Open Mathematics

Research Article

Yunying Huang and Guoliang Chen*

A relaxed block splitting preconditioner for complex symmetric indefinite linear systems

https://doi.org/10.1515/math-2018-0051
Received October 8, 2017; accepted March 29, 2018.

Abstract: In this paper, we propose a relaxed block splitting preconditioner for a class of complex symmetric indefinite linear systems to accelerate the convergence rate of the Krylov subspace iteration method and the relaxed preconditioner is much closer to the original block two-by-two coefficient matrix. We study the spectral properties and the eigenvector distributions of the corresponding preconditioned matrix. In addition, the degree of the minimal polynomial of the preconditioned matrix is also derived. Finally, some numerical experiments are presented to illustrate the effectiveness of the relaxed splitting preconditioner.

Keywords: Complex symmetric linear system, Preconditioner, Krylov subspace method, GMRES

MSC: 65F10, 75F50

1 Introduction

Consider the following large sparse nonsingular complex symmetric linear systems

$$Au = b, \quad A \in \mathbb{C}^{n\times n} \ \text{ and } \ u,\ b \in \mathbb{C}^n, \tag{1}$$

where $A = W + iT$, $W, T \in \mathbb{R}^{n\times n}$ are symmetric matrices and $i = \sqrt{-1}$ denotes the imaginary unit. Such complex linear system (1) arise in many areas of scientific computing and engineering applications, such as FFT-based solution of certain time-dependent PDEs [1], distributed control problems [2], diffuse optical tomography [3], PDE-constrained optimization problems [4], quantum mechanics [5] and so on; see [6–8] and references therein for other applications.

Let $u = x + iy$ and $b = f + ig$ with $x, y, f, g \in \mathbb{R}^n$, then the complex linear system (1) can be equivalently rewritten as the real block two-by-two linear system as follows

$$\widetilde{\mathcal{A}}u = \begin{pmatrix} W & -T \\ T & W \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} = b, \tag{2}$$

or the following two-by-two block real equivalent formulation

$$\mathcal{A}u = \begin{pmatrix} T & -W \\ W & T \end{pmatrix}\begin{pmatrix} x \\ -y \end{pmatrix} = \begin{pmatrix} g \\ f \end{pmatrix} = b. \tag{3}$$

It can be seen that the linear systems (2) and (3) both avoid using complex arithmetic, but the coefficient matrix in (2) and (3) have become doubled in size. Because it is more attractive to use iterative methods

Yunying Huang: School of Mathematical Sciences, Shanghai Key Laboratory of Pure Mathematics and Mathematical Practice, East China Normal University, Dongchuan RD 500, Shanghai 200241, China, E-mail: yunyinghuang15@163.com
***Corresponding Author: Guoliang Chen:** School of Mathematical Sciences, Shanghai Key Laboratory of Pure Mathematics and Mathematical Practice, East China Normal University, Dongchuan RD 500, Shanghai 200241, China, E-mail: glchen@math.ecnu.edu.cn

rather than direct methods for solving (1) or (2), then many efficient iterative methods as well as their numerical properties have been studied in the literature, see [9, 10]. When both W and T are symmetric positive semi-definite with at least one of them being positive definite, some efficient iterative methods have been proposed. Recently, based on the Hermitian and skew-Hermitian splitting (HSS) method [11], Bai et al. [12] developed the modified HSS (MHSS) iteration method to solve the complex linear system (1), which is convergent for any positive constant α. To accelerate the convergence rate of the MHSS iteration method, Bai et al. [13] introduced the preconditioned MHSS (PMHSS) which is unconditionally convergent and shows h-independent convergence behavior. In [14], Bai further analyzed algebraic and convergence properties of the PMHSS iteration method for solving complex linear systems. Moreover, there are also some other effective iterative methods, such as the lopsided PMHSS (LPMHSS) iteration method, the accelerated PMHSS (APMHSS) iteration method, the preconditioned generalized SOR (PGSOR) iterative method, the skew-normal splitting (SNS) method, the double-step scale splitting (DSS) iteration method [15–19], and so on.

If the matrices W and T are symmetric positive semi-definite, some efficient preconditioners have been presented. Bai [20] proposed the rotated block triangular (RBT) preconditioners which is based on the PMHSS preconditioning matrix [21] for the linear system (2). Lang and Ren [22] established the inexact RBT (IRBT) preconditioners. For solving the linear system (2), Liang and Zhang [23] developed the symmetric SOR (SSOR) method.

However, when $W \in \mathbb{R}^{n\times n}$ is symmetric indefinite, $T \in \mathbb{R}^{n\times n}$ is symmetric positive definite, the matrices $\alpha I + W$, $\alpha V + W$ and $\alpha W + T^2$ are indefinite or singular which may lead to the slow convergence speeds of the MHSS method, the PMHSS method and the SNS method. To overcome these problems, in [24], the Hermitian normal splitting (HNS) method and its variant simplified HNS (SHNS) method have been introduced by Wu for solving the complex symmetric linear system (1). Zhang and Dai [25] presented a preconditioned SHNS (PSHNS) iteration method and constructed a new preconditioner. In this paper, we solve the linear system (3) with W being symmetric indefinite and T being symmetric positive definite. In order to solve real formulation (3), Zhang and Dai [26] established a new block preconditioner and also analyzed the spectral properties of the corresponding preconditioned matrix. An improved block (IB) splitting preconditioner was proposed by Zhang et al. in [27] and they gave the convergence property of the corresponding iteration method under suitable conditions.

In this paper, by using the relaxing technique, we construct a relaxed block splitting preconditioner for the real block two-by-two linear system (3). The relaxed splitting preconditioner is much closer to the original coefficient matrix than the block preconditioners in [26]. We analyze the spectral properties of the corresponding preconditioned matrix and show that the corresponding preconditioned matrix has an eigenvalue 1 with algebraic multiplicity at least n. The structure of the eigenvectors and the dimension of the Krylov subspace for the preconditioned matrix are also derived.

The organization of the paper is as follows. In Section 2, a relaxed block splitting preconditioner is presented. In Section 3, we study the distribution of eigenvalues, the form of the eigenvectors and the dimension of the Krylov subspace of the corresponding preconditioned matrix. In Section 4, some numerical examples are presented to compare the new preconditioner with the PSHNS preconditioner and the HSS preconditioner. Finally, we give some brief concluding remarks in Section 5.

2 A new relaxed splitting preconditioner

In this section, by employing the relaxation techniques, we establish a relaxed block splitting preconditioner for solving the linear system (3).

Pan et al. [28] employed the positive-definite and skew-Hermitian splitting (PSS) iteration method in [29] to develop a PSS preconditioner for saddle point problems. For the linear system (3), the coefficient matrix $\mathcal{A}$ can be split as follows

$$\mathcal{A} = \mathcal{J} + \mathcal{K}, \tag{4}$$

where $\mathcal{J} = \begin{pmatrix} T & 0 \\ 0 & 0 \end{pmatrix}$ and $\mathcal{K} = \begin{pmatrix} 0 & -W \\ W & T \end{pmatrix}$. Recently, based on the PSS preconditioner, Zhang and Dai presented a preconditioner in [26] as follows

$$\mathcal{P}_1 = \frac{1}{\alpha}(\alpha I + \mathcal{K})(\alpha I + \mathcal{J}) = \frac{1}{\alpha}\begin{pmatrix} \alpha I & -W \\ W & \alpha I + T \end{pmatrix}\begin{pmatrix} \alpha I + T & 0 \\ 0 & \alpha I \end{pmatrix} = \begin{pmatrix} \alpha I + T & -W \\ W(I + \frac{1}{\alpha}T) & \alpha I + T \end{pmatrix}. \tag{5}$$

Then the difference between the preconditioner $\mathcal{P}_1$ and the coefficient matrix $\mathcal{A}$ is given as follows

$$\mathcal{R}_1 = \mathcal{P}_1 - \mathcal{A} = \begin{pmatrix} \alpha I & 0 \\ \frac{1}{\alpha}WT & \alpha I \end{pmatrix}. \tag{6}$$

A general criterion for an efficient preconditioner is that it should be as close as possible to the coefficient matrix $\mathcal{A}$ which makes the preconditioned matrix have a clustered spectrum, and hope that it is easy to implement. Inspired by the idea of the relaxation technique in [30–33], we give a relaxed splitting preconditioner which is defined as follows

$$\mathcal{P}_2 = \frac{1}{\alpha}\begin{pmatrix} \alpha I & -W \\ W & T \end{pmatrix}\begin{pmatrix} T & 0 \\ 0 & \alpha I \end{pmatrix} = \begin{pmatrix} T & -W \\ \frac{1}{\alpha}WT & T \end{pmatrix}. \tag{7}$$

From (3) and (7), we get the difference between the preconditioner $\mathcal{P}_2$ and the coefficient matrix $\mathcal{A}$

$$\mathcal{R}_2 = \mathcal{P}_2 - \mathcal{A} = \begin{pmatrix} 0 & 0 \\ \frac{1}{\alpha}WT - W & 0 \end{pmatrix}. \tag{8}$$

It is easy to see that the (2, 1)-block matrix in (8) is different from that in (6) and the (1, 1)-block and (2, 2)-block matrix in (8) now vanish, which indicates that the preconditioner $\mathcal{P}_2$ is much closer to the coefficient matrix $\mathcal{A}$ than the preconditioner $\mathcal{P}_1$.

From (8), we know that the preconditioner $\mathcal{P}_2$ can be established by the following splitting of the coefficient matrix $\mathcal{A}$

$$\mathcal{A} = \mathcal{P}_2 - \mathcal{R}_2 = \begin{pmatrix} T & -W \\ \frac{1}{\alpha}WT & T \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ \frac{1}{\alpha}WT - W & 0 \end{pmatrix}. \tag{9}$$

In the following section, we will analyze the spectral properties of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$.

3 Spectral analysis of the preconditioned matrix

In this section, we mainly study the spectral properties of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ and give an upper bound of the dimension of the Krylov subspace for this preconditioned matrix. Therefore, firstly we should get the inverse of the matrix $\mathcal{P}_2$ and the explicit form of the matrix $\mathcal{P}_2^{-1}$ will be given in the following lemma. The proof of this lemma consists of straightforward calculations and is omitted here.

Lemma 3.1. *Let*

$$\mathcal{P} = \begin{pmatrix} \alpha I & -W \\ W & T \end{pmatrix} \quad \text{and} \quad \mathcal{P} = \begin{pmatrix} T & 0 \\ 0 & \alpha I \end{pmatrix}.$$

Here $\mathcal{P}$ has the following block-triangular factorization

$$\mathcal{P} = \begin{pmatrix} I & 0 \\ \frac{1}{\alpha}W & I \end{pmatrix}\begin{pmatrix} \alpha I & 0 \\ 0 & T + \frac{1}{\alpha}W^2 \end{pmatrix}\begin{pmatrix} I & -\frac{1}{\alpha}W \\ 0 & I \end{pmatrix}.$$

Then we obtain

$$\mathcal{P}_2^{-1} = \alpha\widehat{\mathcal{P}}^{-1}\mathcal{P}^{-1} = \begin{pmatrix} T^{-1} - \frac{1}{\alpha}T^{-1}W(T + \frac{1}{\alpha}W^2)^{-1}W & T^{-1}W(T + \frac{1}{\alpha}W^2)^{-1} \\ -\frac{1}{\alpha}(T + \frac{1}{\alpha}W^2)^{-1}W & (T + \frac{1}{\alpha}W^2)^{-1} \end{pmatrix}. \tag{10}$$

In the following, we will analyze the eigenvalue distributions of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$.

Theorem 3.2. *Assume that the coefficient matrix $\mathcal{A}$ is nonsingular, $W \in \mathbb{R}^{n\times n}$ is symmetric indefinite and $T \in \mathbb{R}^{n\times n}$ is symmetric positive definite. Let α be a real positive constant and $U = WT^{-1}W$. The preconditioner $\mathcal{P}_2$ is defined as in (7). Then the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ has eigenvalues at 1 with multiplicity at least n and the remaining eigenvalues are λ_i, $i = 1, 2, \cdots, n$, where λ_i $(i = 1, 2, \cdots, n)$ are the eigenvalues of the matrix $\alpha T^{-1}(\alpha I + U)^{-1}(T + U)$.*

Proof. From (9) and Lemma 3.1, we have

$$\begin{aligned}\mathcal{P}_2^{-1}\mathcal{A} = I - \mathcal{P}_2^{-1}\mathcal{R}_2 &= I - \begin{pmatrix} T^{-1}W(T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I) & 0 \\ (T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I) & 0 \end{pmatrix} \\ &= \begin{pmatrix} I - T^{-1}W(T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I) & 0 \\ -(T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I) & I \end{pmatrix}.\end{aligned}$$

It follows from $U = WT^{-1}W$ that

$$\begin{aligned} I - T^{-1}W(T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I) &= T^{-1}(T - W(T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I)) \\ &= T^{-1}(T - W(W(W^{-1}TW^{-1}+\frac{1}{\alpha}I)W)^{-1}W(\frac{1}{\alpha}T-I)) \\ &= T^{-1}(T - (\frac{1}{\alpha}I + W^{-1}TW^{-1})^{-1}(\frac{1}{\alpha}T-I)) \\ &= T^{-1}(T - (\frac{1}{\alpha}U+I)^{-1}U(\frac{1}{\alpha}T-I)) \\ &= \alpha T^{-1}(\alpha I + U)^{-1}(T+U). \end{aligned}$$

Hence, we get

$$\mathcal{P}_2^{-1}\mathcal{A} = \begin{pmatrix} \alpha T^{-1}(\alpha I+U)^{-1}(T+U) & 0 \\ -(T+\frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T-I) & I \end{pmatrix}. \tag{11}$$

Then, it is obvious from (11) that the results are proved. □

The detail implementation of the preconditioning process will be described as follows. Applying the preconditioner $\mathcal{P}_2$ within a Krylov subspace method, the following system should be solved at each step

$$\mathcal{P}_2 z = \frac{1}{\alpha}\begin{pmatrix} \alpha I & -W \\ W & T \end{pmatrix}\begin{pmatrix} T & 0 \\ 0 & \alpha I \end{pmatrix} z = r, \tag{12}$$

where $z = [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}]^{\mathrm{T}}$ and $r = [r_1^{\mathrm{T}}, r_2^{\mathrm{T}}]^{\mathrm{T}}$ are the current and generalized residual vectors, respectively, $z_1, z_2, r_1, r_2 \in \mathbb{R}^n$. Based on Lemma 3.1 and (12), we have

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \alpha \begin{pmatrix} T^{-1} & 0 \\ 0 & \frac{1}{\alpha}I \end{pmatrix}\begin{pmatrix} I & \frac{1}{\alpha}W \\ 0 & I \end{pmatrix}\begin{pmatrix} \frac{1}{\alpha}I & 0 \\ 0 & (T+\frac{1}{\alpha}W^2)^{-1} \end{pmatrix}\begin{pmatrix} I & 0 \\ -\frac{1}{\alpha}W & I \end{pmatrix}\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \tag{13}$$

Then, we can derive the implementing process of the preconditioner $\mathcal{P}_2$ as follows.

Algorithm 3.3 (The preconditioner $\mathcal{P}_2$). *For a given vector $r = [r_1^{\mathrm{T}}, r_2^{\mathrm{T}}]^{\mathrm{T}}$, the vector $z = [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}]^{\mathrm{T}}$ can be computed by (13) from the following steps:*

```
(1) Compute $t_1 = r_2 - \frac{1}{\alpha}Wr_1$;
(2) Solve $(T + \frac{1}{\alpha}W^2)z_2 = t_1$;
(3) Compute $t_2 = r_1 + Wz_2$;
(4) Solve $Tz_1 = t_2$;
(5) Set the generalized residual vector $z = [z_1^{\mathrm{T}}, z_2^{\mathrm{T}}]^{\mathrm{T}}$.
```

It can be seen from Algorithm 3.3 that we need to solve two sub-linear systems with coefficient matrices $T + \frac{1}{\alpha}W^2$ and T at each iteration. Based on the assumptions of the matrices W and T in Section 1, we know that the two matrices $T + \frac{1}{\alpha}W^2$ and T are both symmetric positive definite. Hence, we can solve the two sub-linear systems $(T + \frac{1}{\alpha}W^2)z_2 = t_1$ and $Tz_1 = t_2$ by applying the sparse Cholesky decomposition, the conjugate gradient (CG) method or the preconditioned CG method. In order to prove the effectiveness of the preconditioner $\mathcal{P}_2$, we compare it with the PSHNS preconditioner and the HSS preconditioner which are given as follows:

$$\mathcal{P}_{\text{PSHNS}} = \frac{1}{2\alpha}(\alpha W + iI)(\alpha T + I) \text{ and } \mathcal{P}_{\text{HSS}} = \frac{1}{\,}\begin{pmatrix} \alpha I + T & 0 \\ 0 & \alpha I + T \end{pmatrix}\begin{pmatrix} \alpha I & -W \\ W & \alpha I \end{pmatrix}.$$

Then, the implementing process about the PSHNS preconditioner and the HSS preconditioner with GMRES will be described as follows, respectively.

Algorithm 3.4 (The PSHNS Preconditioner). *For a given vector r, the vector z can be computed from the following steps:*
(1) Solve $(\alpha V + iW)d = 2\alpha r$;
(2) Solve $(\alpha T + WV^{-1}W)z = Wd$.

Algorithm 3.5 (The HSS Preconditioner). *For a given vector $r = [r_1^{\mathrm{T}},\ r_2^{\mathrm{T}}]^{\mathrm{T}}$, the vector $z = [z_1^{\mathrm{T}},\ z_2^{\mathrm{T}}]^{\mathrm{T}}$ can be computed from the following steps:*
(1) Solve $(\alpha I + T)\nu_1 = r_1$;
(2) Solve $(\alpha I + T)\nu_2 = r_2$;
(3) Compute $\mu_1 = \alpha\nu_2 - W\nu_1$;
(4) Solve $(\alpha I + \frac{1}{\alpha}W^2)z_2 = \mu_1$;
(5) Compute $z_1 = \nu_1 + \frac{1}{\alpha}Wz_2$;
(6) Set the generalized residual vector $z = [z_1^{\mathrm{T}},\ z_2^{\mathrm{T}}]^{\mathrm{T}}$.

From Algorithm 3.4, we can see that the PSHNS preconditioner iteration involves the complex coefficient matrix $\alpha V + iW$ whose corresponding linear subsystem may be hard to be solved. Meanwhile, it can be seen from Algorithm 3.5 that the implementation of the HSS preconditioner need to solve three symmetric positive definite linear subsystems.

In the following, we will discuss the eigenvector distributions of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ in detail.

Theorem 3.6. *Let the preconditioner $\mathcal{P}_2$ be defined as in (7), then the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ has $n + i + j$ $(0 \le i + j \le n)$ linearly independent eigenvectors, which are described as follows:*
(1) n eigenvectors $\begin{pmatrix} 0 \\ v_k \end{pmatrix}$ $(k = 1, 2, \cdots, n)$ that correspond to the eigenvalue 1, where v_k $(k = 1, 2, \cdots, n)$ are arbitrary linearly independent vectors.
(2) i $(0 \le i \le n)$ eigenvectors $\begin{pmatrix} u_k^1 \\ v_k^1 \end{pmatrix}$ $(1 \le k \le i)$ that correspond to the eigenvalue 1, where $u_k^1 \neq 0$, $Tu_k^1 = \alpha u_k^1$ and v_k^1 are arbitrary vectors.
(3) j $(0 \le j \le n)$ eigenvectors $\begin{pmatrix} u_k^2 \\ v_k^2 \end{pmatrix}$ $(1 \le k \le j)$ that correspond to the eigenvalues $\lambda_k \neq 1$, where $\alpha(T + U)u_k^2 = \lambda_k(\alpha I + U)Tu_k^2$, $u_k^2 \neq 0$ and $v_k^2 = \frac{1}{1-\lambda_k}(T + \frac{1}{\alpha}W^2)^{-1}(\frac{1}{\alpha}WT - W)u_k^2$.

Proof. Let λ be an eigenvalue of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ and $\begin{pmatrix} u \\ v \end{pmatrix}$ be the corresponding eigenvector. Then

$$\begin{pmatrix} \alpha T^{-1}(\alpha I + U)^{-1}(T + U) & 0 \\ -(T + \frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T - I) & I \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \lambda\begin{pmatrix} u \\ v \end{pmatrix}.$$

By simple calculation, we have

$$\begin{cases} \alpha(T+U)u = \lambda(\alpha I + U)Tu, \\ (T + \frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T - I)u = (1-\lambda)v. \end{cases} \tag{14}$$

If $\lambda = 1$, the Eq. (14) become

$$\begin{cases} \alpha(T+U)u = (\alpha I + U)Tu, \\ (T + \frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T - I)u = 0. \end{cases} \tag{15}$$

Then we have the following two situations.

(1) When $u = 0$, the Eq. (15) are always true. Hence, there are n linearly independent eigenvectors $\begin{pmatrix} 0 \\ v_k \end{pmatrix}$ $(k = 1, 2, \cdots, n)$ corresponding to the eigenvalue 1, where v_k $(k = 1, 2, \cdots, n)$ are arbitrary linearly independent vectors.

(2) When $u \neq 0$, through a simple calculation for (15), we have $Tu = \alpha u$. Then, there will be i $(0 \leq i \leq n)$ linearly independent eigenvectors $\begin{pmatrix} u_k^1 \\ v_k^1 \end{pmatrix}$ $(1 \leq k \leq i)$ that correspond to the eigenvalue 1, where $u_k^1 \neq 0$, $Tu_k^1 = \alpha u_k^1$ and v_k^1 are arbitrary vectors.

Next, we consider the case $\lambda \neq 1$. It can be seen from (14) that the results in (3) hold.

Finally, we show that the $n + i + j$ eigenvectors are linearly independent. Let $c = [c_1, c_2, \cdots, c_n]^{\mathrm{T}}$, $c^1 = [c_1^1, c_2^1, \cdots, c_i^1]^{\mathrm{T}}$ and $c^2 = [c_1^2, c_2^2, \cdots, c_j^2]^{\mathrm{T}}$ be three vectors with $0 \leq i, j \leq n$. Then, we have to show that

$$\begin{pmatrix} 0 & \cdots & 0 \\ v_1 & \cdots & v_n \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} + \begin{pmatrix} u_1^1 & \cdots & u_i^1 \\ v_1^1 & \cdots & v_i^1 \end{pmatrix} \begin{pmatrix} c_1^1 \\ \vdots \\ c_i^1 \end{pmatrix} + \begin{pmatrix} u_1^2 & \cdots & u_j^2 \\ v_1^2 & \cdots & v_j^2 \end{pmatrix} \begin{pmatrix} c_1^2 \\ \vdots \\ c_j^2 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{16}$$

holds if and only if the vectors c, c^1 and c^2 are all zero vectors, where the first matrix consists of the eigenvectors corresponding to the eigenvalue 1 for the case (1), the second matrix consists of those for the case (2) and the third matrix consists of the eigenvectors corresponding to $\lambda \neq 1$ for the case (3). By multiplying matrix $\mathcal{P}_2^{-1}\mathcal{A}$ from left on both sides of Eq. (16), we get

$$\begin{pmatrix} 0 & \cdots & 0 \\ v_1 & \cdots & v_n \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} + \begin{pmatrix} u_1^1 & \cdots & u_i^1 \\ v_1^1 & \cdots & v_i^1 \end{pmatrix} \begin{pmatrix} c_1^1 \\ \vdots \\ c_i^1 \end{pmatrix} + \begin{pmatrix} u_1^2 & \cdots & u_j^2 \\ v_1^2 & \cdots & v_j^2 \end{pmatrix} \begin{pmatrix} \lambda_1 c_1^2 \\ \vdots \\ \lambda_j c_j^2 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{17}$$

From the difference between (17) and (16), we obtain

$$\begin{pmatrix} u_1^2 & \cdots & u_j^2 \\ v_1^2 & \cdots & v_j^2 \end{pmatrix} \begin{pmatrix} (\lambda_1 - 1)c_1^2 \\ \vdots \\ (\lambda_j - 1)c_j^2 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Because the eigenvalues $\lambda_k \neq 1$ and $\begin{pmatrix} u_k^2 \\ v_k^2 \end{pmatrix}$ $(k = 1, \cdots, j)$ are linearly independent, we know that $c_k^2 = 0$ $(k = 1, \cdots, j)$. Thus, the Eq. (16) reduces to

$$\begin{pmatrix} 0 & \cdots & 0 \\ v_1 & \cdots & v_n \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} + \begin{pmatrix} u_1^1 & \cdots & u_i^1 \\ v_1^1 & \cdots & v_i^1 \end{pmatrix} \begin{pmatrix} c_1^1 \\ \vdots \\ c_i^1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

As the vectors u_k^1 $(k = 1, \cdots, i)$ are also linearly independent, then we have $c_k^1 = 0$ $(k = 1, \cdots, i)$. Hence, the above equation becomes

$$\begin{pmatrix} 0 & \cdots & 0 \\ v_1 & \cdots & v_n \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Since v_k $(k = 1, \cdots, n)$ are linearly independent, we have $c_k = 0$ $(k = 1, \cdots, n)$. Therefore, the $n + i + j$ eigenvectors are linearly independent. The proof of this theorem is completed □

Based on the Krylov subspace theory, we know that the iterative method with an optimal property (such as GMRES [34] method) will terminate at the moment when the degree of the minimal polynomial is attained. Next, we will give an upper bound of the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$.

Theorem 3.7. *Assume that the conditions of Theorem 3.2 are satisfied and let the preconditioner $\mathcal{P}_2$ be defined as in (7). Then the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ is at most $n + 1$. Thus, the dimension of the Krylov subspace $\mathcal{K}(\mathcal{P}_2^{-1}\mathcal{A},\ b)$ is at most $n + 1$.*

Proof. From (11), we know that the preconditioned matrix can be expressed as

$$\mathcal{P}_2^{-1}\mathcal{A} = \begin{pmatrix} \theta_1 & 0 \\ \theta_2 & I \end{pmatrix},$$

where $\theta_1 = \alpha T^{-1}(\alpha I + U)^{-1}(T + U)$ and $\theta_2 = -(T + \frac{1}{\alpha}W^2)^{-1}W(\frac{1}{\alpha}T - I)$. Let μ_i $(i = 1, \cdots, n)$ be the eigenvalues of matrix θ_1. The characteristic polynomial of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ is

$$(\mathcal{P}_2^{-1}\mathcal{A} - I)^n \prod_{i=1}^{n}(\mathcal{P}_2^{-1}\mathcal{A} - \mu_i I).$$

Then

$$(\mathcal{P}_2^{-1}\mathcal{A} - I) \prod_{i=1}^{n}(\mathcal{P}_2^{-1}\mathcal{A} - \mu_i I) = \begin{pmatrix} (\theta_1 - I) \prod_{i=1}^{n}(\theta_1 - \mu_i I) & 0 \\ \theta_2 \prod_{i=1}^{n}(\theta_1 - \mu_i I) & 0 \end{pmatrix}.$$

Because μ_i $(i = 1, \cdots, n)$ are the eigenvalues of θ_1, then by the Hamilton-Cayley theorem, we have

$$\prod_{i=1}^{n}(\theta_1 - \mu_i I) = 0.$$

Therefore, we obtain that the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ is at most $n+1$. It is well known that the degree of the minimal polynomial is equal to the dimension of the corresponding Krylov subspace. So the dimension of the Krylov subspace $\mathcal{K}(\mathcal{P}_2^{-1}\mathcal{A},\ b)$ is also at most $n + 1$. Thus, the proof of this theorem is completed. □

4 Numerical experiments

In this section, some numerical experiments are presented to illustrate the effectiveness of the preconditioner $\mathcal{P}_2$ for the linear system (3). We compare the performance of the preconditioner $\mathcal{P}_2$ with the PSHNS preconditioner and the HSS preconditioner. We denote the number of iteration steps as ”Iter”, the elapsed CPU time in seconds as “CPU", and the relative residual norm as “RES". Unless otherwise specified, we use left preconditioning with the GMRES method [34]. All the computations are implemented in MATLAB on a PC computer with Intel (R) Core (TM) i5 CPU 2.50GHz, and 4.00 GB memory.

In our implementations, the zero vector is adopted as the initial vector and the iteration stops when

$$RES := \frac{\sqrt{\|g - Tx^k - Wy^k\|_2^2 + \|f - Wx^k + Ty^k\|_2^2}}{\sqrt{\|g\|_2^2 + \|f\|_2^2}} < 10^{-6}.$$

The maximum number of iteration steps allowed is set to 3000 for the unpreconditioned GMRES method, and to 500 for the preconditioned GMRES method. It should be emphasized that the sub-linear systems arising from the application of the preconditioners are solved by direct methods. In matlab, the sub-systems of linear equations are solved through sparse Cholesky or LU factorization in combination with AMD or column AMD reordering. A symbol “-" is used to indicate that the method does not obtain the required stopping criterion before maximum iterations or out of memory. We test three values of the parameter α, that is, $\alpha = 0.001$, $\alpha = 0.01$, $\alpha = 1$.

Example 1. *We consider the following complex symmetric linear system which comes from [13, 24, 26]*

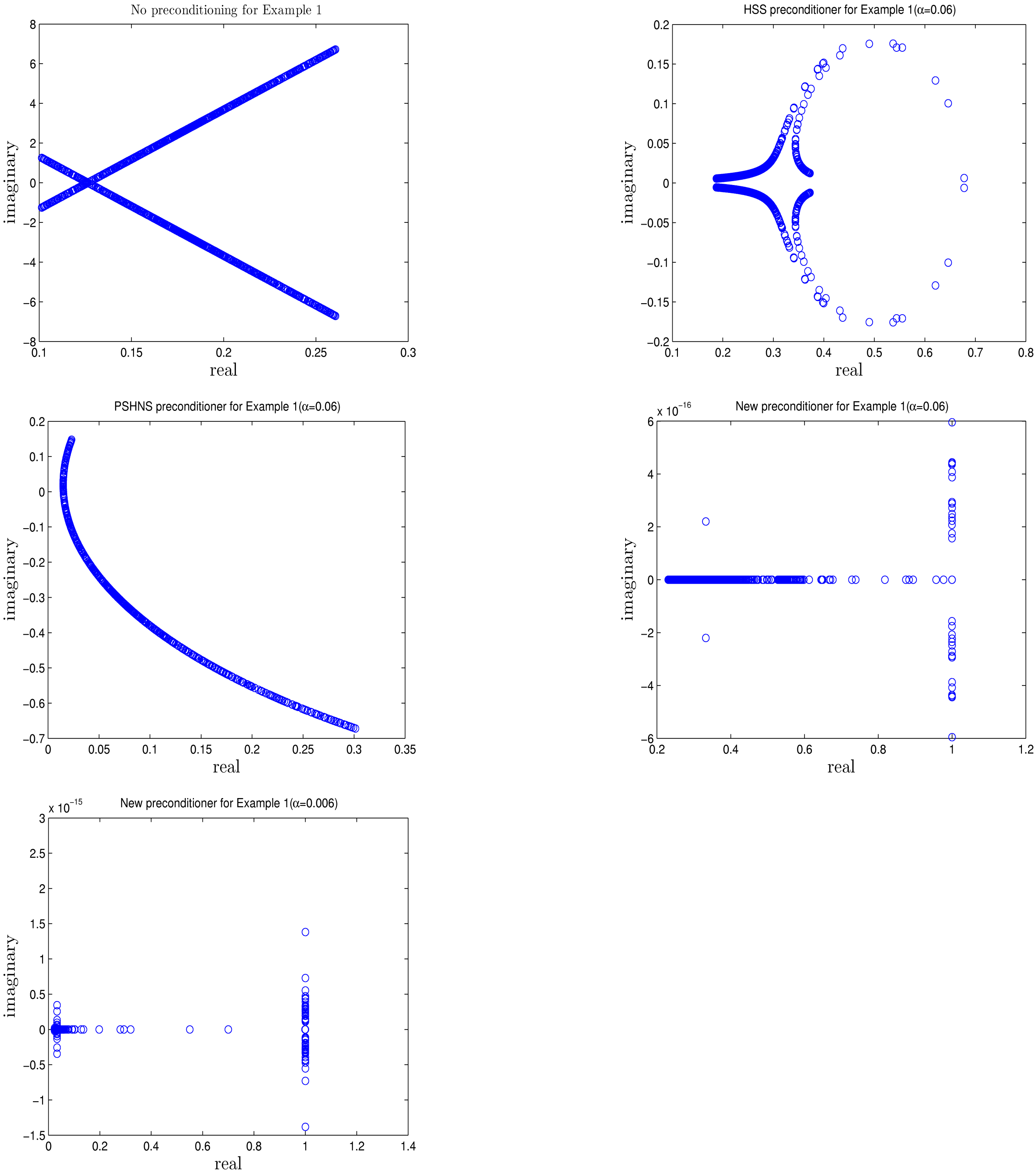
$$[(-\omega^2 M + K) + i(\omega C_v + C_H)]x = b.$$

where M and K are the inertia and stiffness matrices, C_v and C_H are the viscous and hysteretic damping matrices, respectively, ω is the driving circular frequency, $K = I \otimes V_m + V_m \otimes I$, $V_m = h^{-2} tridiag(-1, 2, -1) \in \mathbb{R}^{m \times m}$ is a tridiagonal matrix, $\omega = 2\pi$, $h = \frac{1}{m+1}$, $C_v = \frac{1}{2}M$, $C_H = \mu K$ with $\mu = 0.02$ being a damping coefficient. We choose $W = h^2(-\omega^2 M + K)$, $T = h^2(\omega C_v + C_H)$ and test $M = 10I,\ 15I,\ 25I,\ 35I,\ 50I$. In this example, we set m = 32 and the right-hand side $b = \mathcal{A} * ones(2m^2, 1)$.

Table 1. Numerical results of GMRES and Preconditioned GMRES methods for Example 1

Preconditioner	α	M	10I	15I	25I	35I	50I
unpreconditioner		Iter	126	155	197	243	305
		CPU	0.26206	0.36578	0.56839	0.83868	1.34459
$\mathcal{P}_{\text{PSHNS}}$	$\alpha = 0.001$	Iter	71	76	87	93	108
		CPU	0.16812	0.189197	0.224558	0.241821	0.300449
$\mathcal{P}_{\text{HSS}}$	$\alpha = 0.001$	Iter	22	22	23	26	31
		CPU	0.052688	0.053715	0.05616	0.060838	0.077092
$\mathcal{P}_2$	$\alpha = 0.001$	Iter	12	11	12	12	13
		CPU	0.032933	0.039289	0.041518	0.035258	0.037504
$\mathcal{P}_{\text{PSHNS}}$	$\alpha = 0.01$	Iter	71	76	87	92	108
		CPU	0.04293	0.04455	0.05594	0.06078	0.007809
$\mathcal{P}_{\text{HSS}}$	$\alpha = 0.01$	Iter	15	16	20	21	27
		CPU	0.00858	0.00885	0.01142	0.01187	0.01464
$\mathcal{P}_2$	$\alpha = 0.01$	Iter	12	11	10	11	11
		CPU	0.00792	0.00776	0.00714	0.00751	0.00761
$\mathcal{P}_{\text{PSHNS}}$	$\alpha = 1$	Iter	31	34	37	35	27
		CPU	0.06931	0.0775	0.08297	0.07733	0.05919
$\mathcal{P}_{\text{HSS}}$	$\alpha = 1$	Iter	77	73	54	40	29
		CPU	0.18827	0.18229	0.13074	0.08941	0.06585
$\mathcal{P}_2$	$\alpha = 1$	Iter	15	15	18	16	14
		CPU	0.03641	0.03653	0.0418	0.0383	0.03443

Fig. 1. The eigenvalue distributions of unpreconditioned matrix, the preconditioned matrices $\mathcal{P}_{\mathrm{HSS}}^{-1}\mathcal{A}$, $\mathcal{P}_{\mathrm{PSHNS}}^{-1}\mathcal{A}$ and $\mathcal{P}_2^{-1}\mathcal{A}$ for Example 1 with $M = 35I$.



No preconditioning for Example 1

HSS preconditioner for Example 1(α=0.06)

PSHNS preconditioner for Example 1(α=0.06)

New preconditioner for Example 1(α=0.06)

New preconditioner for Example 1(α=0.006)

By choosing different α and M, we list the numerical results of the unpreconditioned GMRES method, the $\mathcal{P}_2$ preconditioned GMERS method, the PSHNS preconditioned GMRES method and the HSS preconditioned GMRES method in Table 1. Table 1 indicates that applying the $\mathcal{P}_2$ preconditioned GMERS method requires less iteration steps and CPU times than applying the other preconditioned GMRES methods by choosing suitable parameters. In Table 1, we find that the PSHNS preconditioner and the HSS preconditioner are both more sensitive to the parameter α than the preconditioner $\mathcal{P}_2$. Figure 1 describes the eigenvalue distributions of the original coefficient matrix $\mathcal{A}$, the preconditioned matrix $\mathcal{P}_{\mathrm{HSS}}^{-1}\mathcal{A}$ (for $\alpha = 0.06$), the preconditioned matrix $\mathcal{P}_{\mathrm{PSHNS}}^{-1}\mathcal{A}$ (for $\alpha = 0.06$) and the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ (for $\alpha = 0.06$ and $\alpha = 0.006$) with $M = 35I$.

From Figure 1, we observe that the eigenvalue distributions of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ are much better than those of the other preconditioned matrices. Moreover, the smaller parameter α generally makes the eigenvalues of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ become more and more clustered.

Example 2. *We consider the following complex symmetric linear system [25, 26]*

$$[(T_m \otimes I_m + I_m \otimes T_m - k^2h^2(I_m \otimes I_m)) + i\sigma_2(I_m \otimes I_m)]x = b,$$

where $T_m = tridiag(-1, 2, -1)$ *is a tridiagonal matrix with order* m *and* k *denotes the wavenumber,* $\sigma_2 = 0.1$, $h = \frac{1}{m+1}$ *and* $b = \mathcal{A} * ones(2m^2, 1)$. *Here, we choose* $W = T_m \otimes I_m + I_m \otimes T_m - k^2h^2(I_m \otimes I_m)$ *and* $T = \sigma_2(I_m \otimes I_m)$.

Table 2. Numerical results of GMRES and Preconditioned GMRES methods for Example 2

Preconditioner	α	k	10	20	30	40	50
		m^2	16^2	32^2	64^2	128^2	256^2
unpreconditioner		Iter	47	121	324	443	-
		CPU	0.02811	0.24058	5.20897	29.3203	-
$\mathcal{P}_{\text{PSHNS}}$	$\alpha = 0.001$	Iter	29	68	123	116	99
		CPU	0.073258	0.154954	1.349414	3.920827	17.513853
$\mathcal{P}_{\text{HSS}}$	$\alpha = 0.001$	Iter	12	24	45	103	211
		CPU	0.019048	0.047163	0.463713	7.020476	160.268749
$\mathcal{P}_2$	$\alpha = 0.001$	Iter	5	10	16	30	38
		CPU	0.013813	0.02835	0.156661	1.394879	11.409131
$\mathcal{P}_{\text{PSHNS}}$	$\alpha = 0.01$	Iter	29	68	123	115	98
		CPU	0.03101	0. 15599	1.33949	4.06219	30.38906
$\mathcal{P}_{\text{HSS}}$	$\alpha = 0.01$	Iter	9	19	30	48	57
		CPU	0.00913	0.03617	0.23154	1.64472	14.81967
$\mathcal{P}_2$	$\alpha = 0.01$	Iter	6	8	13	18	17
		CPU	0.00809	0.02318	0.12537	0.68779	3.90741
$\mathcal{P}_{\text{PSHNS}}$	$\alpha = 1$	Iter	14	29	41	39	34
		CPU	0.060745	0.04931	0.235227	0.859889	4.907534
$\mathcal{P}_{\text{HSS}}$	$\alpha = 1$	Iter	29	52	62	64	65
		CPU	0.033833	0.127379	0.705647	3.464846	23.253554
$\mathcal{P}_2$	$\alpha = 1$	Iter	10	17	20	21	22
		CPU	0.019189	0.037429	0.176674	0.842907	4.844297

Fig. 2. The eigenvalue distributions of unpreconditioned matrix, the preconditioned matrices $\mathcal{P}_{\mathrm{HSS}}^{-1}\mathcal{A}$, $\mathcal{P}_{\mathrm{PSHNS}}^{-1}\mathcal{A}$ and $\mathcal{P}_2^{-1}\mathcal{A}$ for Example 2 with $k = 20,\ m = 32$.

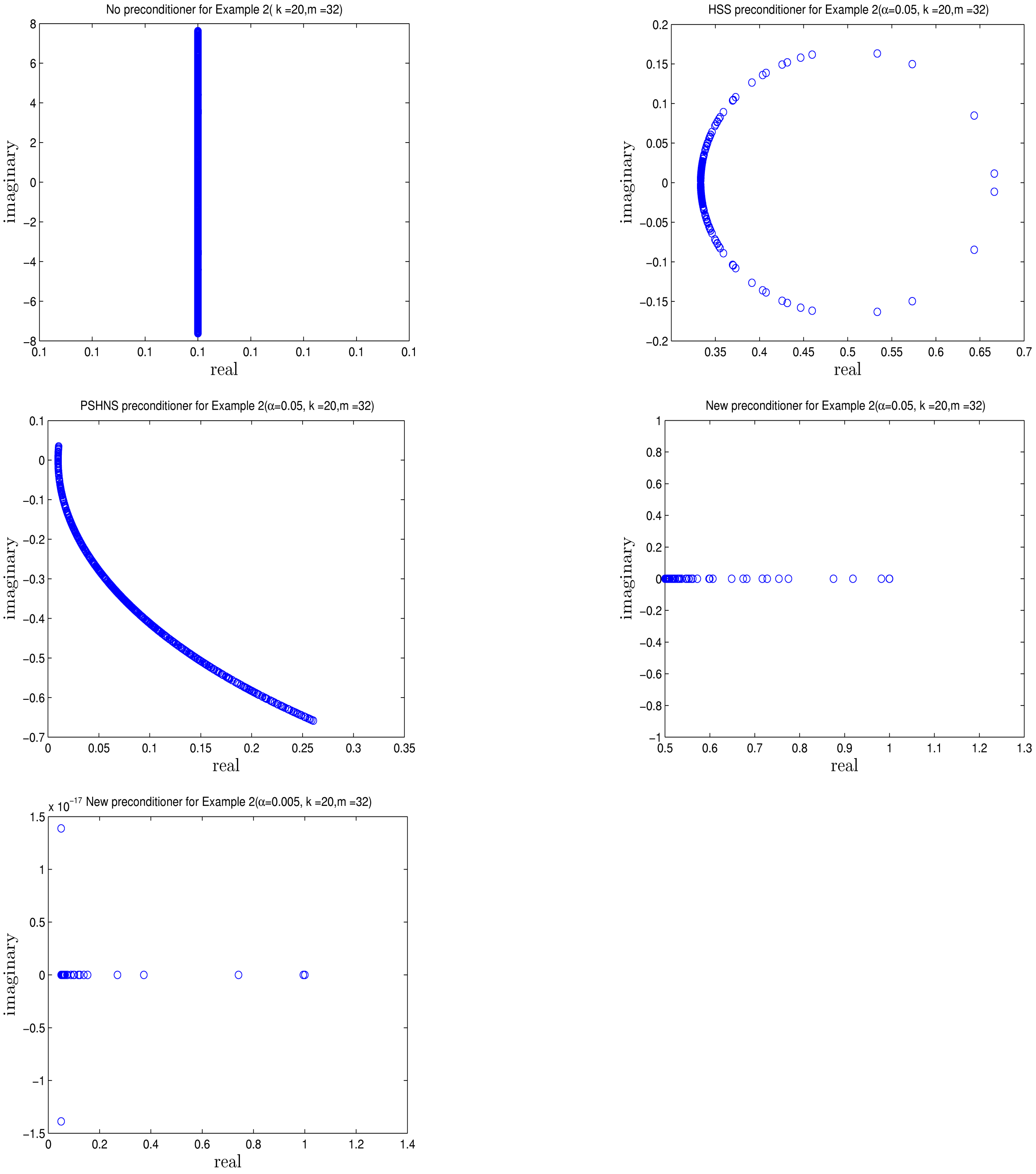


Table 2 shows that these three kinds preconditioned GMRES methods present much better convergence behavior than the unpreconditioned GMRES method. Meanwhile, the $\mathcal{P}_2$ preconditioned GMERS method gives a better numerical results than the PSHNS preconditioned GMRES method and the HSS preconditioned GMRES method with different parameters α in terms of iteration steps and CPU times. From Figure 2, we find that the eigenvalue distributions of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ are much better than those of the other preconditioned matrices and the eigenvalues of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ will be more gathered as α decrease.

5 Conclusions

In this paper, we have proposed a relaxed block splitting preconditioner $\mathcal{P}_2$ for the real block two-by-two linear system (3) and analyzed the eigenvalue distributions of the corresponding preconditioned matrix. Meanwhile, the structure of the eigenvectors and an upper bound for the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_2^{-1}\mathcal{A}$ were also derived. Some numerical experiments are given to illustrate the effectiveness of the preconditioner $\mathcal{P}_2$ for solving the linear system (3).

Acknowledgement: The authors would like to thank the referees for their very detailed comments and suggestions which improved the presentation of this paper greatly.

This work is supported by the National Natural Science Foundation of China No. 11471122 and supported in part by Science and Technology Commission of Shanghai Municipality (No. 18dz2271000).

References

[1] Betts J.-T., Kolmanovsky I., Practical methods foroptimal controlusing nonlinear programming, Appl. Mech. Rev., 2002, 55, 68

[2] Lass O., Vallejos M., Borzi A., Douglas C. C., Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing, 2009, 84, 27-48

[3] Arridge S. R., Optical tomography in medical imaging, Inverse Problem., 1999, 15, R41-R93

[4] Zheng Z., Zhang G.-F., Zhu M.-Z., A note on preconditioners for complex linear systems arising from PDE-constrained optimization problems, Appl. Math. Lett., 2016, 61, 114-121

[5] van Dijk W., Toyama F. M., Accurate numerical solutions of the time-dependent Schrödinger equation, Phys. Rev. E., 2007, 75, 036707-1-036707-10

[6] Benzi M., Bertaccini D., Block preconditioning of real-valued iterative algorithms for complex linear systems, IMA J. Numer. Anal., 2008, 28, 598-618

[7] Feriani A., Perotti F., Simoncini V., Iterative system solvers for the frequency analysis of linear mechanical systems, Comput. Methods Appl. Mech. Engrg., 2000, 190, 1719-1739

[8] Bao G., Sun W.-W., A fast algorithm for the electromagnetic scattering from a large cavity, SIAM J. Sci. Comput., 2005, 27, 553-574

[9] Benzi M., Golub G. H., Liesen J., Numerical solution of saddle point problems, Acta Numer., 2005, 14, 1-137

[10] Saad Y., Iterative Methods for Sparse Linear Systems, SIAM, 2003

[11] Bai Z.-Z., Golub G. H., Ng M. K., Hermitian and skew-Hermitian splitting methods for non- Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl., 2003, 24, 603-626

[12] Bai Z.-Z., Benzi M., Chen F., Modified HSS iteration methods for a class of complex symmetric linear systems, Computing, 2010, 87, 93-111

[13] Bai Z.-Z., Benzi M., Chen F., On preconditioned MHSS iteration methods for complex symmetric linear systems, Numer. Algorithms., 2011, 56, 297-317

[14] Bai Z.-Z., On preconditioned iteration methods for complex linear systems, J. Eng. Math., 2015, 93, 41-60

[15] Li X., Yang A.-L., Wu Y.-J., Lopsided PMHSS iteration method for a class of complex symmetric linear systems, Numer. Algorithms., 2014, 66, 555-568

[16] Zheng Q.-Q., Ma C.-F., Accelerated PMHSS iteration methods for complex symmetric linear systems, Numer. Algorithms., 2016, 73, 501-516

[17] Hezari D., Edalatpour V., Salkuyeh D. K., Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations, Numer. Linear Algebra Appl., 2014, 22, 761-776

[18] Bai Z.-Z., Several splittings for non-Hermitian linear systems, Sci. China Ser. A: Math., 2008, 51, 1339-1348

[19] Zheng Z., Huang F.-L., Peng Y.-C., Double-step scale splitting iteration method for a class of complex symmetric linear systems, Appl. Math. Lett., 2017, 73, 91-97

[20] Bai Z.-Z., Rotated block triangular preconditioning based on PMHSS, Sci. China Math. 2013, 56, 2523-2538

[21] Bai Z.-Z., Benzi M., Chen F., Wang Z.-Q., Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems, IMA J. Numer. Anal., 2013, 33, 343-369

[22] Lang C., Ren Z.-R., Inexact rotated block triangular preconditioners for a class of block two-by-two matrices, J. Eng. Math., 2015, 93, 87-98

[23] Liang Z.-Z., Zhang G.-F., On SSOR iteration method for a class of block two-by-two linear systems, Numer. Algorithms., 2016, 71, 655-671

[24] Wu S.-L., Several variants of the Hermitian and skew-Hermitian splitting method for a class of complex symmetric linear systems, Numer. Linear Algebra Appl., 2015, 22, 338-356

[25] Zhang J.-H., Dai H., A new splitting preconditioner for the iterative solution of complex symmetric indefinite linear systems, Appl. Math. Lett., 2015, 49, 100-106

[26] Zhang J.-H., Dai H., A new block preconditioner for complex symmetric indefinite linear systems, Numer. Algorithms., 2017, 74, 889-903

[27] Zhang J.-L., Fan H.-T., Gu C.-Q., An improved block splitting preconditioner for complex symmetric indefinite linear systems, Numer. Algorithms., 2018, 77, 451-478

[28] Pan J.-Y., Ng M. K., Bai Z.-Z., New preconditioners for saddle point problems, Appl. Math. Comput., 2006, 172, 762-771

[29] Bai Z.-Z., Golub G. H., Lu L.-Z., Yin J.-F., Block triangular and skew-Hermitian splitting method for positive definite linear systems, SIAM J. Sci. Comput., 2005, 26, 844-863

[30] Cao Y., Dong J.-L., Wang Y.-M., A relaxed deteriorated PSS preconditioner for nonsymmetric saddle point problems from the steady Navier-Stokes equation, J. Comput. Appl. Math., 2015, 273, 41-60

[31] Fan H.-T., Zheng B., Zhu X.-Y., A relaxed positive semi-definite and skew-Hermitian splitting preconditioner for non-Hermitian generalized saddle point problems, Numer. Algor., 2016, 72, 813-834

[32] Cao Y., Yao L.-Q., Jiang M.-Q., Niu Q., A relaxed HSS preconditioner for saddle point problems from meshfree discretization, J. Comput. Math., 2013, 31, 398-421

[33] Zhang K., Zhang J.-L., Gu C.-Q., A new relaxed PSS preconditioner for nonsymmetric saddle point problems, Appl. Math. Comput., 2017, 308, 115-129

[34] Saad Y., Schultz M. H., GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 1986, 7, 856-869