Numer Algor (2018) 77:451–478
DOI 10.1007/s11075-017-0323-z

ORIGINAL PAPER

An improved block splitting preconditioner for complex symmetric indefinite linear systems

Ju-Li Zhang[1,2] · Hong-Tao Fan[3] · Chuan-Qing Gu[4]

Received: 6 September 2016 / Accepted: 29 March 2017 / Published online: 3 April 2017

Abstract In this paper, an improved block splitting preconditioner for a class of complex symmetric indefinite linear systems is proposed. By adopting two iteration parameters and the relaxation technique, the new preconditioner not only remains the same computational cost with the block preconditioners but also is much closer to the original coefficient matrix. The theoretical analysis shows that the corresponding iteration method is convergent under suitable conditions and the preconditioned matrix can have well-clustered eigenvalues around (0, 1) with a reasonable choice of the relaxation parameters. An estimate concerning the dimension of the Krylov subspace for the preconditioned matrix is also obtained. Finally, some numerical experiments are presented to illustrate the effectiveness of the presented preconditioner.

The work is partially supported by National Natural Science Foundation (No. 11371243), (No. 11571004), (No. 11471211), Foundation of Zhe-jiang Educational Committee (Y201431769).

✉ Hong-Tao Fan
fanht14@lzu.edu.cn

Ju-Li Zhang
xzhzhangjuli@163.com

1 School of Fundamental Studies, Shanghai University of Engineering Science, Shanghai, 201620, People's Republic of China

2 Department of Mathematics, Zhejiang A&F University, Zhejiang, 311300, People's Republic of China

3 School of Mathematics and Statistics, Gansu Key Laboratory of Applied Mathematics and Complex Systems, Lanzhou University, Lanzhou, Gansu 730000, People's Republic of China

4 Department of Mathematics, Shanghai University, Shanghai 200444, People's Republic of China

Keywords Complex symmetric linear systems · Preconditioning · Relaxation parameters · GMRES

Mathematics Subject Classification (2000) 65F10 · 75N50

1 Introduction

We consider the iterative solution of the large sparse system of linear equations

$$Ax = b,\ A \in \mathbb{C}^{n\times n} \text{ and } x, b \in \mathbb{C}^n, \tag{1}$$

where $A = W + iT$ is a complex symmetric matrix, with $W, T \in \mathbb{R}^{n\times n}$ being symmetric matrices and $i = \sqrt{-1}$ being the imaginary unit. Complex linear systems of this kind arise from many problems such as electromagnetism problem, the discretization of different types of Helmholtz equations, structural dynamics, optical tomography problem, and so on. For more details, readers can consult [1] and references therein.

Let $x = y + iz$ and $b = f + ig$, then the complex symmetric linear system (1) is formally equivalent to the following real block two-by-two system of linear equations:

$$\bar{\mathscr{A}}\bar{x} = \begin{bmatrix} W & -T \\ T & W \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} = \bar{b}. \tag{2}$$

The linear system (2) avoid using complex arithmetic. But meanwhile, the coefficient matrix here has also become doubled in size, i.e., $\bar{\mathscr{A}} \in \mathbb{R}^{2n\times 2n}$. In order to solve the system (1) or (2) efficiently and fast, many methods have been proposed in the past few years [17, 23]. Among all the candidates, the iterative methods are more attractive than direct methods for minimal requirement of computer storage and easiest precess of computer implementation. Recently, many splitting iteration methods based on the Hermitian and skew-Hermitian splitting (HSS) [2, 3, 5] have been proposed to solve the complex symmetric linear system (1). When the matrices W, T are symmetric positive semi-definite with at least one of them being positive definite, Bai et al. [6, 8, 10] introduced the modified Hermitian and skew-Hermitian splitting (MHSS) iteration method and the preconditioned MHSS (PMHSS) iteration method for solving the system (1). It has been proved in [6] that the MHSS iteration method converges to the unique solution of (1) unconditionally. Bai et al. also established the convergence theory for the PMHSS iteration method under suitable conditions in [8] and showed the h-independent behavior. To further generalize the MHSS and PMHSS iteration methods, Zheng et al. [29] proposed an accelerated PMHSS (APMHSS) iteration method for (1) and also analyzed the convergence property. There are also some other effective iterative methods, such as the skew-normal splitting (SNS) method [7], the Hermitian normal splitting (HNS) method and its variant simplified HNS (SHNS) method [24], and so on.

When, however, the matrix W is symmetric indefinite and T is symmetric positive definite, the MHSS (PMHSS) method and SNS method may be unacceptably slow or not applicable due to the fact that the coefficient matrices $\alpha I + W$ and $\alpha W + T^2$ are indefinite or singular. And the HNS or SHNS method includes the complex arithmetics in each inner iteration, which can result in an expensive computational cost. The similar observation can also be found in [28]. Under these circumstances, an appropriate preconditioner is needed to circumvent the difficulties [12, 13, 21].

If the matrices W and T are symmetric positive semi-definite, Bai [11] introduced the rotated block triangular (RBT) preconditioner for the linear system (2). And then to accelerate the computation of the RBT preconditioner, Lang et al. [18] proposed the inexact RBT preconditioner. In this paper, we will focus on the case when the matrix W is symmetric indefinite and T is symmetric positive definite. Since in this case, the complex linear system (1) is also equivalent to the real block two-by-two system as follows:

$$\mathscr{A}x = \begin{bmatrix} T & -W \\ W & T \end{bmatrix} \begin{bmatrix} y \\ -z \end{bmatrix} = \begin{bmatrix} g \\ f \end{bmatrix} = b; \tag{3}$$

this real form can be regarded as a special class of generalized saddle point problems. Based on the preconditioning technique for generalized saddle point problems, Zhang et al. [27] proposed a block splitting (BS) preconditioner for the system (3). However, by observing the residual between the original coefficient matrix and the BS preconditioner, a limitation that the diagonal blocks tend to zero while the nonzero off-diagonal block becomes unbounded as α approaches 0 needs to overcome.

Inspired by this, by employing the relaxation techniques used in [14, 25], in this paper, we construct an improved block (IB) splitting preconditioner for the real block two-by-two system of linear (3). Algorithm 1 below shows that the IB preconditioner has almost equal computational cost comparing to the BS preconditioner in the inner iteration. In addition, theoretical analysis shows that the corresponding IB iteration method will converge to the unique solution of (3) under some suitable conditions, and the preconditioned matrix has an eigenvalue 1 with algebraic multiplicity at least n. We will also investigate the structure of the eigenvectors and the impact upon the corresponding Krylov subspace method in this paper.

The remainder of this work is organized as follows. In Section 2, we present the new preconditioner based on the deteriorated positive-definite and skew-Hermitian splitting (DPSS) preconditioning and the relaxation technique. In Section 3, we analyze the convergence property of the corresponding iteration method. Then in Section 4, we will investigate the eigen info of the preconditioned matrix and the impact upon the convergence of the corresponding Krylov subspace method. In Section 5, some numerical experiments are carried out to validate the effectiveness of the new preconditioner. Finally in Section 6, some conclusions are given.

2 An improved block splitting iteration method and preconditioner

Since the matrix $T \in \mathbb{R}^{n\times n}$ is symmetric positive definite, the coefficient matrix $\mathscr{A}$ of the system (3) has the following splitting:

$$\mathscr{A} = M + N, \tag{4}$$

where $M=\begin{bmatrix} T & -W \\ W & 0 \end{bmatrix}$ and $N=\begin{bmatrix} 0 & 0 \\ 0 & T \end{bmatrix}$. Then based on the preconditioning technique proposed in [4, 9, 20], we get the following block splitting preconditioner for the block two-by-two system (3):

$$\bar{\mathscr{P}}_1 = \frac{1}{2\alpha}(\alpha I + M)(\alpha I + N). \tag{5}$$

The factor 1/2 has no effect on the preconditioner and thus we omit it for convenience. Then the preconditioner $\bar{\mathscr{P}}_1$ can be rewritten as

$$\begin{aligned} \mathscr{P}_1 &= \frac{1}{\alpha}(\alpha I + M)(\alpha I + N) \\ &= \frac{1}{\alpha}\begin{bmatrix} \alpha I + T & -W \\ W & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & 0 \\ 0 & \alpha I + T \end{bmatrix} \\ &= \begin{bmatrix} \alpha I + T & -W(I + \frac{1}{\alpha}T) \\ W & \alpha I + T \end{bmatrix}. \end{aligned} \tag{6}$$

The difference between $\mathscr{P}_1$ and $\mathscr{A}$ is given by

$$\mathscr{R}_1 = \mathscr{P}_1 - \mathscr{A} = \begin{bmatrix} \alpha I & -\frac{1}{\alpha}WT \\ 0 & \alpha I \end{bmatrix}. \tag{7}$$

Two important properties about the difference $\mathscr{R}_1$ should come to our attention. First, when α tends to 0_+, the weight of the two diagonal blocks in $\mathscr{R}_1$ also approaches 0, while the weight of the nonzero off-diagonal block approaches ∞. Hence, the choice of α needs to be balanced. On the other hand, as a preconditioner, we hope it is as close as possible to the coefficient matrix $\mathscr{A}$, that is, the difference matrix is expected to approach zero sufficiently. With these in mind and motivated by the idea of the relaxed preconditioners [14, 15, 25, 26], an improved block (IB) splitting preconditioner can therefore be derived as follows:

$$\mathscr{P}_{IB} = \frac{1}{\alpha}C_1C_2 = \frac{1}{\alpha}\begin{bmatrix} T & -W \\ W & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & 0 \\ 0 & \beta I + T \end{bmatrix} = \begin{bmatrix} T & -\frac{1}{\alpha}W(\beta I + T) \\ W & \beta I + T \end{bmatrix}. \tag{8}$$

The difference between $\mathscr{P}_{IB}$ and $\mathscr{A}$ is

$$\mathscr{R}_{IB} = \mathscr{P}_{IB} - \mathscr{A} = \begin{bmatrix} 0 & (1-\frac{\beta}{\alpha})W - \frac{1}{\alpha}WT \\ 0 & \beta I \end{bmatrix}. \tag{9}$$

We can find that the (1,1)-block matrix in (7) turns to zero, and two iteration parameters α and β are introduced.

Remark 2.1 If $\alpha \to \infty$ and $\beta \to 0$, the difference matrix $\mathscr{R}_{IB}$ will approach zero more sufficiently. Consequently, the preconditioner $\mathscr{P}_{IB}$ can be expected to perform much better for solving the complex symmetric indefinite linear systems.

Remark 2.2 If we take the iteration parameter $\beta = 0$, then the difference between $\mathscr{P}_{IB}$ and $\mathscr{A}$ will be

$$\mathscr{R}_{IB} = \begin{bmatrix} 0 & W - \frac{1}{\alpha}WT \\ 0 & 0 \end{bmatrix}.$$

It seems that the difference matrix has three sub-matrices equal to zero, i.e., the IB preconditioner here is much closer to the coefficient matrix. But unfortunately, the analysis of the spectrum of the preconditioned matrix and the numerical experiments in the following sections show that we can not always obtain better numerical results when we take $\beta = 0$.

Remark 2.3 It should be noted that if we take $\alpha = \beta$, the IB preconditioner will reduce to

$$\mathscr{P}_{IB} = \begin{bmatrix} T & -W(I + \frac{1}{\alpha}T) \\ W & \alpha I + T \end{bmatrix}.$$

The difference between $\mathscr{P}_{IB}$ and $\mathscr{A}$ will be

$$\mathscr{R}_{IB} = \begin{bmatrix} 0 & -\frac{1}{\alpha}WT \\ 0 & \alpha I \end{bmatrix}.$$

The preconditioner here is very likely to the BS preconditioner proposed in [27].

It should come to our attention that the IB preconditioner no longer relates to an alternating direction iteration method, but it is of no effect on the behavior that $\mathscr{P}_{IB}$ is used as a preconditioner. In fact, the preconditioner $\mathscr{P}_{IB}$ can be obtained by the following splitting of the coefficient matrix $\mathscr{A}$:

$$\mathscr{A} = \mathscr{P}_{IB} - \mathscr{R}_{IB} = \begin{bmatrix} T & -\frac{1}{\alpha}W(\beta I + T) \\ W & \beta I + T \end{bmatrix} - \begin{bmatrix} 0 & (1 - \frac{\beta}{\alpha})W - \frac{1}{\alpha}WT \\ 0 & \beta I \end{bmatrix}. \quad (10)$$

Then an iterative method based on this splitting can be obtained, which is defined as follows:

The IB iteration method Let α, β be given positive constants. Given an initial guess $x^{(0)}$, for $k = 0, 1, 2, \cdots$ until $\{x^{(k)}\}$ converges, compute

$$\begin{bmatrix} T & -\frac{1}{\alpha}W(\beta I + T) \\ W & \beta I + T \end{bmatrix} x^{(k+1)} = \begin{bmatrix} 0 & (1 - \frac{\beta}{\alpha})W - \frac{1}{\alpha}WT \\ 0 & \beta I \end{bmatrix} x^{(k)} + b,$$

which can also be written as a fixed-point iteration:

$$x^{(k+1)} = \Gamma x^{(k)} + c, \quad (11)$$

where, $\Gamma = \mathscr{P}_{IB}^{-1}\mathscr{R}_{IB}$ is the iteration matrix, $c = \mathscr{P}_{IB}^{-1}b$.

Theoretically, the IB iteration method converges to the unique solution of the system (3) for arbitrary initial guess $x^{(0)}$ if and only if the spectral radius of the iteration matrix is less than 1, i.e., $\rho(\Gamma) < 1$.

3 Convergence analysis of the iteration method

In this section, we will discuss the convergence properties of the IB iteration method.

Lemma 3.1 *Let $T \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $W \in \mathbb{R}^{n\times n}$ be symmetric indefinite, then the eigenvalues of the matrix $WT^{-1}WT$ are all real and nonnegative.*

Proof Since the matrix T is symmetric positive definite, then according to [16], there exists a symmetric positive definite matrix X such that $T = X^2$. Therefore, we have

$$XWT^{-1}WTX^{-1} = XWT^{-1}WX = X^TWT^{-1}WX. \tag{12}$$

Then the matrix $WT^{-1}WT$ is similar to $XWT^{-1}WX$. Since $W \in \mathbb{R}^{n\times n}$ is symmetric indefinite and $T \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then the matrix $XWT^{-1}WX$ is symmetric positive semi-definite. Therefore, the eigenvalues of the matrix $WT^{-1}WT$ are all real and nonnegative. □

Theorem 3.1 *Let $T \in \mathbb{R}^{n\times n}$ be symmetric positive definite and $W \in \mathbb{R}^{n\times n}$ be symmetric indefinite, α, β be positive constants. Then the iteration matrix Γ of the IB iteration method has an eigenvalue 0 with multiplicity n. The remaining n eigenvalues μ satisfy*

$$\mu = \frac{\alpha\beta + (\beta-\alpha)a_2 + a_3}{\alpha\beta + \alpha a_1 + \beta a_2 + a_3}, \tag{13}$$

where $a_1 = \dfrac{u^*Tu}{u^*u}$, $a_2 = \dfrac{u^*WT^{-1}Wu}{u^*u}$, $a_3 = \dfrac{u^*WT^{-1}WTu}{u^*u}$.

Proof It is not difficult to obtain that the matrix C_1 in (8) has the block-triangular factorization

$$C_1 = \begin{bmatrix} I & -\frac{1}{\alpha}W \\ 0 & I \end{bmatrix}\begin{bmatrix} T+\frac{1}{\alpha}W^2 & 0 \\ 0 & \alpha I \end{bmatrix}\begin{bmatrix} I & 0 \\ \frac{1}{\alpha}W & I \end{bmatrix}. \tag{14}$$

Then the explicit expression of $\mathscr{P}_{IB}^{-1}$ can be given by

$$\begin{aligned}\mathscr{P}_{IB}^{-1} &= \alpha C_2^{-1}C_1^{-1} \\ &= \begin{bmatrix} (T+\frac{1}{\alpha}W^2)^{-1} & \frac{1}{\alpha}(T+\frac{1}{\alpha}W^2)^{-1}W \\ -(\beta I+T)^{-1}W(T+\frac{1}{\alpha}W^2)^{-1} & (\beta I+T)^{-1}[-\frac{1}{\alpha}W(T+\frac{1}{\alpha}W^2)^{-1}W+I] \end{bmatrix}.\end{aligned} \tag{15}$$

Consequently, we can obtain

$$\begin{aligned}\mathscr{P}_{IB}^{-1}\mathscr{R}_{IB} &= \begin{bmatrix} (T+\frac{1}{\alpha}W^2)^{-1} & \frac{1}{\alpha}(T+\frac{1}{\alpha}W^2)^{-1}W \\ -(\beta I+T)^{-1}W(T+\frac{1}{\alpha}W^2)^{-1} & (\beta I+T)^{-1}[-\frac{1}{\alpha}W(T+\frac{1}{\alpha}W^2)^{-1}W+I] \end{bmatrix} \\ &\quad\times \begin{bmatrix} 0 & (1-\frac{\beta}{\alpha})W-\frac{1}{\alpha}WT \\ 0 & \beta I \end{bmatrix} \\ &= \begin{bmatrix} 0 & (T+\frac{1}{\alpha}W^2)^{-1}(-\frac{1}{\alpha}WT+W) \\ 0 & (\beta I+T)^{-1}[-W(T+\frac{1}{\alpha}W^2)^{-1}(W-\frac{1}{\alpha}WT)+\beta I] \end{bmatrix} \\ &= \begin{bmatrix} 0 & (T+\frac{1}{\alpha}W^2)^{-1}(-\frac{1}{\alpha}WT+W) \\ 0 & (\beta I+T)^{-1}[-W(T+\frac{1}{\alpha}W^2)^{-1}(W-\frac{1}{\alpha}WT)+\beta I] \end{bmatrix} \\ &= \begin{bmatrix} 0 & S_1 \\ 0 & S_2 \end{bmatrix}.\end{aligned}$$

Here, $S_1 = (T+\frac{1}{\alpha}W^2)^{-1}(-\frac{1}{\alpha}WT+W)$, $S_2 = (\beta I+T)^{-1}[-W(T+\frac{1}{\alpha}W^2)^{-1}(W-\frac{1}{\alpha}WT)+\beta I]$. Therefore, the iteration matrix $\Gamma = \mathscr{P}_{IB}^{-1}\mathscr{R}_{IB}$ has eigenvalue 0 with multiplicity n.

On the other hand, we have

$$\begin{aligned}S_2 &= (\beta I+T)^{-1}[-W(T+\tfrac{1}{\alpha}W^2)^{-1}(W-\tfrac{1}{\alpha}WT)+\beta I] \\ &= (\beta I+T)^{-1}[(\alpha I+WT^{-1}W)^{-1}WT^{-1}W(T-\alpha I)+\beta I] \\ &= (\beta I+T)^{-1}(\alpha I+WT^{-1}W)^{-1}[\alpha\beta I+(\beta-\alpha)WT^{-1}W+WT^{-1}WT].\end{aligned}$$

Assume that (μ, u) is an eigenpair of S_2, then the remaining eigenvalues of $\mathscr{P}_{IB}^{-1}\mathscr{R}_{IB}$ satisfy the generalized eigenvalue problem:

$$[\alpha\beta I+(\beta-\alpha)WT^{-1}W+WT^{-1}WT]u = \mu(\alpha I+WT^{-1}W)(\beta I+T)u. \quad (16)$$

Multiplying the (16) from left by $\dfrac{u^*}{u^*u}$, we have

$$\alpha\beta+(\beta-\alpha)a_2+a_3 = \mu(\alpha\beta+\alpha a_1+\beta a_2+a_3). \quad (17)$$

Thus, the remaining eigenvalues of the iteration matrix Γ are of the form

$$\mu = \frac{\alpha\beta+(\beta-\alpha)a_2+a_3}{\alpha\beta+\alpha a_1+\beta a_2+a_3}.$$

Since the matrix T is symmetric positive definite, then $a_1 > 0$. By Lemma 3.1, we can also obtain $a_2 \geq 0$, $a_3 \geq 0$. Therefore, the denominator of μ is not equal to 0.

The proof of this theorem is completed. □

Theorem 3.2 *Suppose the conditions of Theorem 3.1 are satisfied. If $a_1 < a_2$, then the IB iteration method for the linear system (3) is convergent for $\beta > \frac{\alpha(a_2-a_1)-2a_3}{2(\alpha+a_2)}$. If $a_1 > a_2$, then the IB iteration method is convergent for $\forall\alpha, \beta > 0$.*

Proof From Theorem 3.1, we have

$$\rho(\Gamma) < 1 \Leftrightarrow |\mu| < 1 \Leftrightarrow |\frac{\alpha\beta + (\beta - \alpha)a_2 + a_3}{\alpha\beta + \alpha a_1 + \beta a_2 + a_3}| < 1. \tag{18}$$

After simply computation, (18) can be reduced to

$$\begin{cases} \beta a_2 - \alpha a_2 < \alpha a_1 + \beta a_2, \\ 2\alpha\beta + 2a_3 + 2\beta a_2 > \alpha(a_2 - a_1). \end{cases} \tag{19}$$

(i) The first inequality of (19) is equivalent to $a_1 > -a_2$. Since $a_1 > 0$, $a_2 \geq 0$, then this inequality holds true for $\forall \alpha, \beta > 0$.

(ii) For the second inequality of (19), if $a_1 < a_2$, this inequality holds true for $\beta > \frac{\alpha(a_2-a_1)-2a_3}{2(\alpha+a_2)}$; If $a_1 > a_2$, then the inequality holds true for $\forall \alpha, \beta > 0$.

Therefore, if $a_1 < a_2$, the IB iteration method for the linear system (3) is convergent for $\beta > \frac{\alpha(a_2-a_1)-2a_3}{2(\alpha+a_2)}$; if $a_1 > a_2$, then the IB iteration method is convergent for $\forall \alpha, \beta > 0$.

The proof of this theorem is completed. □

From (10) and (11), we can find that the system $\mathscr{A}x = b$ is equivalent to the linear system

$$(I - \Gamma)x = \left(\mathscr{P}_{IB}^{-1}\mathscr{A}\right)x = \mathscr{P}_{IB}^{-1}b = c. \tag{20}$$

This equivalent system can be solved by Krylov subspace methods (such as GMRES method). Therefore, the matrix $\mathscr{P}_{IB}$ can be seen as a left preconditioner for Krylov subspace methods, that is, the Krylov subspace methods are used to accelerate the convergence behavior of the IB iteration method.

4 Eigen information of the preconditioned matrix

In this section, we will analyze the spectral property of the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$, since the convergence behavior relates closely to the eigenvalue distribution of the preconditioned matrix. Based on Theorem 3.1, the following theorem describes the eigenvalue distribution of the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$.

Theorem 4.1 *Let the preconditioner $\mathscr{P}_{IB}$ be defined in (8), α, β be real positive constants. $T \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $W \in \mathbb{R}^{n\times n}$ is symmetric indefinite. Then the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ has an eigenvalue 1 with multiplicity at least n. The remaining nonunit eigenvalues λ of $\mathscr{P}_{IB}^{-1}\mathscr{A}$ are of the form $\frac{\alpha(a_1+a_2)}{\alpha\beta+\alpha a_1+\beta a_2+a_3}$, and these nonunit eigenvalues satisfy*

$$0 < \lambda < \frac{\alpha(a_1 + a_2)}{\alpha a_1 + \beta a_2}.$$

Specially, if the parameters satisfy $\alpha = \beta$, then

$$0 < \lambda < 1.$$

Proof Since $\mathscr{P}_{IB}^{-1}\mathscr{A} = I - \mathscr{P}_{IB}^{-1}\mathscr{R}_{IB}$, then from (16), the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ has the form

$$\mathscr{P}_{IB}^{-1}\mathscr{A} = \begin{bmatrix} I_n & -S_1 \\ 0 & I_n - S_2 \end{bmatrix}. \tag{21}$$

Therefore, $\mathscr{P}_{IB}^{-1}\mathscr{A}$ has an eigenvalue 1 with multiplicity at least n. The remaining n nonunit eigenvalues λ are the same with those of the matrix $I_n - S_2$. Thus,

$$\lambda = 1 - \frac{\alpha\beta + (\beta-\alpha)a_2 + a_3}{\alpha\beta + \alpha a_1 + \beta a_2 + a_3} = \frac{\alpha(a_1 + a_2)}{\alpha\beta + \alpha a_1 + \beta a_2 + a_3} < \frac{\alpha(a_1 + a_2)}{\alpha a_1 + \beta a_2}. \tag{22}$$

Since $a_1 > 0$, $a_2 \geq 0$ and $a_3 \geq 0$, then

$$0 < \lambda < \frac{\alpha(a_1 + a_2)}{\alpha a_1 + \beta a_2}.$$

It can be easily obtained that if $\alpha = \beta$, then $0 < \lambda < 1$.

The proof of this theorem is completed. □

Remark 4.1 Let $0 < \alpha_0 < +\infty$, $0 < \beta_0 < +\infty$, it follows that

(i) if the parameters $\alpha \to 0$, $\beta \to \beta_0$, then $\lambda \to 0$;
(ii) if the parameters $\alpha \to +\infty$, $\beta \to +\infty$, then $\lambda \to 0$;
(iii) if the parameters $\alpha \to \alpha_0$, $\beta \to +\infty$, then $\lambda \to 0$;
(iv) if the parameters $\alpha \to +\infty$, $\beta \to \beta_0$, then $\lambda \to \frac{a_1+a_2}{\beta_0+a_1}$, and the interval $(0, \frac{\alpha(a_1+a_2)}{\alpha a_1+\beta a_2}) \to (0, 1)$.

The convergence rate of the corresponding Krylov subspace methods not only depends on the eigenvalue distribution of the preconditioned matrix but also relies on the structure of the corresponding linearly independent eigenvectors.

Theorem 4.2 *Let the preconditioner $\mathscr{P}_{IB}$ be defined in (8), α, β be real positive constants. Then the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ has $n+i+j$ linearly independent eigenvectors which comprise*

(a) *n eigenvectors of the form $\begin{bmatrix} p_l^{(1)} \\ 0 \end{bmatrix}$ $(l = 1, 2, \cdots, n)$ that are associated with the eigenvalue 1, where $p_l^{(1)}$ $(l = 1, 2, \cdots, n)$ are arbitrary linearly independent;*
(b) *i $(1 \leq i \leq n)$ eigenvectors of the form $\begin{bmatrix} p_l^{(2)} \\ q_l^{(2)} \end{bmatrix}$ $(1 \leq l \leq i)$ that are associated with the eigenvalue 1, where $p_l^{(2)}$ are arbitrary vectors, $q_l^{(2)}$ satisfy $(T + WT^{-1}W)q_l^{(2)} = (\alpha I + WT^{-1}W)(\beta I + T)q_l^{(2)}$,and $(T - \alpha I)q_l^{(2)} = 0$;*

(c) *j $(1 \le j \le n)$ eigenvectors of the form* $\begin{bmatrix} p_l^{(3)} \\ q_l^{(3)} \end{bmatrix}$ $(1 \le l \le j)$ *that correspond to the nonunit eigenvalues, where* $q_l^{(3)} \neq 0$, $(T + WT^{-1}W)q_l^{(3)} = \lambda(\alpha I + WT^{-1}W)(\beta I + T)q_l^{(3)}$, *and* $p_l^{(3)} = \dfrac{(\alpha T + W^2)^{-1}W(T - \alpha I)}{\lambda - 1} q_l^{(3)}$.

Proof Let λ be an eigenvalue of the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ and $\begin{bmatrix} p \\ q \end{bmatrix}$ is the corresponding eigenvector. Then from (21), we have

$$\begin{bmatrix} I_n & -S_1 \\ 0 & I_n - S_2 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \lambda \begin{bmatrix} p \\ q \end{bmatrix}, \tag{23}$$

since

$$\begin{aligned} I_n - S_2 &= I_n - (\beta I + T)^{-1}(\alpha I + WT^{-1}W)^{-1}[\alpha\beta I + (\beta - \alpha)WT^{-1}W + WT^{-1}WT] \\ &= \alpha(\beta I + T)^{-1}(\alpha I + WT^{-1}W)^{-1}(T + WT^{-1}W), \end{aligned} \tag{24}$$

then (23) can be rewritten as

$$\begin{cases} (1 - \lambda)p = -(\alpha T + W^2)^{-1}W(T - \alpha I)q, \\ \alpha(T + WT^{-1}W)q = \lambda(\alpha I + WT^{-1}W)(\beta I + T)q. \end{cases} \tag{25}$$

From Lemma 3.1, we know that the matrix $WT^{-1}W$ is symmetric positive semidefinite. Then the matrices $T + WT^{-1}W$, $\alpha I + WT^{-1}W$, and $\beta I + T$ are symmetric positive definite for $\alpha, \beta > 0$.

(i) When $q = 0$, it follows from the first equation of (25) that $\lambda = 1$. Otherwise, we will obtain $p = 0$, which contradicts with $\begin{bmatrix} p \\ q \end{bmatrix}$ being a nonzero eigenvector. Therefore, there are n eigenvectors of the form $\begin{bmatrix} p_l^{(1)} \\ 0 \end{bmatrix}$ correspond to the eigenvalue 1, where $p_l^{(1)}$ $(l = 1, 2, \cdots, n)$ are arbitrary linearly independent.

(ii) When $q \neq 0$ and satisfies the second equation of (25), we will discuss the eigenvectors as follows:

First, we consider the case if α is the eigenvalue of the matrix T, which means the matrix $T - \alpha I$ is singular. When $q \in \ker(T - \alpha I)$, then we can obtain $(\lambda - 1)p = 0$ from the first equation of (25), where $\ker(\cdot)$ denotes the null space of the corresponding matrix. Once $\lambda = 1$, there will be i $(0 \le i \le n)$ linearly independent eigenvectors of the form $\begin{bmatrix} p_l^{(2)} \\ q_l^{(2)} \end{bmatrix}$ $(l = 1, 2, \cdots, i)$ that are associated with the eigenvalue 1, where $p_l^{(2)}$ are arbitrary vectors, and

$q_l^{(2)}$ satisfy $\alpha(T+WT^{-1}W)q_l^{(2)} = (\alpha I + WT^{-1}W)(\beta I + T)q_l^{(2)}$,and $(T-\alpha I)q_l^{(2)} = 0$. Otherwise, when $(T-\alpha I)q \neq 0$, we can obtain that $\lambda \neq 1$ and $p \neq 0$.

Next, if α is not the eigenvalue of the symmetric positive definite matrix T, the matrix $T-\alpha I$ is nonsingular, then it follows from the first equation of (25) that $\lambda \neq 1$ and $p = \frac{(\alpha T + W^2)^{-1}W(T-\alpha I)}{\lambda - 1}q \neq 0$.

Therefore, in summery, there will be j $(0 \leq j \leq n)$ linearly independent eigenvectors of the form $\begin{bmatrix} p_l^{(3)} \\ q_l^{(3)} \end{bmatrix}$ $(l = 1, 2, \cdots, j)$ that correspond to eigenvalues $\lambda \neq 1$, where $q_l^{(3)} \neq 0$, $(T+WT^{-1}W)q_l^{(3)} = \lambda(\alpha I + WT^{-1}W)(\beta I + T)q_l^{(3)}$, and $p_l^{(3)} = \frac{(\alpha T + W^2)^{-1}W(T-\alpha I)}{\lambda - 1}q_l^{(3)}$.

Finally, we validate the linear independence of these $n+i+j$ eigenvectors. Let $k^{(1)} = [k_1^{(1)}, k_2^{(1)}, \cdots, k_n^{(1)}]^T$, $k^{(2)} = [k_1^{(2)}, k_2^{(2)}, \cdots, k_i^{(2)}]^T$ and $k^{(3)} = [k_1^{(3)}, k_2^{(3)}, \cdots, k_j^{(3)}]^T$ be three vectors with $0 \leq i, j \leq n$. It is left to show that

$$\begin{bmatrix} p_1^{(1)} & \cdots & p_n^{(1)} \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} k_1^{(1)} \\ \vdots \\ k_n^{(1)} \end{bmatrix} + \begin{bmatrix} p_1^{(2)} & \cdots & p_i^{(2)} \\ q_1^{(2)} & \cdots & q_i^{(2)} \end{bmatrix} \begin{bmatrix} k_1^{(2)} \\ \vdots \\ k_i^{(2)} \end{bmatrix} + \begin{bmatrix} p_1^{(3)} & \cdots & p_j^{(3)} \\ q_1^{(3)} & \cdots & q_j^{(3)} \end{bmatrix} \begin{bmatrix} k_1^{(3)} \\ \vdots \\ k_j^{(3)} \end{bmatrix} = 0 \tag{26}$$

holds if and only if the coefficient vectors $k^{(1)}$, $k^{(2)}$, and $k^{(3)}$ are zero. Here $[(p_s^{(1)})^T, 0]^T$ $(s = 1, \ldots, n)$ denotes the sth eigenvector that corresponds to the eigenvalue 1 for the case (a), while $[(p_t^{(2)})^T, (q_t^{(2)})^T]^T$ $(t = 1, \ldots, i)$ is the tth eigenvector associated with the eigenvalue 1 for the case (b), and $[(p_h^{(3)})^T, (q_h^{(3)})^T]^T$ is the hth eigenvector associated with the nonunit eigenvalues for $h = 1, \ldots, j$. By multiplying $\mathscr{P}_{IB}^{-1}\mathscr{A}$ on both sides of (26), we have

$$\begin{bmatrix} p_1^{(1)} & \cdots & p_n^{(1)} \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} k_1^{(1)} \\ \vdots \\ k_n^{(1)} \end{bmatrix} + \begin{bmatrix} p_1^{(2)} & \cdots & p_i^{(2)} \\ q_1^{(2)} & \cdots & q_i^{(2)} \end{bmatrix} \begin{bmatrix} k_1^{(2)} \\ \vdots \\ k_i^{(2)} \end{bmatrix} \begin{bmatrix} p_1^{(3)} & \cdots & p_j^{(3)} \\ q_1^{(3)} & \cdots & q_j^{(3)} \end{bmatrix} \begin{bmatrix} \lambda_1 k_1^{(3)} \\ \vdots \\ \lambda_j k_j^{(3)} \end{bmatrix} = 0 \tag{27}$$

Subtracting (27) from (26), we obtain

$$\begin{bmatrix} p_1^{(3)} & \cdots & p_j^{(3)} \\ q_1^{(3)} & \cdots & q_j^{(3)} \end{bmatrix} \begin{bmatrix} (\lambda_1 - 1)k_1^{(3)} \\ \vdots \\ (\lambda_j - 1)k_j^{(3)} \end{bmatrix} = 0.$$

Since $\lambda_h \neq 1$ $(h = 1, \ldots, j)$ and the column vectors of $\begin{bmatrix} p_1^{(3)} & \cdots & p_j^{(3)} \\ q_1^{(3)} & \cdots & q_j^{(3)} \end{bmatrix}$ are linearly independent, then we have $k_h^{(3)} = 0$ $(h = 1, \ldots, j)$. Since the vectors $q_t^{(2)}$ $(t = 1, \ldots, i)$ are also linearly independent, then we have $k_t^{(2)} = 0$ $(t = 1, \ldots, i)$. Thus, the (27) reduces to

$$\begin{bmatrix} p_1^{(1)} & \cdots & p_n^{(1)} \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} k_1^{(1)} \\ \vdots \\ k_n^{(1)} \end{bmatrix} = 0.$$

$\begin{bmatrix} p_1^{(1)} & \cdots & p_n^{(1)} \\ 0 & \cdots & 0 \end{bmatrix}$ are also linearly independent, then $k_s^{(1)} = 0$ $(s = 1, \ldots, n)$. Therefore, the $n + i + j$ eigenvectors are linearly independent.

The proof of this theorem is completed. □

The idea of preconditioning is trying to improve the spectral properties, such that the total number of iterations required to solve the system to within some tolerance will be indeed decreased. The iterative method employed will terminate when the degree of the minimal polynomial, or the dimension of the corresponding Krylov subspace, is achieved [23]. The next theorem provides detailed analysis to the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{\mathrm{IB}}^{-1}\mathscr{A}, b)$.

Theorem 4.3 *Let the preconditioner $\mathscr{P}_{IB}$ be defined in (8), then the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{IB}^{-1}\mathscr{A}, b)$ is at most $n + 1$. Specially, once the matrix $I_n - S_2$ has k $(1 \le k \le n)$ distinct eigenvalues θ_i $(1 \le i \le k)$, of respective multiplicity m_i, where $\sum_{i=1}^{k} m_i = n$, the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{IB}^{-1}\mathscr{A}, b)$ is at most $k + 1$.*

Proof From (21), the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ is a block upper triangular matrix of the following form:

$$\mathscr{P}_{IB}^{-1}\mathscr{A} = \begin{bmatrix} I_n & -S_1 \\ 0 & I_n - S_2 \end{bmatrix}.$$

Assume λ_i $(i = 1, \cdots, n)$ be the eigenvalues of the matrix $I_n - S_2$, then they are also the nonunit eigenvalues of the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$. According to the eigenvalue distribution described in Theorem 3.1, the characteristic polynomial of the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ is

$$\det(\mathscr{P}_{IB}^{-1}\mathscr{A} - \lambda I) = (\lambda - 1)^n \prod_{i=1}^{n} (\lambda - \lambda_i).$$

Expanding the polynomial $(\mathscr{P}_{IB}^{-1}\mathscr{A}-I)\prod_{i=1}^{n}(\mathscr{P}_{IB}^{-1}\mathscr{A}-\lambda_i I)$ of degree $n+1$, we have

$$\Phi(\mathscr{P}_{IB}^{-1}\mathscr{A}) = (\mathscr{P}_{IB}^{-1}\mathscr{A}-I)\prod_{i=1}^{n}(\mathscr{P}_{IB}^{-1}\mathscr{A}-\lambda_i I) = \begin{bmatrix} 0 & -S_1\prod_{i=1}^{n}((1-\lambda_i)I-S_2) \\ 0 & -S_2\prod_{i=1}^{n}((1-\lambda_i)I-S_2) \end{bmatrix}.$$

Since λ_i are the eigenvalues of matrix $I_n - S_2$, by the Cayley-Hamilton theorem, then

$$\prod_{i=1}^{n}((1-\lambda_i)I-S_2) = 0.$$

Therefore, the degree of the minimal polynomial of $\mathscr{P}_{IB}^{-1}\mathscr{A}$ is at most $n+1$. Consequently, the dimension of the corresponding Krylov subspace $\mathscr{K}(\mathscr{P}_{IB}^{-1}\mathscr{A}, b)$ is at most $n+1$.

Once the matrix $I_n - S_2$ has k ($1 \le k \le n$) distinct eigenvalues θ_i ($1 \le i \le k$) of respective multiplicity m_i. We write the characteristic polynomial of the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ as

$$\underbrace{(\mathscr{P}_{IB}^{-1}\mathscr{A}-I)^{n-1}\prod_{i=1}^{k}(\mathscr{P}_{IB}^{-1}\mathscr{A}-\theta_i I)^{m_i-1}}\ \underbrace{(\mathscr{P}_{IB}^{-1}\mathscr{A}-I)\prod_{i=1}^{k}(\mathscr{P}_{IB}^{-1}\mathscr{A}-\theta_i I)}.$$

Let $\Psi(\mathscr{P}_{IB}^{-1}\mathscr{A}) = (\mathscr{P}_{IB}^{-1}\mathscr{A}-I)\prod_{i=1}^{k}(\mathscr{P}_{IB}^{-1}\mathscr{A}-\theta_i I)$, then

$$\Psi = \begin{bmatrix} 0 & -S_1\prod_{i=1}^{k}((1-\theta_i)I-S_2) \\ 0 & -S_2\prod_{i=1}^{k}((1-\theta_i)I-S_2) \end{bmatrix}.$$

Since $\prod_{i=1}^{k}((1-\theta_i)I-S_2) = 0$, then $\Psi(\mathscr{P}_{IB}^{-1}\mathscr{A})$ is a zero matrix. Therefore, in this case, the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{\text{VDPSS}}^{-1}\mathscr{A}, b)$ is at most $k+1$.

The proof of this theorem is completed. □

Remark 4.2 Theorem 4.2 indicates that although the linear system (3) becomes doubled in size, the iteration steps will not increase compared with the original linear system (1), i.e., the iteration steps will not exceed $n+1$ when the IB-preconditioned GMRES method is used for solving the system (3). Sometimes, the termination will even occur in at most $k+1$ ($1 \le k \le n$) steps.

Now we shall touch upon some computational aspects of the preconditioner $\mathscr{P}_{IB}$. When applying the preconditioner $\mathscr{P}_{IB}$ within a Krylov subspace method, a linear system $\mathscr{P}_{IB}z = r$ needs to be solved at each step, where $z = [z_1^T, z_2^T]^T$, $r = [r_1^T, r_2^T]^T$, $z_1, z_2, r_1, r_2 \in \mathbb{R}^n$. If n is large, then computing $z = \mathscr{P}_{IB}^{-1}r$ directly is impractical for memory reason. Thus, we avert such ostensibly "feasible" approach. Based on the matrix factorization (8) and (15), we have

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \alpha \begin{bmatrix} \frac{1}{\alpha}I & 0 \\ 0 & (\beta I + T)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -\frac{1}{\alpha}W & I \end{bmatrix} \begin{bmatrix} (T + \frac{1}{\alpha}W^2)^{-1} & 0 \\ 0 & \frac{1}{\alpha}I \end{bmatrix} \begin{bmatrix} I & \frac{1}{\alpha}W \\ 0 & I \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \tag{28}$$

Hence, the following implementing process about the preconditioner $\mathscr{P}_{IB}$ can be derived:

Algorithm 1 For a given $r = [r_1^T, r_2^T]^T$, compute $z = [z_1^T, z_2^T]^T$ by (28) from the following steps:

(i) compute $d_1 = r_1 + \frac{1}{\alpha}Wr_2$;
(ii) solve $(T + \frac{1}{\alpha}W^2)z_1 = d_1$;
(iii) compute $d_2 = r_2 - Wz_1$, solve $(\beta I + T)z_2 = d_2$.

Remark 4.3 From Algorithm 1, we can find the IB preconditioner only needs real arithmetic in actual implementation. Meanwhile, suppose all matrices and vectors involved are dense, then in step (i), it requires $\mathscr{O}(n^2)$ flops. In step (ii), we need to solve a linear system of size n, which requires $\mathscr{O}(n^3)$ flops, and in step (iii), it requires $\mathscr{O}(n^2 + n^3)$ flops. Therefore, solving the linear system $\mathscr{P}_{IB}z = r$ needs $\mathscr{O}(n^2 + n^3)$ flops. In practical, the involved matrices are generally sparse, then the total cost will be reduced.

For DPSS preconditioner $\mathscr{P}_1$, we can also derive the computational process in Algorithm 2:

Algorithm 2 For a given $r = [r_1^T, r_2^T]^T$, compute $z = [z_1^T, z_2^T]^T$ from the following steps:

(i) compute $d_1 = r_1 + \frac{1}{\alpha}Wr_2$;
(ii) solve $(\alpha I + T + \frac{1}{\alpha}W^2)z_1 = d_1$;
(iii) compute $d_2 = r_2 - Wz_1$, solve $(\alpha I + T)z_2 = d_2$.

From Algorithm 1 and Algorithm 2, we can find that the IB preconditioner has almost equal computational cost comparing to the BS preconditioner [27] and DPSS preconditioner $\mathscr{P}_1$ in the inner iteration. Here we should mention that it is not necessary to compute the inverse of the matrices involved directly, especially the inverse of the matrices $T + \frac{1}{\alpha}W^2$ and $\alpha I + T + \frac{1}{\alpha}W^2$, since the cost could be very expensive. For each of these two algorithms, we only need to solve two symmetric positive definite subsystems (with the coefficient matrices $T + \frac{1}{\alpha}W^2$ and $\beta I + T$ for the preconditioner $\mathscr{P}_{IB}$, $\alpha I + T + \frac{1}{\alpha}W^2$ and $\alpha I + T$ for the preconditioner $\mathscr{P}_1$), respectively. Therefore, in inexact manner, we can employ the conjugate gradient (CG) method to solve the two sub-linear systems, or they can be solved exactly with the sparse Cholesky factorization.

5 Numerical experiments

In this section, we carry out some numerical experiments of complex symmetric linear systems to validate the effectiveness of the IB preconditioner $\mathscr{P}_{IB}$. For comparison, we also choose the HSS preconditioner and DPSS preconditioner $\mathscr{P}_1$ coupled with GMRES(30) [22]. Both of them are induced by alternating direction splitting iterative methods. The HSS preconditioner is defined as

$$\mathscr{P}_{HSS} = \frac{1}{\alpha}(\alpha I + H)(\alpha I + S) = \begin{bmatrix} \alpha I + T & -(I + \frac{1}{\alpha}T)W \\ (I + \frac{1}{\alpha}T)W & \alpha I + T \end{bmatrix},$$

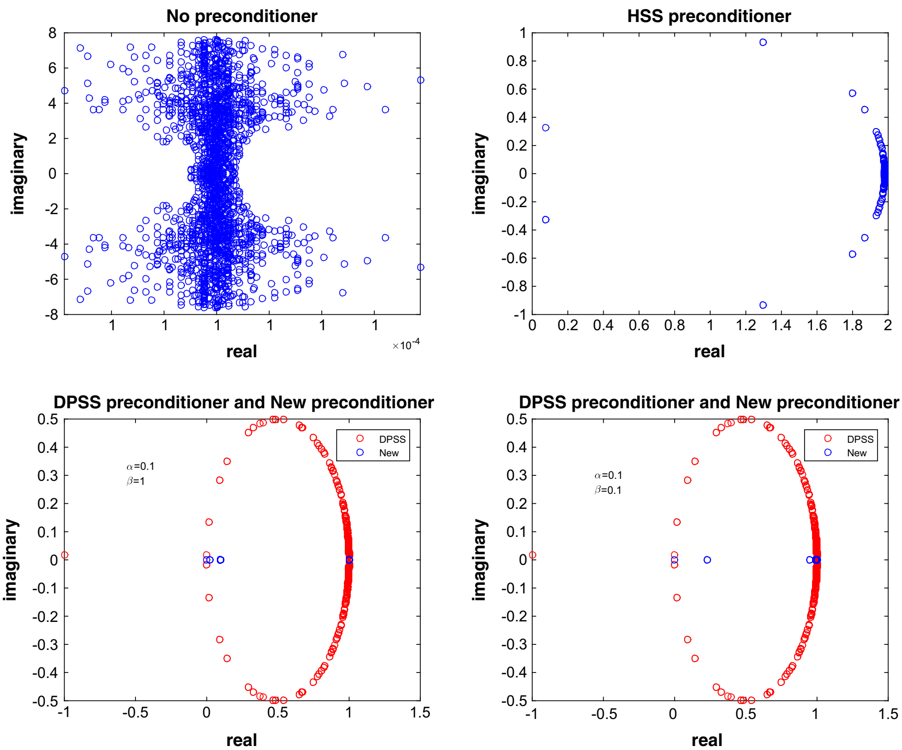

Fig. 1 The eigenvalue distributions of original matrix $\mathscr{A}$ and preconditioned matrices $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, $\mathscr{P}_1^{-1}\mathscr{A}$, and $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for the Example 5.1

here, $H = (\mathscr{A} + \mathscr{A}^T)/2$, $S = (\mathscr{A} - \mathscr{A}^T)/2$. All runs are implemented using double precision float point arithmetic in MATLAB (version R2015a). In actual computations, we set the initial guess to be $x_0 = \text{zeros}(2n, 1)$ and the stopping criterion $\frac{||r_i||_2}{||r_0||_2} \leq 10^{-6}$, where $r_i = b - \mathscr{A}x_i$. The iteration steps, CPU time, and the relative

Table 1 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.1

Preconditioner	k / m^2	10 / 16^2	20 / 32^2	30 / 64^2	40 / 128^2	50 / 256^2
HSS (α=0.01)	Iter	6	6	12	26	81
	CPU	0.0144	0.0748	0.5524	9.4723	87.0941
	RES	$4.4e-08$	$9.6e-08$	$3.4e-07$	$2.5e-07$	$8.9e-07$
DPSS (α=0.01)	Iter	6	6	12	5	81
	CPU	0.0143	0.0665	0.5459	5.3561	85.9648
	RES	$4.5e-08$	$1.0e-07$	$3.4e-07$	$2.8e-07$	$9.2e-07$
IB (α=0.01, β=1)	Iter	3	2	3	6	8
	CPU	0.0093	0.0376	0.2135	1.4186	9.8164
	RES	$1.7e-10$	$3.6e-07$	$6.3e-08$	$2.8e-06$	$1.1e-07$
IB (α=0.01, β=0.1)	Iter	3	3	3	4	8
	CPU	0.0110	0.0358	0.1988	1.1969	10.2638
	RES	$1.6e-09$	$1.6e-10$	$5.8e-07$	$2.8e-07$	$9.7e-07$
IB (α=0.01, β=0.01)	Iter	3	3	4	6	8
	CPU	0.0092	0.0496	0.1982	1.3776	8.9928
	RES	$2.1e-10$	$1.0e-10$	$2.9e-04$	$1.0e-08$	$1.0e-07$
DPSS (α=0.001)	Iter	4	4	5	8	94
	CPU	0.0203	0.0678	0.4367	9.4248	108.3084
	RES	$1.8e-08$	$1.6e-09$	$3.6e-07$	$9.6e-07$	$6.1e-07$
IB (α=0.001, β=1)	Iter	2	2	2	3	8
	CPU	0.0089	0.0271	0.1563	1.0293	5.0648
	RES	$2.1e-07$	$3.4e-09$	$1.2e-07$	$2.2e-07$	$1.6e-07$
IB (α=0.001, β=0.1)	Iter	2	2	3	5	4
	CPU	0.0096	0.0286	0.1564	1.0058	5.0438
	RES	$1.4e-11$	$3.3e-08$	$5.3e-10$	$9.6e-09$	$9.3e-08$
IB (α=0.001, β=0.01)	Iter	3	2	3	4	4
	CPU	0.0091	0.0270	0.1583	1.0091	4.9531
	RES	$1.5e-11$	$3.0e-07$	$4.5e-09$	$8.8e-09$	$8.6e-07$
IB (α=0.01, β=0)	Iter	4	3	4	7	10
	CPU	0.1169	0.1919	2.8705	100.6862	1.8579
	RES	$4.3e-12$	$1.8e-07$	$7.1e-07$	$2.2e-09$	$8.0e-07$

residual error are denoted by Iter, CPU (in seconds) and RES, respectively. The maximum number of iteration steps allowed is set to 1000. The symbol "–" in these tables indicates that the corresponding method fails to reach the required accuracy within the prescribed number of restarts. The sub-linear systems arising from the application of the preconditioners are solved by direct methods. In MATLAB, this corresponds to computing the Cholesky or LU factorization in combination with AMD or column AMD reordering.

Example 5.1 We consider the following complex symmetric linear system [27]

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

where $T_m = \text{tridiag}(-1, 2 - 1)$ is a tridiagonal matrix with order m and k denoting the wavenumber. We choose the matrices $W = T_m \otimes I_m + I_m \otimes T_m - k^2h^2(I_m \otimes I_m)$ and $T = \sigma_2(I_m \otimes I_m)$, where $h = \frac{1}{m+1}$ and $\sigma_2 = 10^{-4}$. T is symmetric positive definite. In actual implementations, we set the right-hand side $b = \mathscr{A} * \text{ones}(2m^2, 1)$. Fig. 1 demonstrates the eigenvalue distributions of the original coefficient matrix $\mathscr{A}$, the HSS preconditioned matrix $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, the DPSS preconditioned matrix $\mathscr{P}_1^{-1}\mathscr{A}$, and the new preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for $m = 32$, $\alpha = 0.1, \beta = 1$, and

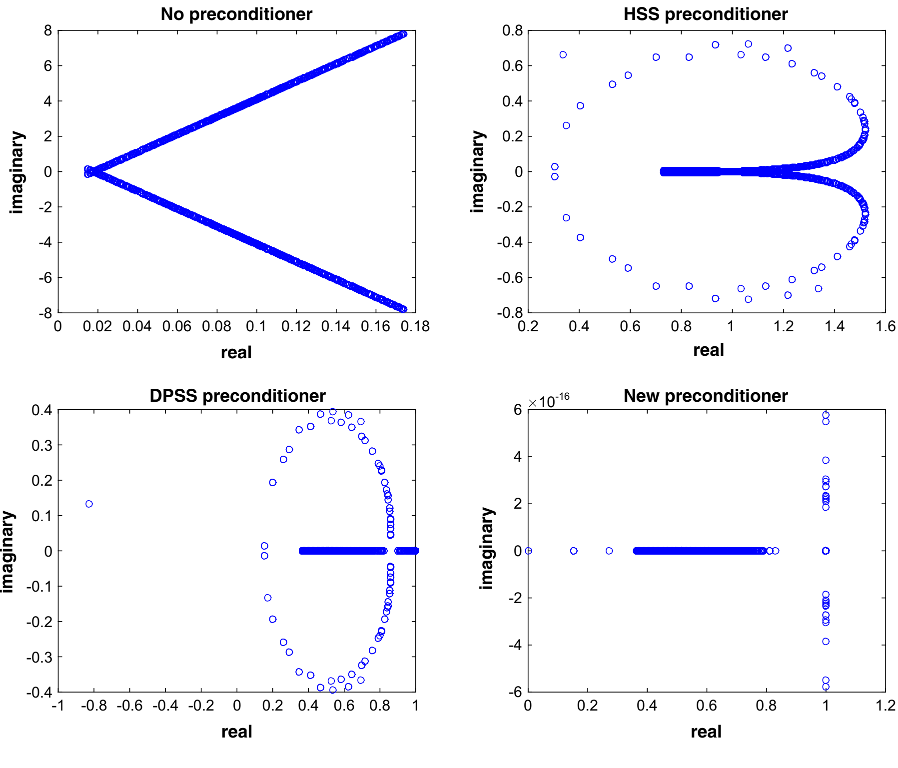

Fig. 2 The eigenvalue distributions of original matrix $\mathscr{A}$ and preconditioned matrices $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, $\mathscr{P}_1^{-1}\mathscr{A}$, and $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for the Example 5.2 with $\mathbf{M} = 5I_{m^2}$

$\alpha = 0.1$, $\beta = 0.1$. The numerical results for different choice of α, β, k, and m are listed in Table 1.

From Fig. 1 and Table 1, some conclusions can be obtained. In Fig. 1, we find that the eigenvalue distribution of the new preconditioned matrix $\mathscr{P}_{\mathrm{IB}}^{-1}\mathscr{A}$ is much better than that of the other preconditioned matrices. Moreover, the preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ has at least $1024 = 32^2$ eigenvalues 1, and the nonunit eigenvalues are also well distributed. This phenomenon confirms the theoretical analysis in Section 3.

Table 1 shows that by choosing different parameters, the IB preconditioner is superior to the DPSS and HSS preconditioners in terms of CPU time, iteration steps, and relative residual error, which confirms that the new preconditioner can virtually improve the convergence behavior of the GMRES(30). Additionally, if we choose suitable α and β, the numerical results of IB preconditioner are expected to be much better than the BS preconditioner (see the case of $\alpha = \beta = 0.01$). With the size of problem increasing, the iteration steps of IB preconditioner remain stable and computational costs are inexpensive (in most cases, the CPU times are less than 11 seconds , while those of the DPSS and HSS preconditioners go beyond 80 seconds). Besides, we can find that the IB preconditioner is more insensitive to the parameters α and

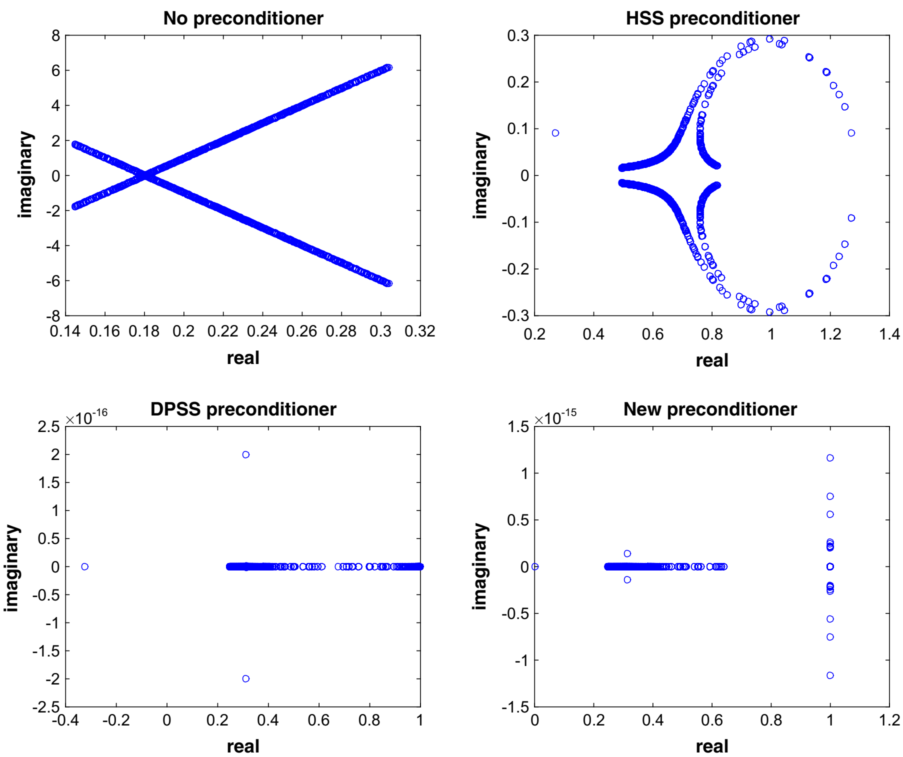

Fig. 3 The eigenvalue distributions of original matrix $\mathscr{A}$ and preconditioned matrices $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, $\mathscr{P}_{1}^{-1}\mathscr{A}$, and $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for the Example 5.2 with $\mathbf{M} = 50I_{m^2}$

β than the DPSS and HSS preconditioners. As a consequence, our proposed preconditioner is more effective and practical for solving the complex symmetric linear system (3). But we can also observe from the last row of Table 1 that if we choose $\beta = 0$, the numerical results will not always be absolutely excellent; see the CPU time when $k = 40$, $m = 128$.

Example 5.2 We consider the following complex symmetric linear system [8, 24]

$$[(-\omega^2\mathbf{M} + \mathbf{K}) + i(\omega C_{\mathbf{V}} + C_{\mathbf{H}})]x = b.$$

where $\mathbf{M}$ and $\mathbf{K}$ are the inertia and stiffness matrices and $C_{\mathbf{V}}$ and $C_{\mathbf{H}}$ are the viscous and hysteretic damping matrices, respectively. ω is the driving circular frequency. $\mathbf{K} = I_m \otimes V_m + V_m \otimes I_m$, $V_m = h^{-2}\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m}$ is a tridiagonal matrix, $h = \frac{1}{m+1}$, $C_{\mathbf{V}} = \frac{1}{2}\mathbf{M}$, $C_{\mathbf{H}} = \mu\mathbf{K}$ with μ being a damping coefficient.

We choose the matrices $W = h^2(-\omega^2\mathbf{M} + \mathbf{K})$ and $T = h^2(\omega C_{\mathbf{V}} + C_{\mathbf{H}})$ and set $\omega = 2\pi$ and $\mu = 0.02$. For $\mathbf{M} = 5I_{m^2}, 10I_{m^2}, 20I_{m^2}, 30I_{m^2}, 40I_{m^2}, 50I_{m^2}, 70I_{m^2}$, $90I_{m^2}$, and $100I_{m^2}$, we can easily show that the matrix W is symmetric indefinite and

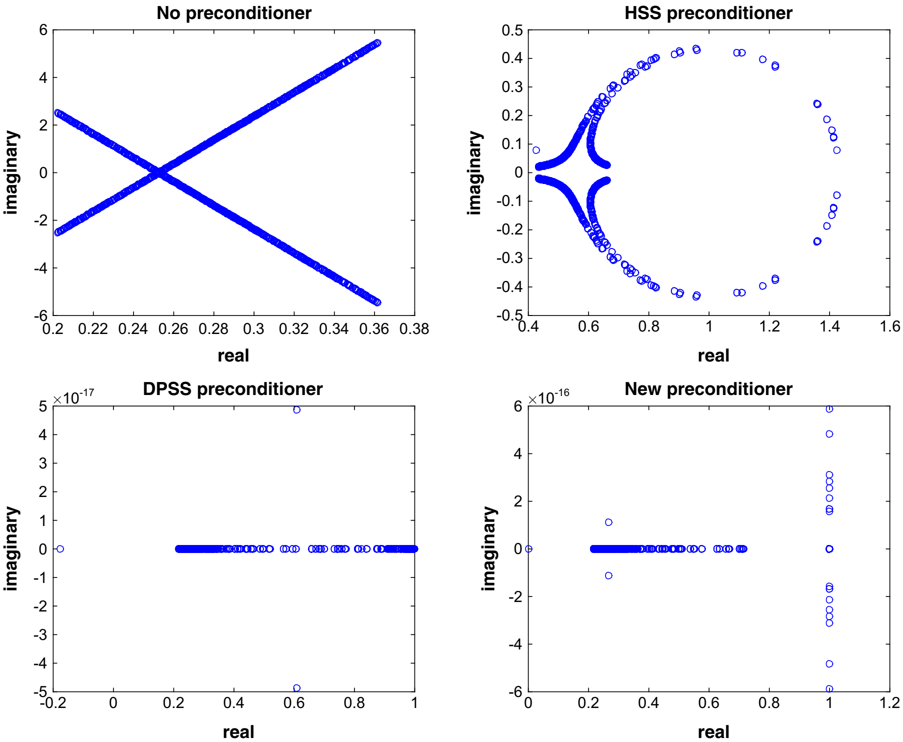

Fig. 4 The eigenvalue distributions of original matrix $\mathscr{A}$ and preconditioned matrices $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, $\mathscr{P}_1^{-1}\mathscr{A}$, and $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for the Example 5.2 with $\mathbf{M} = 70I_{m^2}$

Table 2 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.2 with $\alpha = 0.01$, $\beta = 0.1$

$\mathbf{M} \rightarrow$		$5I_{m^2}$	$10I_{m^2}$	$20I_{m^2}$	$30I_{m^2}$	$40I_{m^2}$	$50I_{m^2}$	$70I_{m^2}$	$90I_{m^2}$	$100I_{m^2}$
HSS $_{(\alpha=0.01)}$	Iter	15	14	18	19	19	25	13	27	25
	CPU	0.3456	0.3947	0.5451	0.4873	0.4032	0.5917	0.9593	1.3282	1.7656
	RES	$6.7e-07$	$5.2e-08$	$4.5e-07$	$6.8e-07$	$9.7e-07$	$5.5e-07$	$9.6e-07$	$7.9e-07$	$8.3e-07$
DPSS $_{(\alpha=0.01)}$	Iter	17	14	14	14	15	15	16	17	19
	CPU	0.3272	0.3388	0.2810	0.2843	0.2939	0.3089	0.3409	0.3361	0.4721
	RES	$5.0e-07$	$6.8e-07$	$6.4e-07$	$6.2e-07$	$3.6e-07$	$5.6e-07$	$7.2e-07$	$8.7e-07$	$5.3e-07$
IB $_{(\alpha=0.01,\beta=0.1)}$	Iter	7	7	9	9	10	12	13	14	16
	CPU	0.1543	0.1524	0.2158	0.2271	0.2214	0.2395	0.2645	0.3101	0.3428
	RES	$7.2e-7$	$5.5e-07$	$7.5e-07$	$7.2e-07$	$5.9e-07$	$2.6e-07$	$6.1e-07$	$8.1e-07$	$5.4e-07$

Table 3 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.2 with $\alpha = 0.1$, $\beta = 0.01$

$\mathbf{M} \rightarrow$		$5I_{m^2}$	$10I_{m^2}$	$20I_{m^2}$	$30I_{m^2}$	$40I_{m^2}$	$50I_{m^2}$	$70I_{m^2}$	$90I_{m^2}$	$100I_{m^2}$
HSS $(\alpha=0.1)$	Iter	17	12	8	6	8	10	13	17	19
	CPU	0.4637	0.3000	0.1849	0.2402	0.1881	0.2596	0.2880	0.4381	0.4003
	RES	$4.8e-07$	$9.2e-08$	$2.7e-07$	$8.2e-07$	$6.1e-07$	$7.3e-07$	$8.0e-07$	$6.8e-07$	$6.5e-07$
DPSS $(\alpha=0.1)$	Iter	18	14	10	9	10	10	10	11	11
	CPU	0.3603	0.3931	0.2884	0.2013	0.2238	0.2560	0.2180	0.2555	0.2328
	RES	$8.1e-07$	$7.1e-07$	$5.3e-07$	$7.5e-07$	$2.7e-07$	$5.3e-07$	$9.9e-07$	$7.2e-07$	$7.4e-07$
IB $(\alpha=0.1,\beta=0.01)$	Iter	9	7	7	6	7	8	9	10	10
	CPU	0.1708	0.1852	0.1633	0.1596	0.1569	0.1918	0.1752	0.1951	0.3255
	RES	$4.2e-07$	$6.2e-07$	$3.2e-07$	$1.0e-06$	$4.4e-07$	$3.5e-07$	$2.5e-07$	$4.3e-07$	$5.8e-07$

Table 4 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.2 with $\alpha = 0.1$, $\beta = 0.001$

$\mathbf{M} \rightarrow$		$5I_{m^2}$	$10I_{m^2}$	$20I_{m^2}$	$30I_{m^2}$	$40I_{m^2}$	$50I_{m^2}$	$70I_{m^2}$	$90I_{m^2}$	$100I_{m^2}$
HSS $(\alpha=0.1)$	Iter	17	12	8	6	8	10	13	17	19
	CPU	0.3658	0.3041	0.5328	0.1751	0.1857	0.2526	0.2921	0.3727	0.4012
	RES	$4.8e-07$	$9.2e-08$	$2.7e-07$	$8.2e-07$	$6.1e-07$	$7.3e-07$	$8.0e-07$	$6.8e-07$	$6.5e-07$
DPSS $(\alpha=0.1)$	Iter	18	14	10	9	10	10	10	11	11
	CPU	0.4637	0.2736	0.2059	0.1900	0.2892	0.2307	0.2183	0.2324	0.2307
	RES	$8.1e-07$	$7.1e-07$	$5.3e-07$	$7.5e-07$	$2.7e-07$	$5.3e-07$	$9.9e-07$	$7.2e-07$	$7.4e-07$
IB $(\alpha=0.1,\beta=0.001)$	Iter	13	11	9	8	8	8	9	10	10
	CPU	0.2833	0.2228	0.1868	0.1729	0.1835	0.2005	0.1899	0.1993	0.2133
	RES	$9.1e-07$	$7.3e-07$	$6.5e-07$	$4.0e-06$	$3.3e-07$	$7.9e-07$	$4.6e-07$	$3.6e-07$	$8.0e-07$

Table 5 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.2 with $\alpha = 0.1$, $\beta = 10^{-5}$

M $\rightarrow$		$5I_{m^2}$	$10I_{m^2}$	$20I_{m^2}$	$30I_{m^2}$	$40I_{m^2}$	$50I_{m^2}$	$70I_{m^2}$	$90I_{m^2}$	$100I_{m^2}$
HSS $(\alpha=0.1)$	Iter	17	12	8	6	8	10	13	17	19
	CPU	0.4525	0.3473	0.3187	0.1956	0.2027	0.2209	0.3096	0.3286	0.3972
	RES	$4.8e-07$	$9.2e-08$	$2.7e-07$	$8.2e-07$	$6.1e-07$	$7.3e-07$	$8.0e-07$	$6.8e-07$	$6.5e-07$
DPSS $(\alpha=0.1)$	Iter	18	14	10	9	10	10	10	11	11
	CPU	0.3719	0.3248	0.2071	0.2294	0.2096	0.3130	0.2181	0.3391	0.2330
	RES	$8.1e-07$	$7.1e-07$	$5.3e-07$	$7.5e-07$	$2.7e-07$	$5.3e-07$	$9.9e-07$	$7.2e-07$	$7.4e-07$
IB $(\alpha=0.1,\beta=10^{-5})$	Iter	14	11	9	8	8	8	9	10	10
	CPU	0.2711	0.2288	0.1837	0.1822	0.1774	0.2085	0.1989	0.2055	0.2722
	RES	$4.1e-07$	$7.9e-07$	$6.9e-07$	$4.2e-06$	$3.3e-07$	$8.0e-07$	$4.6e-07$	$3.7e-07$	$8.0e-07$

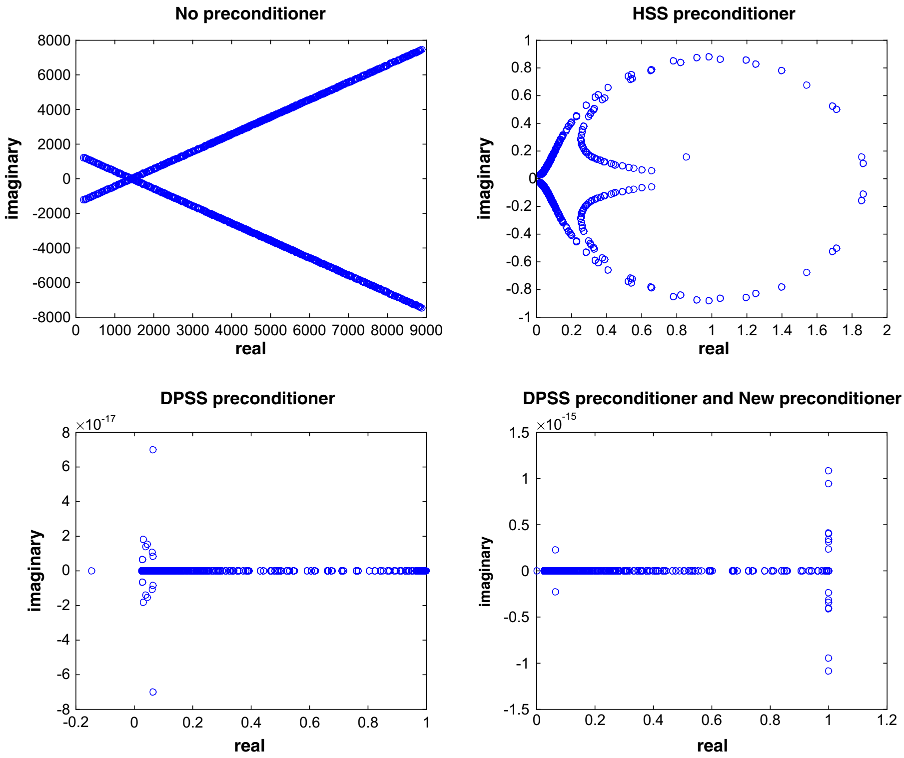

Fig. 5 The eigenvalue distributions of original matrix $\mathscr{A}$ and preconditioned matrices $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, $\mathscr{P}_1^{-1}\mathscr{A}$, and $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for the Example 5.3 with $m = 32$

the matrix T is symmetric positive definite. In this example, we set $m = 32$ and the right-hand side $b = \mathscr{A} * \text{ones}(2m^2, 1)$.

Table 6 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.3 with $\alpha = 1, \beta = 0.001$

Preconditioner	m	16	32	64	128
HSS (α=1)	Iter	490	408	–	–
	CPU	1.9356	61.1594	–	–
	RES	$1.0e-06$	$1.0e-06$	–	–
DPSS (α=1)	Iter	44	50	78	108
	CPU	0.1639	0.9870	8.1601	59.4973
	RES	$8.5e-08$	$8.8e-07$	$9.6e-07$	$9.9e-07$
IB (α=1, β=0.001)	Iter	43	49	75	100
	CPU	0.1537	0.9650	7.7582	54.9888
	RES	$9.6e-7$	$8.7e-07$	$9.5e-07$	$1.0e-06$

Table 7 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.3 with $\alpha = 10, \beta = 0.001$

Preconditioner	m	16	32	64	128
HSS (α=10)	Iter	276	595	–	–
	CPU	1.1183	11.7187	–	–
	RES	$9.7e-07$	$9.9e-07$	–	–
DPSS (α=10)	Iter	25	39	74	139
	CPU	0.0921	0.8323	8.0476	76.2899
	RES	$5.3e-08$	$9.7e-07$	$9.5e-07$	$9.6e-07$
IB (α=10, β=0.001)	Iter	22	38	70	130
	CPU	0.0835	0.8254	7.7649	73.0925
	RES	$5.5e-7$	$9.2e-07$	$9.9e-07$	$9.7e-07$

Eigenvalue distributions of the original matrix $\mathscr{A}$, HSS preconditioned matrix $\mathscr{P}_{HSS}^{-1}\mathscr{A}$, DPSS preconditioned matrix $\mathscr{P}_1^{-1}\mathscr{A}$, and new preconditioned matrix $\mathscr{P}_{IB}^{-1}\mathscr{A}$ for $\mathbf{M} = 5I_{m^2}$, $\mathbf{M} = 50I_{m^2}$, and $\mathbf{M} = 70I_{m^2}$ are displayed in Figs. 2, 3, and 4. In these figures, we only test $\alpha = 0.1$, $\beta = 0.001$. Numerical results with different α and β are listed in Tables 2, 3, 4, and 5. The conclusion obtained from these figures and tables is similar to that of Example 5.1. The performance of the IB preconditioned GMRES(30) is better than that of its two counterparts. Furthermore, we can also observe that the iteration steps of the GMRES(30) are stable with the inertia matrix $\mathbf{M}$ for the new preconditioner $\mathscr{P}_{IB}$.

Example 5.3 We consider the following complex symmetric linear system [6, 19]

$$[(\mathbf{K} - (3-\sqrt{3})\omega^2 I_{m^2}) + i(\mathbf{K} + (3+\sqrt{3})\tau^2 I_{m^2})]x = b,$$

Table 8 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.3 with $\alpha = 100, \beta = 0.001$

Preconditioner	m	16	32	64	128
HSS (α=100)	Iter	49	139	318	–
	CPU	0.2062	2.7704	32.9252	–
	RES	$1.0e-06$	$9.9e-07$	$9.9e-07$	–
DPSS (α=100)	Iter	13	26	43	71
	CPU	0.0530	0.5359	4.5402	39.6449
	RES	$9.9e-08$	$7.2e-07$	$9.7e-07$	$9.6e-07$
IB (α=100, β=0.001)	Iter	13	24	38	66
	CPU	0.0431	0.4687	3.1354	34.2463
	RES	$4.2e-7$	$6.7e-07$	$9.5e-07$	$9.4e-07$

Table 9 Iter, CPU, and RES for the preconditioned GMRES(30) method in Example 5.3 with $\alpha = 1000$, $\beta = 0.001$

Preconditioner	m	16	32	64	128
HSS (α=1000)	Iter	11	20	42	–
	CPU	0.0530	0.4319	4.4682	–
	RES	$4.4e-07$	$7.9e-07$	$8.9e-07$	–
DPSS (α=1000)	Iter	12	15	23	37
	CPU	0.0473	0.2995	2.4310	21.5431
	RES	$5.3e-08$	$4.4e-07$	$7.2e-07$	$7.8e-07$
IB (α=1000,β=0.001)	Iter	9	13	21	35
	CPU	0.0356	0.2691	1.2315	12.5431
	RES	$1.8e-7$	$6.7e-07$	$9.5e-07$	$9.1e-07$

where $\mathbf{K} = I_m \otimes V_m + V_m \otimes I_m$, $\omega = 10\pi$, $\tau = 2\pi$, $h = \frac{1}{m+1}$, $N = m^2$ and $V_m = h^{-2}\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m}$ is a tridiagonal matrix. We choose the symmetric indefinite matrix $W = \mathbf{K}-(3-\sqrt{3})\omega^2 I_{m^2}$ and the symmetric positive definite matrix $T = \mathbf{K} + (3+\sqrt{3})\omega^2 I_{m^2}$.

Figure 5 shows the eigenvalue distributions of the original matrix $\mathscr{A}$, the HSS preconditioned matrix $\mathscr{P}_{HSS}^{-1}\mathscr{A}$ and the preconditioned matrices $\mathscr{P}_1^{-1}\mathscr{A}$ and $\mathscr{P}_{IB}^{-1}\mathscr{A}$ with $m = 32$, $\alpha = 100$, and $\beta = 0.001$. Numerical results for different α, β, and m are listed in Tables 6, 7, 8, and 9.

From Tables 6, 7, 8, and 9, we can see that the HSS preconditioned GMRES(30) sometimes does not converge for $m \geq 64$, and the IB preconditioner returns better numerical results than the HSS and DPSS preconditioners in terms of the iteration steps, CPU time, and relative residual error. Moreover, we also find the numerical results for the IB preconditioner $\mathscr{P}_{IB}$ in Table 9 (when $\alpha = 1000, \beta = 0.001$) are much better than the other cases. This result echoes the conclusion obtained in Section 2 that when $\alpha \to \infty$ and $\beta \to 0$, the preconditioner $\mathscr{P}_{IB}$ is much closer to the coefficient matrix $\mathscr{A}$, then the rate of convergence will be rapid, i.e., the GMRES(30) will terminate within a small number of steps.

6 Conclusion

To solve a class of complex symmetric indefinite linear systems, an improved block splitting preconditioner is proposed in this paper. By adopting two iteration parameters and the relaxation technique, the new preconditioner not only remains the same computational cost with the block preconditioner but also is much closer to the original coefficient matrix. Theoretical analysis proves that the corresponding iteration method is convergent under suitable conditions and the preconditioned matrix has a well-clustered eigenvalue distribution with a reasonable choice of the relaxation parameters. Numerical experiments are presented to illustrate the presented preconditioner is competitive with other existing block preconditioners.

References

1. Axelsson, O., Kucherov, A.: Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra Appl. **7**, 197–218 (2000)
2. 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. **24**, 603–626 (2003)
3. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**, 1–32 (2004)
4. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. SIAM. J. Sci. Comput. **26**, 844–863 (2005)
5. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA. J. Numer. Anal. **27**, 1–23 (2007)
6. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
7. Bai, Z.-Z.: Several splittings for non-Hermitian linear systems. Sci. China Ser. A: Math. **51**, 1339–1348 (2008)
8. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algor. **56**, 297–317 (2011)
9. Bai, Z.-Z.: Block preconditioners for elliptic PDE-constrained optimization problems. Computing. **91**, 379–395 (2011)
10. 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 application to distributed control problems. IMA J. Numer. Anal. **33**, 343–369 (2013)
11. Bai, Z.-Z.: Rotated block triangular preconditioning based on PMHSS. Sci. China Math. **56**, 2523–2538 (2013)
12. Bai, Z.-Z.: On preconditioned iteration methods for complex linear systems. J. Eng. Math. **93**, 41–60 (2015)
13. Benzi, M., Bertaccini, D.: Block preconditioning of real-valued iterative algorithms for complex linear systems. IMA. J. Numer. Anal. **28**, 598–618 (2008)
14. Fan, H.-T., Zhu, X.-Y.: A generalized relaxed positive-definite and skew-Hermitian splitting preconditioner for non-Hermitian saddle point problems. Appl. Math. Comput. **58**, 36–48 (2015)
15. Fan, H.-T., Zhu, X.-Y.: A modified relaxed splitting preconditioner for generalized saddle point problems from the incompressible Navier-Stokes equations. Appl. Math. Lett. **55**, 18–26 (2016)
16. Golub, G.H., Van Loan, C.F. Matrix computations, 3rd edn. The Johns Hopkins University Press, (1996)
17. Golub, G.H., Greif, C.: On solving block-structured indefinite linear systems. SIAM J. Sci. Comput. **24**, 2076–2092 (2003)
18. Lang, C., Ren, Z.-R.: Inexact rotated block triangular preconditioners for a class of block two-by-two matrices. J. Eng. Math. **93**, 87–98 (2015)
19. Li, X., Yang, A.-L., Wu, Y.-J.: Lopsided PMHSS iteration method for a class of complex symmetric linear systems. Numer. Algor. **66**, 555–568 (2014)
20. Pan, J.-Y., Ng, M.K., Bai, Z.-Z.: New preconditioners for saddle point problems. Appl. Math. Comput. **72**, 762–771 (2006)
21. Pennacchio, M., Simoncini, V.: Fast structured AMG preconditioning for the bidomain model in electrocardiology. SIAM J. Sci. Comput. **33**, 721–745 (2011)
22. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
23. Saad, Y. Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
24. Wu, S.-L.: Several variants of the Hermitian and skew-Hermitian splittingmethod for a class of complex symmetric linear systems. Numer. Linear Algebra Appl. **22**, 338–356 (2015)
25. Zhang, J.-L., Gu, C.-Q., Zhang, K.: A relaxed positive-definite and skew-Hermitian splitting preconditioner for saddle point problems. Appl. Math. Comput. **249**, 468–479 (2014)

26. Zhang, J.-L., Gu, C.-Q.: A variant of the deteriorated PSS preconditioner for nonsymmetric saddle point problems. BIT Numer. Math. **56**, 587–604 (2016)
27. Zhang, J.-H., Dai, H.: A new block preconditioner for complex symmetric indefinite linear systems. Numer. Algor. doi:10.1007/s11075-016-0175-y
28. Zhang, J.-H., Dai, H.: A new splitting preconditioner for the iterative solution of complex symmetric indefinite linear systems. Appl. Math. Lett. **49**, 100–106 (2015)
29. Zheng, Q.-Q., Ma, C.-F.: Accelerated PMHSS iteration methods for complex symmetric linear systems. Numer. Algor. doi:10.1007/s11075-016-0105-z