1}B^T \end{bmatrix}. \quad (3.1)$$

From (3.1) we immediately obtain that the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ has an eigenvalue 1 with multiplicity n , and the remaining m eigenvalues are the eigenvalues of the matrix $\widehat{C}^{-1}BA^{-1}B^T$.

Furthermore, if $\widehat{C} = BA^{-1}B^T$, then the IRPSS preconditioned matrix has the following structure

$$\mathcal{P}_{IRPSS}^{-1}\mathcal{A} = \begin{bmatrix} I & -\frac{1}{\alpha}B^T \\ 0 & I \end{bmatrix}, \quad (3.2)$$

which implies that all eigenvalues of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ are 1. \square

Remark 3.1. Theorem 3.1 presents an ideal eigenvalue distribution when $\widehat{C} = BA^{-1}B^T$. However, it is not practical in actual computation. In this remark, we give further results about the remaining m eigenvalues of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ for a general symmetric positive definite matrix \widehat{C} .

- If $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, then the Schur complement matrix $BA^{-1}B^T \in \mathbb{R}^{m \times m}$ is also symmetric positive definite. Assume that the smallest and the largest eigenvalues of $BA^{-1}B^T$ are ξ_1 and ξ_m , respectively. Let $\widehat{C} \in \mathbb{R}^{m \times m}$ be symmetric positive definite and its smallest and largest eigenvalues are μ_1 and μ_m , respectively. Then the remaining m eigenvalues of the preconditioned matrix are located in the positive real interval

$$\left[\frac{\xi_1}{\mu_m}, \frac{\xi_m}{\mu_1} \right].$$

- If $A \in \mathbb{R}^{n \times n}$ is nonsymmetric positive definite. Further assume that all eigenvalues of the Schur complement matrix $BA^{-1}B^T$ are enclosed in the rectangle $[\xi_1, \xi_m] \times [-\eta, \eta]$. Let $\widehat{C} \in \mathbb{R}^{m \times m}$ be symmetric positive definite and its smallest and largest eigenvalues are μ_1 and μ_m , respectively. Then the remaining m eigenvalues of the preconditioned matrix are located in

$$\left[\frac{\xi_1}{\mu_m}, \frac{\xi_m}{\mu_1} \right] \times \left[-\frac{\eta}{\mu_1}, \frac{\eta}{\mu_1} \right].$$

From the block structure of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$, we can obtain the minimal polynomial of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$, which is presented in the following theorem.

Theorem 3.2. *Let the IRPSS preconditioner \mathcal{P}_{IRPSS} be defined as in (2.2). Then both the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ and the dimension of the Krylov subspace $\mathcal{K}(\mathcal{P}_{IRPSS}^{-1}\mathcal{A}, b)$ are at most $m + 1$. Furthermore, if $\widehat{C} = BA^{-1}B^T$, then the minimal polynomial of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ is $(\lambda - 1)^2$.*

Proof. Let $\Theta_1 = \widehat{C}^{-1}BA^{-1}B^T$ and $\Theta_2 = A^{-1}B^T - (\frac{1}{\alpha}I + A^{-1})B^T\widehat{C}^{-1}BA^{-1}B^T$. Then from (3.1) the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ can be rewritten as

$$\mathcal{P}_{IRPSS}^{-1}\mathcal{A} = \begin{bmatrix} I & \Theta_2 \\ 0 & \Theta_1 \end{bmatrix},$$

which is a block upper triangular matrix with the (1,1) block being an identity matrix.

Let λ_i ($i = 1, \dots, m$) be the eigenvalues of Θ_1 . The characteristic polynomial of the matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ is

$$\Phi(\lambda) = \det(\mathcal{P}_{IRPSS}^{-1}\mathcal{A} - \lambda I) = (-1)^{m+n}(\lambda - 1)^n \prod_{i=1}^m (\lambda - \lambda_i).$$

By expanding the $(m + 1)$ -degree polynomial $(\lambda - 1) \prod_{i=1}^m (\lambda - \lambda_i)$, we have

$$(\mathcal{P}_{IRPSS}^{-1}\mathcal{A} - I) \prod_{i=1}^m (\mathcal{P}_{IRPSS}^{-1}\mathcal{A} - \lambda_i I) = \begin{bmatrix} 0 & \Theta_2 \prod_{i=1}^m (\Theta_1 - \lambda_i I) \\ 0 & (\Theta_1 - I) \prod_{i=1}^m (\Theta_1 - \lambda_i I) \end{bmatrix}.$$

Since λ_i ($i = 1 \cdots m$) are the eigenvalues of $\Theta_1 \in \mathbb{R}^{m \times m}$, by Hamilton-Cayley theorem the equality $\prod_{i=1}^m (\Theta_1 - \lambda_i I) = 0$ holds. Therefore, the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ is at most $m + 1$. From [35, Proposition 6.1], we know that the dimension of the Krylov subspace $\mathcal{K}(\mathcal{P}_{IRPSS}^{-1}\mathcal{A}, b)$ is equal to the degree of the minimal polynomial of the corresponding preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$. So, the dimension of the Krylov subspace $\mathcal{K}(\mathcal{P}_{IRPSS}^{-1}\mathcal{A}, b)$ is at most $m + 1$, too.

Moreover, if $\widehat{C} = B A^{-1} B^T$, then it follows from (3.2) that the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ is a block upper triangular matrix with block diagonal matrices being the identity matrix. Therefore, we obtain the minimal polynomial of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$. \square

From Theorems 3.1 and 3.2, we find that the theoretical optimal IRPSS preconditioner is the one when $\widehat{C} = B A^{-1} B^T$. For this case, we denote it by

$$\begin{aligned} \mathcal{P}_{OIRPSS} &= \frac{1}{\alpha} \begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{\alpha} B & I \end{bmatrix} \begin{bmatrix} \alpha I & 0 \\ 0 & B A^{-1} B^T \end{bmatrix} \begin{bmatrix} I & (\frac{1}{\alpha} I + A^{-1}) B^T \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} A & B^T + \frac{1}{\alpha} A B^T \\ -B & -\frac{1}{\alpha} B B^T \end{bmatrix}. \end{aligned} \quad (3.3)$$

As discussed in Remark 3.1, the optimal IRPSS preconditioner is difficult to be implemented although nice properties of the preconditioned matrix $\mathcal{P}_{OIRPSS}^{-1}\mathcal{A}$ are shown in Theorems 3.1 and 3.2. Therefore, in actual computation, we should choose the preconditioning matrix \widehat{C} which is an approximation to the Schur complement matrix $B A^{-1} B^T$. Numerical behavior of two choices of the preconditioning matrix \widehat{C} will be shown in Section 4.

The termination of a Krylov subspace method is not only related to the location of the eigenvalues, but also to the number of corresponding linearly independent eigenvectors. The following theorem describes the eigenvector distribution of the IRPSS preconditioned matrix.

Theorem 3.3. *Let the IRPSS preconditioner \mathcal{P}_{IRPSS} be defined as in (2.2). Then the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ has $n + i$ ($0 \leq i \leq m$) linearly independent eigenvectors. There are*

- n eigenvectors $\begin{bmatrix} u_p^1 \\ 0 \end{bmatrix}$ ($p = 1, 2, \dots, n$) that correspond to the eigenvalue 1, where u_p^1 ($p = 1, 2, \dots, n$) are arbitrary linearly independent vectors.
- i ($0 \leq i \leq m$) eigenvectors $\begin{bmatrix} u_p^2 \\ v_p^2 \end{bmatrix}$ ($0 \leq p \leq i$) that correspond to nonunit eigenvalues, where $v_p^2 \neq 0$, $B A^{-1} B^T v_p^2 = \lambda \widehat{C} v_p^2$ and $u_p^2 = [\frac{\lambda}{\alpha(1-\lambda)} I - A^{-1}] B^T v_p^2$.

Proof. Let λ be an eigenvalue of the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ and $\begin{bmatrix} u \\ v \end{bmatrix}$ be the corresponding eigenvector. From (3.1) we get the following eigenvalue problem

$$\begin{bmatrix} I & A^{-1}B^T - (\frac{1}{\alpha}I + A^{-1})B^T\widehat{C}^{-1}BA^{-1}B^T \\ 0 & \widehat{C}^{-1}BA^{-1}B^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix},$$

which is equal to

$$\begin{cases} (1 - \lambda)u = (\frac{1}{\alpha}I + A^{-1})B^T\widehat{C}^{-1}BA^{-1}B^Tv - A^{-1}B^Tv, \\ BA^{-1}B^Tv = \lambda\widehat{C}v. \end{cases} \quad (3.4)$$

Substituting the second equation into the first equation in (3.4), we obtain that

$$(1 - \lambda)u = (\lambda - 1)A^{-1}B^Tv + \frac{\lambda}{\alpha}B^Tv. \quad (3.5)$$

If $\lambda = 1$, then from (3.5) we have

$$B^Tv = 0.$$

Since B has full row rank, v is a zero vector for this case. Therefore, there are n linearly independent eigenvectors $\begin{bmatrix} u_p^1 \\ 0 \end{bmatrix}$ ($p = 1, 2, \dots, n$) that correspond to the eigenvalue 1, where u_p^1 ($p = 1, 2, \dots, n$) are arbitrary linearly independent vectors.

If $\lambda \neq 1$, then from (3.5) we have

$$u = \left[\frac{\lambda}{\alpha(1-\lambda)}I - A^{-1} \right] B^Tv.$$

It must be $v \neq 0$. Otherwise, $v = 0$ implies that $u = 0$, which contradicts with $\begin{bmatrix} u \\ v \end{bmatrix}$ being an eigenvector. When there exists any $v \neq 0$ which satisfies the second equation in (3.4), there will be i ($0 \leq i \leq m$) eigenvectors $\begin{bmatrix} u_p^2 \\ v_p^2 \end{bmatrix}$ ($0 \leq p \leq i$) that correspond to nonunit eigenvalues, where $v_p^2 \neq 0$ are the eigenvectors of the generalized eigenvalue problem of the second equation in (3.4) and $u_p^2 = [\frac{\lambda}{\alpha(1-\lambda)}I - A^{-1}]B^Tv_p^2$.

Now, we show that the $n+i$ eigenvectors are linearly independent. Let $a^1 = [a_1^1 \ \dots \ a_n^1]^T$ and $a^2 = [a_1^2 \ \dots \ a_i^2]^T$ be two vectors. Then we need to show that

$$\begin{bmatrix} u_1^1 & \dots & u_n^1 \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} a_1^1 \\ \vdots \\ a_n^1 \end{bmatrix} + \begin{bmatrix} u_1^2 & \dots & u_i^2 \\ v_1^2 & \dots & v_i^2 \end{bmatrix} \begin{bmatrix} a_1^2 \\ \vdots \\ a_i^2 \end{bmatrix} = 0 \quad (3.6)$$

holds true if and only if the vectors a^k ($k = 1, 2$) are zero vectors. In (3.6), the first matrix consists of the eigenvectors corresponding to the eigenvalue 1 and the second matrix consists of the eigenvectors corresponding to the nonunit eigenvalues. By multiplying $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ on both sides of (3.6), we have

$$\begin{bmatrix} u_1^1 & \dots & u_n^1 \\ 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} a_1^1 \\ \vdots \\ a_n^1 \end{bmatrix} + \begin{bmatrix} u_1^2 & \dots & u_i^2 \\ v_1^2 & \dots & v_i^2 \end{bmatrix} \begin{bmatrix} \lambda_1 a_1^2 \\ \vdots \\ \lambda_i a_i^2 \end{bmatrix} = 0. \quad (3.7)$$

Subtracting (3.6) from (3.7), we obtain

$$\begin{bmatrix} u_1^2 & \cdots & u_i^2 \\ v_1^2 & \cdots & v_i^2 \end{bmatrix} \begin{bmatrix} (\lambda_1 - 1)a_1^2 \\ \vdots \\ (\lambda_i - 1)a_i^2 \end{bmatrix} = 0.$$

Since $\lambda_i \neq 1$ and $\begin{bmatrix} u_p^2 \\ v_p^2 \end{bmatrix}$ ($p = 1, \dots, i$) are linearly independent, we know that $a_p^2 = 0$ ($p = 1, \dots, i$). As u_p^1 ($p = 1, \dots, n$) are linearly independent, from (3.7) we have $a_p^1 = 0$ ($p = 1, \dots, n$). Therefore, the preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ has $n + i$ ($0 \leq i \leq m$) linearly independent eigenvectors. \square

Since the optimal IRPSS preconditioner (3.3) is a special case of the IRPSS preconditioner, we directly obtain the following Corollary for the eigenvector distribution of the optimal IRPSS preconditioned matrix $\mathcal{P}_{OIRPSS}^{-1}\mathcal{A}$.

Corollary 3.1. *Let the optimal IRPSS preconditioner \mathcal{P}_{OIRPSS} be defined as in (3.3). Then the preconditioned matrix $\mathcal{P}_{OIRPSS}^{-1}\mathcal{A}$ has n linearly independent eigenvectors $\begin{bmatrix} u_p^1 \\ 0 \end{bmatrix}$ ($p = 1, 2, \dots, n$), where u_p^1 ($p = 1, 2, \dots, n$) are arbitrary linearly independent vectors.*

4. Numerical Experiments

In this section, we use three test problems to verify the feasibility and effectiveness of the IRPSS preconditioners for the saddle point problem (1.1). The first and the second test problems are standard saddle point problems with the (1,1) block matrix A being symmetric positive definite. The third test problem is a nonsymmetric saddle point problem with the (1,1) block matrix A being nonsymmetric positive definite. We compare the IRPSS preconditioners with the DPSS preconditioner in (1.3), the RPSS preconditioner in (1.4) and the RDPSS preconditioner in [33] from aspects of numbers of total iteration steps (denoted by ‘IT’) and elapsed CPU times in seconds (denoted by ‘CPU’). In actual computations, we use the left preconditioned GMRES method. Numerical results of the GMRES method without preconditioning are also listed, which show advantages of the preconditioning techniques.

In our implementations, the initial guess is chosen to be the zero vector and the iteration is terminated if the current iteration satisfies

$$ERR = \frac{\|b - Au^k\|_2}{\|b\|_2} \leq 10^{-6}.$$

To accelerate the convergence rate of GMRES, we can choose the IRPSS preconditioner according to the preconditioning matrix \widehat{C} . Besides the optimal choice $\widehat{C} = BA^{-1}B^T$, we use other two choices of \widehat{C} which are $\widehat{C} = \frac{1}{\alpha}BB^T$ and $\widehat{C} = \frac{1}{\alpha}B\widehat{A}^{-1}B^T$ with $\widehat{A} = \text{diag}(A)$. The corresponding IRPSS preconditioners are denoted by

$$\mathcal{P}_{IRPSS_1} = \frac{1}{\alpha} \begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{\alpha}B & I \end{bmatrix} \begin{bmatrix} \alpha I & 0 \\ 0 & \frac{1}{\alpha}BB^T \end{bmatrix} \begin{bmatrix} I & (\frac{1}{\alpha}I + A^{-1})B^T \\ 0 & I \end{bmatrix}, \quad (4.1)$$

$$\mathcal{P}_{IRPSS_2} = \frac{1}{\alpha} \begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{\alpha}B & I \end{bmatrix} \begin{bmatrix} \alpha I & 0 \\ 0 & \frac{1}{\alpha}B\widehat{A}^{-1}B^T \end{bmatrix} \begin{bmatrix} I & (\frac{1}{\alpha}I + A^{-1})B^T \\ 0 & I \end{bmatrix}. \quad (4.2)$$

In all these preconditioners, the parameter α is involved. To implement these preconditioners efficiently, we need to choose the parameter α appropriately. By making use of the techniques studied in [36], the following estimates

$$\alpha_{DPSS} = \sqrt{\frac{\|A\|_F \cdot \|B\|_F}{\|I_n\|_F + \|I_m\|_F}} \quad \text{and} \quad \alpha_{RPSS} = \sqrt{\frac{\|A\|_F \cdot \|B\|_F}{\|I_m\|_F}}$$

are used for the DPSS preconditioner and the RPSS preconditioner, respectively. Here, $\|\cdot\|_F$ denotes the Frobenius norm of the corresponding matrix, and I_n and I_m denote the $n \times n$ and the $m \times m$ identity matrices, respectively. For the RDPSS preconditioner, when the (1,1) block matrix A is symmetric positive definite, the RDPSS preconditioner reduces to the relaxed HSS preconditioner and the parameter α is chosen by [9, Theorem 3.1]; when the (1,1) block matrix A is nonsymmetric positive definite, the parameter α is chosen by [33, Theorem 2.2]. Both of them are theoretically optimal for the RDPSS preconditioner. Since the parameter α does not influence the minimal polynomial of the optimal IRPSS preconditioned matrix $\mathcal{P}_{OIRPSS}^{-1}\mathcal{A}$, we always choose $\alpha = 1$ for the optimal IRPSS preconditioner. From Remark 3.1, we choose

$$\alpha_{IRPSS} = \frac{\xi_1}{\mu_m}$$

for the IRPSS preconditioners (4.1) and (4.2). For this case, all eigenvalues of the IRPSS preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ are located in $(0, 1]$ for the first and the second examples and the real part of the eigenvalues of the IRPSS preconditioned matrix $\mathcal{P}_{IRPSS}^{-1}\mathcal{A}$ are located in $(0, 1]$ for the third example. In addition, the linear sub-systems with the symmetric positive definite coefficient matrices are solved by sparse Cholesky factorizations and the linear sub-systems with the nonsymmetric positive definite coefficient matrices are solved by sparse LU factorizations. All codes are run in MATLAB (version R2010b) in double precision and all experiments are performed on an Intel Core (4G RAM) Windows 7 system.

Example 4.1. ([12, 16, 17]) Consider the saddle point problem (1.1), in which

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

where

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

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

This example can be regarded as the discretized linear system of the Stokes equation with the upwind difference scheme [16]. For this example, we have $n = 2q^2$ and $m = q^2$. Thus, the total number of variables of the corresponding saddle point problem is $n + m = 3q^2$. In the test problems, we choose four grids with $q = 8, 16, 32, 64$.

In Table 4.1, we list the parameters α for the DPSS preconditioner, the RPSS preconditioner, the RDPSS preconditioner and the IRPSS preconditioners. In Table 4.2, we show numerical results of the preconditioned GMRES methods with no preconditioner (denoted by ‘‘I’’), the DPSS preconditioner, the RPSS preconditioner, the RDPSS preconditioner, the OIRPSS preconditioner, the IRPSS₁ preconditioner in (4.1) and the IRPSS₂ preconditioner in (4.2). From Table 4.2, we see that all these preconditioners can accelerate the convergence rate of the GMRES method. If preconditioning technique is not applied, the GMRES method converges very

Table 4.1: The parameter α for Example 4.1.

Grids	8×8	16×16	32×32	64×64
DPSS	1.7092e+2	6.3469e+2	2.4412e+3	9.5699e+3
RPSS	2.6557e+2	9.8617e+2	3.7930e+3	1.4869e+4
RDPSS	45.3643	49.2549	51.1942	52.1320
IRPSS ₁	5.5167	5.2345	5.0868	5.0114
IRPSS ₂	0.0170	0.0045	0.0012	0.0003

Table 4.2: Numerical results of the preconditioned GMRES method for Example 4.1.

P-GMRES		Grids			
		8×8	16×16	32×32	64×64
I	IT	54	119	233	501
	CPU	0.0450	0.1882	2.1357	26.9776
	ERR	8.7414e-7	8.5548e-7	9.0958e-7	9.6239e-7
DPSS	IT	32	62	115	240
	CPU	0.0193	0.0654	0.4026	5.9913
	ERR	8.7332e-7	8.6804e-7	8.9561e-7	9.4197e-7
RPSS	IT	9	9	10	10
	CPU	0.0244	0.0601	0.8826	16.6941
	ERR	1.7442e-7	7.0233e-7	8.1376e-7	6.8020e-7
RDPSS	IT	14	19	28	41
	CPU	0.0134	0.0257	0.1548	0.9057
	ERR	3.9359e-7	8.8039e-7	8.6786e-7	9.4785e-7
OIRPSS	IT	3	3	3	3
	CPU	0.0125	0.0514	0.7107	13.7626
	ERR	7.1259e-13	4.4518e-12	4.9479e-11	3.8931e-10
IRPSS ₁	IT	16	25	40	63
	CPU	0.0130	0.0201	0.1163	0.9046
	ERR	6.8725e-7	7.1917e-7	6.7647e-7	7.5079e-7
IRPSS ₂	IT	23	39	67	116
	CPU	0.0167	0.0494	0.2061	2.1301
	ERR	3.7089e-7	9.3355e-7	7.6667e-7	8.0765e-7

slow. From the iteration steps of these preconditioned GMRES methods, the OIRPSS preconditioner is the best one and shows h -independent convergence property which is consistent with the theoretical results in Theorem 3.2. However, the elapsed CPU times of the OIRPSS preconditioned GMRES method are much more than those of the IRPSS and the RDPSS preconditioned GMRES methods. The IRPSS and the RDPSS preconditioned GMRES methods cost much less CPU times than the DPSS, the RPSS and the OIRPSS preconditioned GMRES methods. Moreover, for the IRPSS₁ and IRPSS₂ preconditioned GMRES methods, the former needs less iteration steps and CPU times. Therefore, the IRPSS₁ preconditioner is the best one among these preconditioners in accelerating the GMRES method for Example 4.1.

Figure 4.1 describes the residual curves for Example 4.1 with 32×32 grids. From this figure, we see that the convergence rate of the GMRES method is very slow. Among these preconditioned GMRES methods, the IRPSS preconditioned ones are more effective than the

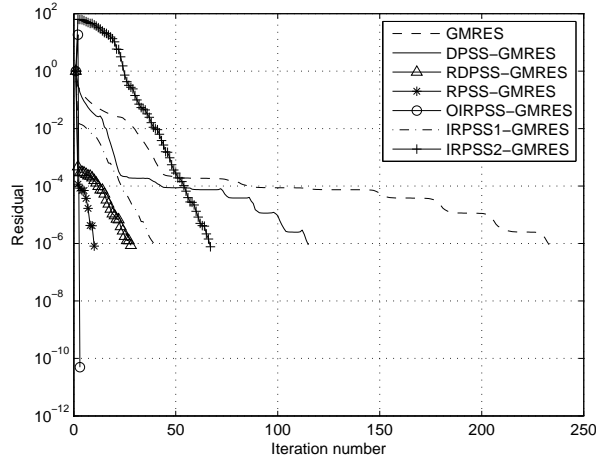


Fig. 4.1. Residual curves of preconditioned GMRES methods for Example 4.1 (32×32 grids).

existing DPSS and RPSS preconditioned ones and the residuals of these preconditioned GMRES methods decrease sharply.

Example 4.2. The second example arises from the meshfree discretization of the elasticity mechanics cantilever beam problem [6–8]. Consider a beam of length $L = 48m$ and height $H = 12m$ subjected to a parabolic traction of force $P = 1000kN$ at the free right end. The Young’s modulus and the Poisson’s ratio of the beam are $E = 3.0 \times 10^7 kPa$ and $\nu = 0.3$, respectively. The element-free Galerkin method is used to discretize this problem and the essential boundary conditions are enforced by the augmented Lagrangian approach [8]. For detail of the discretization method, see [7–9].

For this example, four regular node distributions, i.e., 37×9 , 49×19 , 97×25 and 121×31 , are used to discretize the problem domain. For these four node distributions, we use 10×4 , 20×8 , 40×12 and 50×15 background cells to obtain integrals, respectively. The orders of saddle point matrix \mathcal{A} are listed in Table 4.3.

Table 4.3: The orders of saddle point matrix \mathcal{A} for Example 4.2.

Nodes	Background Cells	n	m	$n + m$
49×19	20×8	1862	38	1900
97×25	40×12	4850	50	4900
121×31	50×15	7502	62	7564
145×37	60×18	10730	74	10804

Table 4.4: The parameter α for Example 4.2.

Nodes	49×19	97×25	121×31	145×37
DPSS	4.7794	4.1285	3.8964	3.7183
RPSS	13.5183	13.5983	13.4974	13.4281
RDPSS	1.0050	0.8364	0.8104	0.7967
IRPSS ₁	0.0253	0.0174	0.0167	0.0164
IRPSS ₂	0.0019	0.0013	0.0012	0.0012

Table 4.5: Numerical results of the preconditioned GMRES method for Example 4.2.

P-GMRES		Nodes			
		49×19	97×25	121×31	145×37
I	IT	601	799	918	1019
	CPU	5.7131	29.2376	52.9165	115.0041
	ERR	9.9511e-7	9.9085e-7	9.9369e-7	9.9653e-7
DPSS	IT	246	310	345	373
	CPU	2.9570	13.4543	30.1783	52.4869
	ERR	9.9345e-7	9.7324e-7	9.9713e-7	9.9335e-7
RPSS	IT	18	19	17	15
	CPU	0.4916	1.8247	6.8323	10.6932
	ERR	9.5116e-7	8.1598e-7	9.6517e-7	9.9575e-7
RDPSS	IT	20	24	26	28
	CPU	0.5068	1.2874	3.5162	6.8649
	ERR	7.3480e-7	8.8039e-7	6.9706e-7	7.7798e-7
OIRPSS	IT	3	3	3	3
	CPU	0.2514	1.2251	4.9205	9.9664
	ERR	2.7069e-14	1.1492e-13	3.8259e-14	3.4558e-13
IRPSS ₁	IT	23	27	30	32
	CPU	0.2350	0.8361	3.3936	6.8385
	ERR	7.8618e-7	9.1518e-7	6.6395e-7	7.2202e-7
IRPSS ₂	IT	27	31	35	37
	CPU	0.2528	1.0878	3.5824	7.7637
	ERR	5.9441e-7	7.5487e-7	5.1603e-7	6.2919e-7

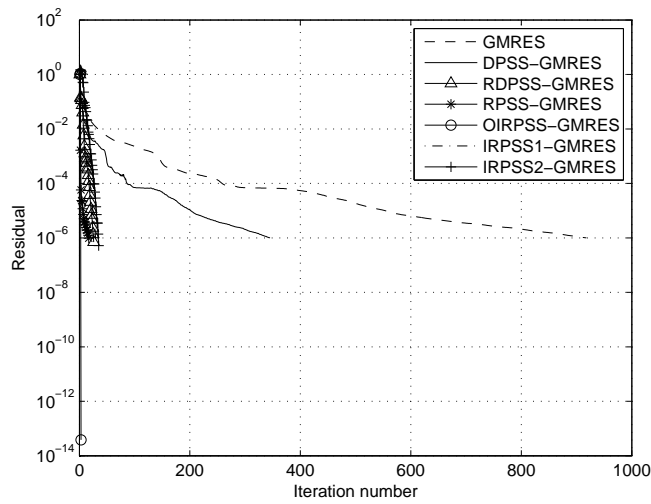


Fig. 4.2. Residual curves of preconditioned GMRES methods for Example 4.2 (121×31 nodes).

In Table 4.4, we list the parameters α for the DPSS, the RPSS, the RDPSS, the IRPSS preconditioners for Example 4.2. We list numerical results of the preconditioned GMRES methods in Table 5 and describe residual curves of preconditioned GMRES methods with 121×31 nodes in Figure 4.2 for Example 4.2. From Table 4.5 and Figure 4.2, we can see that GMRES method

converges very slow while the preconditioned GMRES methods converge very fast. The iteration steps of the OIRPSS preconditioned GMRES method are the same for different problem sizes, which further confirms the theoretical results. Both the iteration steps and the elapsed CPU times show that the DPSS preconditioner is less efficient than the other preconditioners for this example. Although the iteration steps of the RPSS preconditioned GMRES method are a little less than those of the RDPSS, the IRPSS₁ and the IRPSS₂ preconditioned GMRES methods, the elapsed CPU times of it are much more than those of the RDPSS, the IRPSS₁ and the IRPSS₂ preconditioned GMRES methods. And our proposed IRPSS₁ preconditioner is the most effective for Example (4.2).

Example 4.3. Consider the linearization of the steady-state Navier-Stokes equation, i.e., the Oseen problems of the following form

$$\begin{cases} -\nu\Delta u + \omega \cdot \nabla u + \nabla p = f, \\ \nabla \cdot u = 0, \end{cases} \quad \text{in } \Omega$$

where Ω is a bounded domain, $\nu > 0$ is the viscosity, ω is the velocity field obtained from the previous Picard step, the vector field u stands for the velocity, and p represents the pressure.

Table 4.6: The parameter α for Example 4.3.

Grids	8×8	16×16	32×32	64×64
DPSS	3.0851	3.3008	3.4103	3.4657
RPSS	5.2142	5.4633	5.5883	5.6511
RDPSS	0.7160	0.1909	0.0488	0.0123
IRPSS ₁	2.0587e-2	1.8299e-3	1.7196e-4	1.7415e-5
IRPSS ₂	4.6229e-3	4.1248e-4	3.8867e-5	3.9443e-6

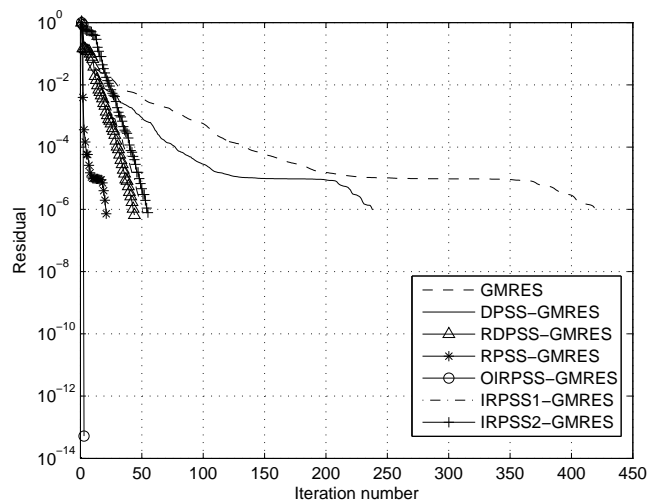
The test problem is the two dimensional ‘‘leaky’’ lid-driven cavity problem in a square domain ($0 \leq x \leq 1, 0 \leq y \leq 1$). The boundary conditions are $u_x = u_y = 0$ on the three fixed walls ($x = 0, y = 0, x = 1$), and $u_x = 1, u_y = 0$ on the moving wall ($y = 1$). We discretize the Oseen equation by Q2-P1 finite element method on the uniform grids with $q = 8, 16, 32, 64$. To generate the discretizations, the IFISS software package [4] developed by Elman et al. is used. For this example, we have $n = 2(q + 1)^2$ and $m = \frac{3}{4}q^2$.

For Example 4.3, we list the parameters α for the DPSS, the RPSS, the RDPSS, the IRPSS preconditioners in Table 4.6. In Table 4.7, we list numerical results of the preconditioned GMRES methods. From Table 4.7, we can see that the OIRPSS preconditioned GMRES method needs the least iteration steps which keep invariant with respect to grids, but it costs the most CPU time among these preconditioned GMRES methods except the RPSS preconditioned GMRES method. However, the IRPSS₁ and the IRPSS₂ preconditioned GMRES methods cost much less CPU times than the DPSS, the RPSS, the RDPSS and the OIRPSS preconditioned GMRES methods, and their iteration steps are also less than those of the DPSS preconditioned GMRES method. Therefore, the IRPSS₁ and the IRPSS₂ preconditioners are effective in accelerating the GMRES method for Example 4.3.

Figure 4.3 plots the residual curves of the preconditioned GMRES methods for Example 4.3 with 32×32 grids. From this figure, we see that the residuals of the GMRES method and the DPSS preconditioned GMRES method decrease very slow, while the residuals the RPSS, the RDPSS and the IRPSS preconditioned GMRES methods decrease very sharply.

Table 4.7: Numerical results of the preconditioned GMRES method for Example 4.3.

P-GMRES		Grids			
		8×8	16×16	32×32	64×64
I	IT	93	201	420	550
	CPU	0.0820	0.4604	6.1480	26.2488
	ERR	$8.7532e-7$	$9.5649e-7$	$9.1073e-7$	$9.9721e-7$
DPSS	IT	57	117	238	309
	CPU	0.0436	0.1352	1.6616	10.7101
	ERR	$9.4452e-7$	$9.2778e-7$	$9.5966e-7$	$9.9935e-7$
RPSS	IT	19	20	21	12
	CPU	0.0377	0.1676	2.1948	48.3061
	ERR	$5.6075e-7$	$7.6293e-7$	$7.1870e-7$	$9.1268e-7$
RDPSS	IT	17	28	44	67
	CPU	0.0177	0.0486	0.3317	2.4769
	ERR	$6.0654e-7$	$8.5818e-7$	$6.5853e-7$	$9.6179e-7$
OIRPSS	IT	3	3	3	3
	CPU	0.0151	0.1543	2.1221	45.0370
	ERR	$4.1499e-15$	$4.0477e-15$	$5.1984e-14$	$5.3835e-14$
IRPSS ₁	IT	19	33	51	81
	CPU	0.0139	0.0513	0.1776	1.4417
	ERR	$7.5302e-7$	$2.7659e-7$	$7.4276e-8$	$8.2083e-7$
IRPSS ₂	IT	20	34	55	88
	CPU	0.0169	0.0419	0.1862	1.5417
	ERR	$7.0012e-7$	$4.5659e-7$	$7.8723e-7$	$9.1930e-7$

Fig. 4.3. Residual curves of preconditioned GMRES methods for Example 4.3 (32×32 grids).

Combining the numerical results in Table 4.7, we arrive at the conclusion that our proposed IRPSS preconditioners are effective for the saddle point problem (1.1) and improve the existing DPSS, RPSS, RDPSS preconditioners greatly.

5. Conclusions

In this paper, we have proposed an *improved relaxed positive-definite and skew-Hermitian splitting* (IRPSS) preconditioner for the saddle point problem (1.1), which is essentially based on the *positive-definite and skew-Hermitian splitting* (PSS) of the coefficient matrix and the relaxed splitting techniques. Theoretical analyses show that the IRPSS preconditioned saddle point matrix has a clustered eigenvalue distribution. An upper bound about the degree of the minimal polynomial of the IRPSS preconditioned saddle point matrix shows the finite-steps termination merit of the IRPSS preconditioned GMRES method. According to theoretical results about the IRPSS preconditioned matrix, an optimal IRPSS preconditioner is obtained. Numerical examples arising from the Navier-Stokes equation and the elasticity mechanics equation confirm the correctness of theoretical results and the effectiveness of the proposed preconditioners.

However, in the IRPSS preconditioner, only an approximation \widehat{C} to the Schur complement matrix $BA^{-1}B^T$ is considered in this paper. In fact, the linear system with the coefficient matrix A is likely to be too costly, especially when solving problems arising from the discretization of 3D partial differential equations. So how to give an approximation to the matrix A should be further studied in future work.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (Nos. 11771225, 11301521, 11771467, 11572210).

References

- [1] Z.J. Bai and Z.Z. Bai, On nonsingularity of block two-by-two matrices, *Linear Algebra Appl.*, **439** (2013), 2388-2404.
- [2] M. Benzi, G.H. Golub and J. Liesen, Numerical solution of saddle point problems, *Acta Numer.*, **14** (2005), 1-137.
- [3] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [4] 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, 2014.
- [5] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
- [6] Y. Cao, L.Q. Yao and S.C. Yi, A weighted nodal-radial point interpolation meshless method for 2D solid problems, *Eng. Anal. Boundary Elem.*, **39** (2014), 88-100.
- [7] T. Belytschko, Y.Y. Lu and L. Gu, Element-free Galerkin methods, *Int. J. Numer. Meth. Engrg.*, **37** (1994), 229-256.
- [8] G. Ventura, An augmented Lagrangian approach to essential boundary conditions in meshless methods, *Int. J. Numer. Meth. Engrg.*, **53** (2002), 825-842.
- [9] 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** (2013), 398-421.
- [10] Z.Z. Bai, Motivations and realizations of Krylov subspace methods for large sparse linear systems, *J. Comput. Appl. Math.*, **283** (2015), 71-78.
- [11] 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** (2003), 603-626.
- [12] Z.Z. Bai and G.H. Golub, Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems, *IMA J. Numer. Anal.*, **27** (2007), 1-23.
- [13] M. Benzi and G.H. Golub, A preconditioner for generalized saddle point problems, *SIAM J. Matrix Anal. Appl.*, **26** (2004), 20-41.

- [14] M.F. Murphy, G.H. Golub and A.J. Wathen, A note on preconditioning for indefinite linear systems, *SIAM J. Sci. Comput.*, **21** (2000), 1969-1972.
- [15] Z.Z. Bai and M.K. Ng, On inexact preconditioners for nonsymmetric matrices, *SIAM J. Sci. Comput.*, **26** (2005), 1710-1724.
- [16] Y. Cao, M.Q. Jiang and Y.L. Zheng, A splitting preconditioner for saddle point problems, *Numer. Linear Algebra Appl.*, **18** (2011), 875-895.
- [17] Z.Z. Bai, G.H. Golub and J.Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, *Numer. Math.*, **98** (2004), 1-32.
- [18] Z.Z. Bai, G.H. Golub and M.K. Ng, On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *Linear Algebra Appl.*, **428** (2008), 413-440.
- [19] Y. Cao, L.Q. Yao and M.Q. Jiang, A modified dimensional split preconditioner for generalized saddle point problems, *J. Comput. Appl. Math.*, **250** (2013), 70-82.
- [20] Z.Z. Bai, Block alternating splitting implicit iteration methods for saddle-point problems from time-harmonic eddy current models, *Numer. Linear Algebra Appl.*, **19** (2012), 914-936.
- [21] 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** (2016), 922-946.
- [22] V. Simoncini and M. Benzi, Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems, *SIAM J. Matrix Anal. Appl.*, **26** (2004), 377-389.
- [23] L.C. Chan, M.K. Ng and N.K. Tsing, Spectral analysis for HSS preconditioners, *Numer. Math. Theor. Meth. Appl.*, **1** (2008), 57-77.
- [24] M. Benzi, M.J. Gander and G.H. Golub, Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems, *BIT*, **43** (2003), 881-900.
- [25] Z.Z. Bai, G.H. Golub and 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.
- [26] Z.Z. Bai, Optimal parameters in the HSS-like methods for saddle-point problems, *Numer. Linear Algebra Appl.*, **16** (2009), 447-479.
- [27] Y.M. Huang, A practical formula for computing optimal parameters in the HSS iteration methods, *J. Comput. Appl. Math.*, **255** (2014), 142-149.
- [28] 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** (2005), 844-863.
- [29] J.Y. Pan, M.K. Ng and Z.Z. Bai, New preconditioners for saddle point problems, *Appl. Math. Comput.*, **172** (2006), 762-771.
- [30] S.Q. Shen, A note on PSS preconditioners for generalized saddle point problems, *Appl. Math. Comput.*, **237** (2014), 723-729.
- [31] J.L. Zhang, C.Q. Gu and K. Zhang, A relaxed positive-definite and skew-Hermitian splitting preconditioner for saddle point problems, *Appl. Math. Comput.*, **249** (2014), 468-479.
- [32] M. Benzi, M.K. Ng, Q. Niu and Z. Wang, A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations, *J. Comput. Phys.*, **230** (2011), 6185-6202.
- [33] 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** (2015), 41-60.
- [34] H.T. Fan and X.Y. Zhu, A generalized relaxed positive-definite and skew-Hermitian splitting preconditioner for non-Hermitian saddle point problems, *Appl. Math. Comput.*, **258** (2015), 36-48.
- [35] Y. Saad, *Iterative Methods for Sparse Linear Systems* (2nd edn), SIAM: Philadelphia, 2003.
- [36] 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** (2016), 129-145.