



Preconditioned triangular splitting iteration method for a class of complex symmetric linear systems

Zheng-Ge Huang¹ · Zhong Xu² · Jing-Jing Cui²

Received: 13 February 2019 / Accepted: 15 May 2019 / Published online: 18 May 2019
© Istituto di Informatica e Telematica (IIT) 2019

Abstract

For the large sparse complex symmetric linear systems, we construct a preconditioned triangular splitting (PTS) iteration method based on utilizing the preconditioned technique and the triangular splitting of a matrix. Compared with the two-parameter two-step scale-splitting one established by Salkuyeh and Siahkolaei (Calcolo 55:8, 2018), PTS iteration method does not involve the complex arithmetic. The convergence theory of the PTS iteration method is established and the spectral properties of the PTS-preconditioned matrix are analyzed. In addition, by applying the minimum residual technique to the PTS iteration method, we develop the minimum residual PTS (MRPTS) iteration method to further improve the efficiency of the PTS one, then establish the corresponding convergence theory. Also, inexact version of the MRPTS iteration method and its convergence properties are presented. Numerical experiments are reported to verify the effectiveness of the proposed methods.

Keywords Complex symmetric linear systems · Preconditioned triangular splitting iteration method · Minimum residual technique · Convergence properties · Inexact variant

Mathematics Subject Classification 65F10 · 65F50

This research was supported by the National Natural Science Foundation of China (No. 10802068).

✉ Zheng-Ge Huang
ZhenggeHuang@mail.nwpu.edu.cn

Zhong Xu
zhongxu@nwpu.edu.cn

Jing-Jing Cui
JingjingCui@mail.nwpu.edu.cn

¹ Faculty of Science, Guangxi University for Nationalities, Nanning 530006, People's Republic of China

² Department of Applied Mathematics, School of Science, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, People's Republic of China

1 Introduction

In this paper we consider iterative methods for solving the complex symmetric linear systems of the form:

$$Ax \equiv (W + iT)x = b, \quad (1)$$

where $W, T \in \mathbb{R}^{n \times n}$ are symmetric positive semi-definite matrices with at least one of them, for example, W , being positive definite, $b \in \mathbb{R}^n$ and $i = \sqrt{-1}$.

Linear systems of the form (1) arise frequently from many practical problems in scientific computing and engineering applications, such as FFT-based solution of certain time-dependent PDEs, diffuse optical tomography, molecular scattering, Helmholtz equations [9,10,22,23,35], and many more, see e.g. [18].

Due to the universal existence and significance of the complex symmetric linear systems, there has been a surge of interest in (1), and numerous solution techniques have been proposed for this type of system in recent years. Based on the Hermitian and skew-Hermitian splitting (HSS) of the matrix A in (1): $A = H + S$ with $H = \frac{1}{2}(A + A^*) = W$ and $S = \frac{1}{2}(A - A^*) = iT$, Bai et al. [14] initially established the HSS iteration method. Here, A^* denotes the conjugate transpose of the matrix A . However, a shifted skew-Hermitian linear system needs to be solved at each iteration step of the HSS iteration method. To overcome this difficult, Bai et al. [9] skillfully designed a modified HSS (MHSS) method. Then, Bai et al. [10] proposed a preconditioned version of the MHSS iteration method called PMHSS for solving (1), and proved it is convergent unconditionally. Due to the excellent properties of the PMHSS iteration method, it is a matter of great concern and its some modified and generalized versions are derived recently. Dehghan et al. [21] presented the generalized PMHSS (GPMHSS) iteration method by introducing an additional parameter. After that, the lopsided PMHSS (LPMHSS) iteration method was proposed by Li et al. [26], which outperforms the PMHSS one when the real part of the coefficient matrix is dominant. In addition to the aforementioned methods, there exist other meaningful variants of the MHSS method, see [37,39] for more details.

Moreover, by multiplying a complex number through both sides of the complex system (1), Hezari et al. [23] designed a scale-splitting (SCSP) iteration method. Using the idea of [23], Zheng et al. [42] proposed a double-step scale splitting (DSS) iteration method. Subsequently, Salkuyeh and Siahkolaei [32] constructed a two-parameter version of the DSS iteration method called the two-parameter two-step SCSP (TTSCSP) method, which is more efficient than the DSS one and some known ones. The TTSCSP iteration method can be described as follows:

The two-parameter two-step SCSP (TTSCSP) iteration method (see [32]) Let α and β be two positive constants. Given an initial guess $x^{(0)}$. For $k = 0, 1, 2, \dots$, until $x^{(k)}$ converges, compute

$$\begin{cases} (\alpha W + T)x^{(k+\frac{1}{2})} = i(W - \alpha T)x^{(k)} + (\alpha - i)b, \\ (\beta T + W)x^{(k+1)} = i(\beta W - T)x^{(k+\frac{1}{2})} + (1 - \beta i)b. \end{cases} \quad (2)$$

The iteration matrix of the TTSCSP iteration method is

$$T(\alpha, \beta) = (\beta T + W)^{-1}(\beta W - T)(\alpha W + T)^{-1}(\alpha T - W). \quad (3)$$

Besides, the CRI (the combination method of real part and imaginary part) iteration method was derived by Wang et al. [34], where coefficient matrices of the two sub-linear systems are the same as those in the DSS one. Recently, Xiao and Wang [36] constructed the parameterized variant of the fixed-point iteration adding the asymmetric error (PFPAE) iteration method for solving (1). To further accelerate the convergence rates of the PFPAE and the DSS iteration methods, Huang et al. [24] used the idea of PFPAE method and the scaling technique to design the two-step parameterized (TSP) iteration method. Some numerical experiments showed that the performance of the TSP method is superior to some existing ones.

As stated in [8], in order to simplify data handling and construct efficient preconditioning matrix, it is beneficial to transform (1) with $x = u + iv$, $b = p + iq$ and $u, v, p, q \in \mathbb{R}^n$ into the two-by-two block real equivalent formulation

$$\mathcal{A} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} W & -T \\ T & W \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} p \\ q \end{pmatrix} \quad (4)$$

with u and v being unknown vectors. This can avoid complex arithmetic when solving it. And it may be efficiently solved in real arithmetics by many iteration methods. The linear system (4) can be formally regarded as a special case of the generalized saddle point problem [17] and frequently arises from many important practical problems. As an illustration, the distributed control problem can be transformed into the linear system:

$$\begin{pmatrix} M & 0 & K \\ 0 & \beta M & -M \\ K & -M & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ 0 \\ \mathbf{d} \end{pmatrix},$$

by using the discretize-then optimize method [5], where M and K are the mass matrix and the stiffness matrix, respectively.

For saddle point problems, Bai et al. [15] established the generalized successive overrelaxation (GSOR) iteration method. After that, Bai and Wang [16] proposed the parameterized inexact Uzawa (PIU) iteration method, which is a generalized version of the GSOR one. Based on [15], Salkuyeh et al. [31] defined the GSOR iteration method for solving the linear system (4) recently. To further improve the efficiency of the GSOR iteration method, Hezari et al. [22] designed the preconditioned GSOR (PGSOR) iteration method for (4), whose optimal convergence factor is smaller than that of the GSOR one. Alternatively in [27], Li et al. newly derived a symmetric block triangular splitting (SBTS) iteration method. Then its preconditioned variant was proposed by Zhang et al. [41], which performs better than the SBTS one under some restrictions. Recently, Axelsson and Salkuyeh [3] designed a novel method called the transformed matrix iteration (TMIT) method by transforming (4) into

$$\begin{pmatrix} W + \alpha T & 2\alpha W + (\alpha^2 - 1)T \\ T & W + \alpha T \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{v} \end{pmatrix} = \begin{pmatrix} p + \alpha q \\ q \end{pmatrix},$$

and using the triangular splitting of the corresponding coefficient matrix. And the corresponding transformed matrix preconditioner (TMP) was also given. The authors proved that the proposed method and the preconditioned matrix have tight eigenvalue bounds. Numerical results included in [3] showed that the proposed method and preconditioner outperform some existing ones. For more iteration methods for linear system (4), we refer to [2,8,11,25,28,33,40] and the references therein.

It can be seen from (2) that a potential difficulty with the TTSCSP iteration method is the need to use complex arithmetic, which may cost some computing times. To overcome this disadvantage, in this current work, we are going to develop a new approach called the preconditioned triangular splitting (PTS) method for solving (4). Although its convergence factor is the same as that of the TTSCSP one, in the PTS iteration method we deal only with real arithmetic. So it is expected that the PTS iteration method exhibits better numerical results than the TTSCSP one. Besides as a solver, the PTS iteration is also used as a preconditioner to accelerate Krylov subspace methods such as the generalized minimal residual (GMRES) method. In order to further improve the convergence behavior of the PTS iteration method, we establish the minimum residual PTS (MRPTS) iteration method by introducing a variable for the PTS iteration method. It is noteworthy that this idea comes essentially from [38], which is firstly applied to develop the MRHSS iteration method in [38]. Besides, by adopting the approximate solution strategy, we correspondingly design an inexact MRPTS (IMRPTS) iteration method. Theoretically, we discuss the convergence of the proposed methods and the spectral properties of the PTS-preconditioned matrix.

The organization of this paper is as follows. The next section introduces the PTS iteration method and analyzes its convergence. In Sect. 3, we propose the PTS preconditioner and study the spectral properties of the PTS-preconditioned matrix. We derive the MRPTS iteration method and investigate its convergence properties in Sect. 4. The inexact MRPTS iteration method and its convergence analysis are given in Sect. 5. Section 6 is devoted to some numerical results to demonstrate that the performances of the proposed methods are better than the aforementioned ones. Finally, in Sect. 7 we put forth some concluding remarks to end the paper.

We end this section with an introduction of some notations that will be used in the subsequent analysis. For a square matrix H , $\rho(H)$, $\kappa(H)$ and $\det(H)$ denote the spectral radius, the condition number and the determinant of H , respectively. Moreover, the Euclidean norm of a vector or a matrix is represented by $\|\cdot\|_2$.

2 The preconditioned triangular splitting (PTS) iteration method

In this section, we establish the preconditioned triangular splitting (PTS) iteration method and its convergence properties.

Firstly, we premultiply (4) with the block matrix

$$P = \begin{pmatrix} \alpha I & I \\ -\beta I & I \end{pmatrix}$$

with $\alpha, \beta > 0$ and obtain

$$P\mathcal{A} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha W + T & W - \alpha T \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \alpha p + q \\ q - \beta p \end{pmatrix} := \bar{b}. \tag{5}$$

Here, the matrix P is nonsingular owing to the fact that $\det(P) = \alpha + \beta > 0$.

Then we split the coefficient matrix of (5) into

$$\begin{aligned} \bar{\mathcal{A}} := P\mathcal{A} &= \begin{pmatrix} \alpha W + T & W - \alpha T \\ T - \beta W & \beta T + W \end{pmatrix} = \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix} \\ &\quad - \begin{pmatrix} 0 & \alpha T - W \\ 0 & 0 \end{pmatrix} := \mathcal{M}(\alpha, \beta) - \mathcal{N}(\alpha, \beta), \end{aligned} \tag{6}$$

which is the block lower and upper triangular splitting of the matrix $\bar{\mathcal{A}}$.

Based on the splitting (6), we construct the preconditioned triangular spitting (PTS) iterative scheme:

$$\begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} u^{(k+1)} \\ v^{(k+1)} \end{pmatrix} = \begin{pmatrix} 0 & \alpha T - W \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix} + \begin{pmatrix} \alpha p + q \\ q - \beta p \end{pmatrix}, \tag{7}$$

which results in the following PTS iteration method.

The preconditioned triangular spitting (PTS) iteration method Let $\alpha, \beta > 0$. Given initial vectors $u^{(0)}$ and $v^{(0)}$. For $k = 0, 1, 2, \dots$, until $(u^{(k)}; v^{(k)})$ converges, compute

$$\begin{cases} (\alpha W + T)u^{(k+1)} = (\alpha T - W)v^{(k)} + \alpha p + q, \\ (\beta T + W)v^{(k+1)} = (\beta W - T)u^{(k+1)} + q - \beta p. \end{cases} \tag{8}$$

It is noteworthy that the motivation of constructing the PTS iteration method stems from [3] where Axelsson and Salkuyeh transformed the given matrix to a proper form and designed the efficient method. The idea of using such a simple block two-by-two matrix P in (5) to transform the block two-by-two matrix \mathcal{A} was first given by Axelsson and Kucherov [1]. This idea was further generalized by Bai [6,8], in which the author also first technically employed block Gauss–Seidel and block symmetric Gauss–Seidel splitting techniques to the corresponding transformed matrix and obtained the rotated block triangular preconditionings based on PMHSS. Moreover, the idea of introducing different parameters into the two block equations separately is dated back to [13,15,16].

The iteration scheme (8) indicates that the main costs at each step of the PTS iteration method are solving two linear subsystems with respect to the symmetric positive definite matrices $\alpha W + T$ and $\beta T + W$. They can be solved exactly by the Cholesky

factorization or inexactly by the conjugate gradient (CG) or the preconditioned CG (PCG) methods.

It follows from (7) that the iteration matrix of the PTS iteration method is

$$\begin{aligned}
 \mathcal{G}(\alpha, \beta) &= \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix}^{-1} \begin{pmatrix} 0 & \alpha T - W \\ 0 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} (\alpha W + T)^{-1} & 0 \\ -(\beta T + W)^{-1}(T - \beta W)(\alpha W + T)^{-1} & (\beta T + W)^{-1} \end{pmatrix} \begin{pmatrix} 0 & \alpha T - W \\ 0 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 0 & (\alpha W + T)^{-1}(\alpha T - W) \\ 0 & (\beta T + W)^{-1}(T - \beta W)(\alpha W + T)^{-1}(W - \alpha T) \end{pmatrix} \\
 &= \begin{pmatrix} 0 & (\alpha W + T)^{-1}(\alpha T - W) \\ 0 & T(\alpha, \beta) \end{pmatrix}. \tag{9}
 \end{aligned}$$

Remark 2.1 Equation (9) implies that $\rho(\mathcal{G}(\alpha, \beta)) = \rho(T(\alpha, \beta))$, and the PTS and the TTSCSP iteration methods have the same optimal convergence factors for the same linear systems. However, the PTS iteration method in (8) does not involve complex arithmetic compared with the TTSCSP one in (2). So it may cost less times than the TTSCSP one, which is illustrated by the numerical experiments.

In what follows, according to Theorem 3 in [32], we obtain the convergence conditions of the PTS iteration method.

Theorem 2.1 [32] *Let the matrices W and T be symmetric positive definite and symmetric positive semidefinite, respectively. Then, the PTS iteration method is convergent if α and β satisfy*

$$0 < \beta < \alpha \text{ and } \alpha > \frac{1}{2} \left(\frac{1}{\mu_s} - \mu_s \right) \text{ and } \beta > \frac{1}{2} \left(\frac{1}{\mu_n} - \mu_n \right),$$

where μ_s and μ_n are the minimum and maximum nonzero eigenvalues of the matrix $W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$, respectively.

3 The PTS preconditioner

The present section consists of two main parts. In the first part, the PTS preconditioner and its implementing procedure are presented. The second one is concerned with the spectral properties of the PTS-preconditioned matrix.

It is known that the Krylov subspace methods, such as the GMRES method, are a class of efficient iteration methods for solving large sparse linear system (4) [30]. However, in many practical problems, the coefficient matrices of linear system (4) are ill-conditioned. This makes the Krylov subspace methods often suffer from slow convergence or even stagnation, so preconditioning is in most cases indispensable for the iterative solution of (4).

3.1 The PTS preconditioner and its implementing procedure

It can be observed in (6) that the PTS iteration method is induced by the matrix splitting $\bar{A} = \mathcal{M}(\alpha, \beta) - \mathcal{N}(\alpha, \beta)$. The splitting matrix $\mathcal{M}(\alpha, \beta)$ can be employed to precondition the coefficient matrix of (5) and will be referred to as the PTS preconditioner. The form of the PTS preconditioner is

$$\mathcal{P}_{PTS} = \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix}. \quad (10)$$

Note that the PTS preconditioner is used to precondition the linear system (5), not (4).

In order to apply the PTS preconditioner of the form (10) within a Krylov-type method, it is necessary to solve the following linear system at each step:

$$\begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \quad (11)$$

with $z_1, z_2, r_1, r_2 \in \mathbb{R}^n$. In view of (11), we have the following algorithmic implementation of the PTS iterations:

Algorithm 3.1 For a given vector $r = (r_1; r_2)$, the vector $z = (z_1; z_2)$ of the linear system (11) can be solved according to the following procedure:

- (1) solve z_1 from the linear subsystem $(\alpha W + T)z_1 = r_1$;
- (2) compute $t_1 = r_2 - (T - \beta W)z_1$;
- (3) solve z_2 from the linear subsystem $(\beta T + W)z_2 = t_1$.

Remark 3.1 The form of PTS preconditioner is similar to that of the rotated block lower triangular (RBLT) preconditioner [6], which is structured as

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

Note that two linear subsystems with different coefficient matrices need to be solved at each step of the PTS iteration, whereas in the RBLT iterations, the coefficient matrices of the two linear subsystems are the same. Nevertheless, the PTS preconditioner contains two parameters, and we expect that choosing proper parameters can lead to less iteration steps than the RBLT one.

3.2 Spectral analysis of the PTS-preconditioned matrix

The convergence behaviors of the Krylov subspaces methods are closely related to the eigenvalues and eigenvectors of the coefficient matrices [7] as well as the sizes of the Jordan blocks [4]. In view of this fact, next we study the spectral distribution and independency of the eigenvectors of the PTS-preconditioned matrix $\mathcal{P}_{PTS}^{-1}\bar{A}$.

Theorem 3.1 *Let W be symmetric positive definite and T be symmetric positive semi-definite, respectively. Denote by μ_i ($1 \leq i \leq n$) the eigenvalues of the matrix $S = W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$, and by μ_{\max} and μ_{\min} the maximum and minimum values among them, respectively. Then the eigenvalues λ of the PTS-preconditioned matrix are all real numbers, which contain an eigenvalue 1 with multiplicity at least n , and the remaining ones are of the form:*

$$\frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha}. \tag{12}$$

Their upper and the lower bounds are

(a) *If $\mu_{\max} \leq 1$, then*

$$\begin{aligned} & \min \left\{ \frac{(\alpha + \beta)(\mu_{\max}^2 + 1)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}, \frac{2(\alpha + \beta)\mu_{\min}^2}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \right\} \\ & \leq \lambda \leq \max \left\{ \frac{(\alpha + \beta)(\mu_{\min}^2 + 1)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha}, \frac{2(\alpha + \beta)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \right\}; \end{aligned} \tag{13}$$

(b) *If $\mu_{\min} \geq 1$, then*

$$\begin{aligned} & \min \left\{ \frac{2(\alpha + \beta)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}, \frac{(\alpha + \beta)(\mu_{\min}^2 + 1)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \right\} \\ & \leq \lambda \leq \max \left\{ \frac{(\alpha + \beta)(\mu_{\max}^2 + 1)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}, \frac{2(\alpha + \beta)\mu_{\max}^2}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha} \right\}; \end{aligned} \tag{14}$$

(c) *If $\mu_{\min} = \mu_1 \leq \mu_2 \leq \dots \leq \mu_k \leq 1 \leq \mu_{k+1} \leq \dots \leq \mu_n = \mu_{\max}$, then*

$$\begin{aligned} & (\alpha + \beta) \min \left\{ \frac{\mu_k^2 + 1}{(\beta\mu_k + 1)(\alpha + \mu_k)}, \frac{2\mu_{\min}^2}{(\beta\mu_{\min} + 1)(\alpha + \mu_{\min})}, \right. \\ & \quad \left. \frac{2}{(\beta\mu_{\max} + 1)(\alpha + \mu_{\max})}, \frac{\mu_{k+1}^2 + 1}{(\beta\mu_{k+1} + 1)(\alpha + \mu_{k+1})} \right\} \\ & \leq \lambda \leq (\alpha + \beta) \max \left\{ \frac{\mu_{\min}^2 + 1}{(\beta\mu_{\min} + 1)(\alpha + \mu_{\min})}, \frac{2}{(\beta\mu_{\min} + 1)(\alpha + \mu_{\min})}, \right. \\ & \quad \left. \frac{\mu_{\max}^2 + 1}{(\beta\mu_{\max} + 1)(\alpha + \mu_{\max})}, \frac{2\mu_{\max}^2}{(\beta\mu_{\max} + 1)(\alpha + \mu_{\max})} \right\}. \end{aligned}$$

Furthermore, the quasi-optimal parameters of the PTS-preconditioned matrix are

$$\begin{aligned} \alpha^* &= \frac{1 - \mu_{\max}\mu_{\min} + \sqrt{(1 + \mu_{\min}^2)(1 + \mu_{\max}^2)}}{\mu_{\max} + \mu_{\min}}, \\ \beta^* &= \frac{\mu_{\max}\mu_{\min} - 1 + \sqrt{(1 + \mu_{\min}^2)(1 + \mu_{\max}^2)}}{\mu_{\max} + \mu_{\min}}, \end{aligned} \tag{15}$$

which minimize the upper bound of $|\lambda - 1|$. Then

$$|\lambda - 1| \leq \left(\frac{\mu_{\max}\sqrt{1 + \mu_{\min}^2} - \mu_{\min}\sqrt{1 + \mu_{\max}^2}}{\mu_{\max}\sqrt{1 + \mu_{\min}^2} + \mu_{\max}\sqrt{1 + \mu_{\max}^2}} \right)^2, \tag{16}$$

if the quasi-optimal parameters are adopted.

Proof It follows from (9) that

$$\begin{aligned} \mathcal{P}_{PTS}^{-1}\bar{A} &= I - \mathcal{G}(\alpha, \beta) \\ &= \begin{pmatrix} I & -(\alpha W + T)^{-1}(\alpha T - W) \\ 0 & I - \mathcal{T}(\alpha, \beta) \end{pmatrix} \\ &= \begin{pmatrix} I & -(\alpha W + T)^{-1}(\alpha T - W) \\ 0 & I - (\beta T + W)^{-1}(T - \beta W)(\alpha W + T)^{-1}(W - \alpha T) \end{pmatrix} \\ &= \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} I & -(\alpha I + S)^{-1}(\alpha S - I) \\ 0 & I - (\beta S + I)^{-1}(S - \beta I)(\alpha I + S)^{-1}(I - \alpha S) \end{pmatrix} \begin{pmatrix} W^{\frac{1}{2}} & 0 \\ 0 & W^{\frac{1}{2}} \end{pmatrix} \end{aligned}$$

with $S = W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$, which is similar to the matrix

$$\begin{pmatrix} I & -(\alpha I + S)^{-1}(\alpha S - I) \\ 0 & (\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1} \end{pmatrix}. \tag{17}$$

The above relation confirms that the eigenvalues of the PTS-preconditioned matrix are given by 1 with algebraic multiplicity at least n , and the remaining eigenvalues are those of the matrix $(\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1}$. Let μ_i ($1 \leq i \leq n$) be the eigenvalues of the matrix S , then the remaining eigenvalues of $\mathcal{P}_{PTS}^{-1}\bar{A}$ are of the form (12). From Lemma 2 of [31], we see that $\mu_i \in \mathbb{R}$ and hence $\lambda \in \mathbb{R}$. Straightforward computations reveal that

$$\begin{aligned} g(\mu_i) &:= \frac{\partial}{\partial \mu_i} \left(\frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \right) \\ &= \frac{(\alpha + \beta)[(\alpha\beta + 1)(\mu_i^2 - 1) + 2\mu_i(\alpha - \beta)]}{(\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha)^2}. \end{aligned} \tag{18}$$

Now we discuss the eigenvalue distributions of $\mathcal{P}_{PTS}^{-1}\bar{A}$ in the following cases:

- If $\mu_{\max} \leq 1$ and $\alpha \leq \beta$, then $\mu_i^2 - 1 \leq 0$ for all $i = 1, 2, \dots, n$ and hence $g(\mu_i) \leq 0$, which leads to

$$\frac{(\alpha + \beta)(\mu_{\max}^2 + 1)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha} \leq \lambda \leq \frac{(\alpha + \beta)(\mu_{\min}^2 + 1)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha}; \tag{19}$$

if $\mu_{\max} \leq 1$ and $\alpha > \beta$, then

$$\begin{aligned} \frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} &\leq \frac{2(\alpha + \beta)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \\ &\leq \frac{2(\alpha + \beta)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha}, \frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \\ &\geq \frac{2\mu_i^2(\alpha + \beta)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha}. \end{aligned}$$

In addition, it is not difficult to verify that

$$h(\mu_i) := \frac{\partial}{\partial \mu_i} \left(\frac{2\mu_i^2(\alpha + \beta)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \right) = \frac{2(\alpha + \beta)[(\alpha\beta + 1)\mu_i^2 + 2\alpha\mu_i]}{(\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha)^2} > 0. \tag{20}$$

One may deduce the following result

$$\frac{2(\alpha + \beta)\mu_{\min}^2}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \leq \lambda \leq \frac{2(\alpha + \beta)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha},$$

which together with (19) gives (13).

- If $\mu_{\min} \geq 1$ and $\alpha \geq \beta$, then $\mu_i^2 - 1 \geq 0$ for all $i = 1, 2, \dots, n$ and hence $g(\mu_i) \geq 0$. As a consequence,

$$\frac{(\alpha + \beta)(\mu_{\min}^2 + 1)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \leq \lambda \leq \frac{(\alpha + \beta)(\mu_{\max}^2 + 1)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}$$

is valid; if $\mu_{\min} \geq 1$ and $\alpha < \beta$, then we derive

$$\begin{aligned} \frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} &\leq \frac{2(\alpha + \beta)\mu_i^2}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \\ &\leq \frac{2(\alpha + \beta)\mu_{\max}^2}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}, \frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \\ &\geq \frac{2(\alpha + \beta)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha} \end{aligned}$$

$$\geq \frac{2(\alpha + \beta)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}$$

in view of (20). This leads to (14) what we were proving.

- If $\mu_{\min} = \mu_1 \leq \mu_2 \leq \dots \leq \mu_k \leq 1 \leq \mu_{k+1} \leq \dots \leq \mu_n = \mu_{\max}$, then similar to the derivation of (13) and (14), the following inequalities can be obtained

$$\begin{aligned} & \min \left\{ \frac{(\alpha + \beta)(\mu_k^2 + 1)}{\beta\mu_k^2 + (\alpha\beta + 1)\mu_k + \alpha}, \frac{2(\alpha + \beta)\mu_{\min}^2}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \right\} \\ & \leq \{\lambda | \mu_1 \leq \mu_2 \leq \dots \leq \mu_k \leq 1\} \\ & \leq \max \left\{ \frac{(\alpha + \beta)(\mu_{\min}^2 + 1)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha}, \frac{2(\alpha + \beta)}{\beta\mu_{\min}^2 + (\alpha\beta + 1)\mu_{\min} + \alpha} \right\} \end{aligned} \tag{21}$$

and

$$\begin{aligned} & \min \left\{ \frac{2(\alpha + \beta)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}, \frac{(\alpha + \beta)(\mu_{k+1}^2 + 1)}{\beta\mu_{k+1}^2 + (\alpha\beta + 1)\mu_{k+1} + \alpha} \right\} \\ & \leq \{\lambda | 1 \leq \mu_{k+1} \leq \dots \leq \mu_n\} \\ & \leq \max \left\{ \frac{(\alpha + \beta)(\mu_{\max}^2 + 1)}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha}, \frac{2(\alpha + \beta)\mu_{\max}^2}{\beta\mu_{\max}^2 + (\alpha\beta + 1)\mu_{\max} + \alpha} \right\}. \end{aligned} \tag{22}$$

Combining (21) and (22) yields that

$$\begin{aligned} & (\alpha + \beta) \min \left\{ \frac{\mu_k^2 + 1}{(\beta\mu_k + 1)(\alpha + \mu_k)}, \frac{2\mu_{\min}^2}{(\beta\mu_{\min} + 1)(\alpha + \mu_{\min})}, \right. \\ & \left. \frac{2}{(\beta\mu_{\max} + 1)(\alpha + \mu_{\max})}, \frac{\mu_{k+1}^2 + 1}{(\beta\mu_{k+1} + 1)(\alpha + \mu_{k+1})} \right\} \\ & \leq \lambda \leq (\alpha + \beta) \max \left\{ \frac{\mu_{\min}^2 + 1}{(\beta\mu_{\min} + 1)(\alpha + \mu_{\min})}, \frac{2}{(\beta\mu_{\min} + 1)(\alpha + \mu_{\min})}, \right. \\ & \left. \frac{\mu_{\max}^2 + 1}{(\beta\mu_{\max} + 1)(\alpha + \mu_{\max})}, \frac{2\mu_{\max}^2}{(\beta\mu_{\max} + 1)(\alpha + \mu_{\max})} \right\}. \end{aligned}$$

Furthermore, it is known that $\bar{\lambda} = 1 - \lambda$ holds, where $\bar{\lambda}$ stands for the eigenvalue of the PTS iteration matrix. By Theorem 1 of [32], we have

$$|\lambda - 1| \leq \max_{1 \leq i \leq n} \left| \frac{1 - \alpha\mu_i}{\alpha + \mu_i} \right| \left| \frac{\beta - \mu_i}{1 + \beta\mu_i} \right| := \sigma(\alpha, \beta). \tag{23}$$

Minimizing the upper bound $\sigma(\alpha, \beta)$ of $|\lambda - 1|$ in (23) may lead to clustered spectrum of the PTS-preconditioned matrix. Hence, all we need is to determine the quasi-optimal parameters which minimize $\sigma(\alpha, \beta)$. According to Theorem 2 of [32], the quasi-optimal parameters in (15) are obtained. Substituting them into (23) directly derives (16). This completes our proof of this theorem. \square

Theorem 3.2 *Let W be symmetric positive definite and T be symmetric positive semi-definite, respectively. Then the eigenvalues λ of the PTS-preconditioned matrix are 1 or satisfy*

- When $\alpha \rightarrow 0_+$ and $\beta \rightarrow 0_+$, then $\lambda \rightarrow 0_+$;
- When $\alpha \rightarrow +\infty$ and $\beta \rightarrow +\infty$, then $\lambda \rightarrow 0_+$;
- When $\alpha \rightarrow +\infty$ and $\beta \rightarrow 0_+$, then $\mu_{\min}^2 + 1 \leq \lambda \leq \mu_{\max}^2 + 1$;
- When $\alpha \rightarrow 0_+$ and $\beta \rightarrow +\infty$, then $\frac{1}{\mu_{\max}^2} + 1 \leq \lambda \leq \frac{1}{\mu_{\min}^2} + 1$.

Proof From Theorem 3.2, it can be seen that $\lambda = 1$ or

$$\lambda = \frac{(\alpha + \beta)(\mu_i^2 + 1)}{\beta\mu_i^2 + (\alpha\beta + 1)\mu_i + \alpha}, \quad i = 1, 2, \dots, n. \tag{24}$$

When $\alpha \rightarrow 0_+$ and $\beta \rightarrow 0_+$, then $\lambda \rightarrow 0_+$ in view of (24); when $\alpha \rightarrow +\infty$ and $\beta \rightarrow +\infty$, we infer that

$$\lambda = \frac{\left(\frac{1}{\alpha} + \frac{1}{\beta}\right)(\mu_i^2 + 1)}{\frac{1}{\alpha}\mu_i^2 + \left(1 + \frac{1}{\alpha\beta}\right)\mu_i + \frac{1}{\beta}} \rightarrow 0_+;$$

when $\alpha \rightarrow +\infty$ and $\beta \rightarrow 0_+$, it follows from straightforward computations that

$$\lambda = \frac{\left(1 + \frac{\beta}{\alpha}\right)(\mu_i^2 + 1)}{\frac{\beta}{\alpha}\mu_i^2 + \left(\beta + \frac{1}{\alpha}\right)\mu_i + 1} \rightarrow \mu_i^2 + 1,$$

thus, $\mu_{\min}^2 + 1 \leq \lambda \leq \mu_{\max}^2 + 1$ holds. At last, when $\alpha \rightarrow 0_+$ and $\beta \rightarrow +\infty$, it holds that

$$\lambda = \frac{\left(1 + \frac{\alpha}{\beta}\right)(\mu_i^2 + 1)}{\mu_i^2 + \left(\alpha + \frac{1}{\beta}\right)\mu_i + \frac{\alpha}{\beta}} \rightarrow 1 + \frac{1}{\mu_i^2},$$

which yields that $\frac{1}{\mu_{\max}^2} + 1 \leq \lambda \leq \frac{1}{\mu_{\min}^2} + 1$. \square

Theorem 3.2 implies that the PTS-preconditioned matrix has clustered spectrum if proper parameters are chosen. Meanwhile, both α and β can not be adopted so small or large that the PTS-preconditioned matrix becomes too close to being singular. Moreover, when $\alpha \rightarrow +\infty$ and $\beta \rightarrow 0_+$ or $\alpha \rightarrow 0_+$ and $\beta \rightarrow +\infty$, we get $\mu_{\min}^2 + 1 \leq \lambda \leq \mu_{\max}^2 + 1$ or $\frac{1}{\mu_{\max}^2} + 1 \leq \lambda \leq \frac{1}{\mu_{\min}^2} + 1$, respectively in accordance to

Theorem 3.2. However, when $\mu_{\min} \ll \mu_{\max}$, the spectrum of the PTS-preconditioned may be very scattered. Thus more proper choice of the parameters contained in the PTS preconditioner needs to be investigated.

As mentioned in [7], when the coefficient matrix A is nonsymmetric but diagonalizable with the eigenvector matrix V , the k th residual of the Krylov subspace methods is bounded by

$$\|r^{(k)}\|_2 \leq \kappa(V) \min_{\psi_k(0)=1} \max_{1 \leq i \leq n} |\psi_k(\lambda_i)| \|b\|_2, \tag{25}$$

where $\psi_k(x)$ is the polynomial of degree not greater than k . This indicates that the Euclidean norm of the residual is determined not only by the eigenvalues, but also by the eigenvectors of A for this case. The following theorem gives some analyses about the eigenvector distributions of the PTS-preconditioned matrix.

Theorem 3.3 *Let W be symmetric positive definite and T be symmetric positive semi-definite, respectively, then the PTS-preconditioned matrix has $2n$ linearly independent eigenvectors. There are*

- (1) i linearly independent eigenvectors of the form $\begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{bmatrix} I_i \\ 0 \end{bmatrix}$ that correspond to the eigenvalue 1, where I_i denotes the $i \times i$ ($i \geq n$) unit matrix;
- (2) $2n - i$ linearly independent eigenvectors of the form $\begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{bmatrix} D\bar{V} \\ \bar{V} \end{bmatrix}$ with $D = \text{diag}(\frac{\beta\mu_1+1}{\beta-\mu_1}, \dots, \frac{\beta\mu_{2n-i}+1}{\beta-\mu_{2n-i}})$ that correspond to the eigenvalues $\frac{(\alpha+\beta)(\mu_k^2+1)}{\beta\mu_k^2+(\alpha\beta+1)\mu_k+\alpha}$ ($1 \leq k \leq 2n - i$), where \bar{V} is the eigenvectors corresponding to the nonunit eigenvalues of $\mathcal{P}_{PTS}^{-1}\bar{A}$.

Besides, let

$$H = \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} I_i & D\bar{V} \\ 0 & \bar{V} \end{pmatrix} \tag{26}$$

be the $2n$ linearly independent eigenvectors of $\mathcal{P}_{PTS}^{-1}\bar{A}$, then its condition number satisfies

$$\kappa_2(H) \leq \frac{\left[\bar{\mu}_{\max}^2 + 2 + \bar{\mu}_{\max} \sqrt{\bar{\mu}_{\max}^2 + 4} \right] \sqrt{\lambda_{\max}}}{2\sqrt{\lambda_{\min}}},$$

where λ_{\max} and λ_{\min} are the maximum and minimum eigenvalues of the matrix W , respectively, and

$$\bar{\mu}_{\max} = \max \left\{ \left| \frac{\beta\tilde{\mu}_{\min} + 1}{\tilde{\mu}_{\min} - \beta} \right|, \left| \frac{\beta\tilde{\mu}_{\max} + 1}{\beta - \tilde{\mu}_{\max}} \right| \right\} \tag{27}$$

is the maximum value of the diagonal elements of D in modulus, with $\tilde{\mu}_{\max}$ and $\tilde{\mu}_{\min}$ being the maximum and minimum nonunit eigenvalues of G defined in (28), respectively.

Proof According to Theorem 3.1, we see that $\mathcal{P}_{PTS}^{-1}\bar{A}$ has the structure of the following form:

$$\mathcal{P}_{PTS}^{-1}\bar{A} = \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} I & -(\alpha I + S)^{-1}(\alpha S - I) \\ 0 & (\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1} \end{pmatrix} \begin{pmatrix} W^{\frac{1}{2}} & 0 \\ 0 & W^{\frac{1}{2}} \end{pmatrix}.$$

Now we first consider the eigenvectors of the matrix

$$G = \begin{pmatrix} I & -(\alpha I + S)^{-1}(\alpha S - I) \\ 0 & (\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1} \end{pmatrix}. \tag{28}$$

Equation (28) implies that the matrix G has the eigenvalue 1 with multiplicity at least n , and the remaining ones are those of the matrix $(\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1}$. We can check easily

$$G \begin{bmatrix} I_n \\ 0 \end{bmatrix} = \begin{pmatrix} I & -(\alpha I + S)^{-1}(\alpha S - I) \\ 0 & (\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1} \end{pmatrix} \begin{bmatrix} I_n \\ 0 \end{bmatrix} = \begin{bmatrix} I_n \\ 0 \end{bmatrix}.$$

Then G has n linearly independent eigenvectors of the form $\begin{bmatrix} I_n \\ 0 \end{bmatrix}$ that correspond to the eigenvalue 1. In addition, taking into account that S is symmetric positive semi-definite matrix, there exists an orthogonal matrix V such that $G = V\Lambda V^T$, where Λ is the diagonal matrix with the eigenvalues of S as its diagonal elements. Let $(x; y)$ be the eigenvector of G that corresponds to the eigenvalue λ , then it has

$$G \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} x \\ y \end{bmatrix}.$$

We rewrite the above equation in the form

$$\begin{cases} x + (\alpha I + S)^{-1}(I - \alpha S)y = \lambda x, \\ (\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1}y = \lambda y. \end{cases} \tag{29}$$

If $y = 0$, then from first equation of (29) it can be deduced that $(\lambda - 1)x = 0$. This leads to $\lambda = 1$ due to $x \neq 0$ in this case. Hence $(x; y)$ belongs to the Case (1). Now we consider the case that $y \neq 0$. Notice that $\lambda \neq 1$, otherwise (29) implies that $y = (\beta S + I)^{-1}(S - \beta I)(\alpha I + S)^{-1}(I - \alpha S)y = 0$ in contradiction with the assumption that $y \neq 0$. The second equation of (29) reveals that λ is the eigenvalue of $(\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1}$, and therefore

$$\lambda_k = \frac{(\alpha + \beta)(\mu_k^2 + 1)}{\beta\mu_k^2 + (\alpha\beta + 1)\mu_k + \alpha} = 1 - \frac{(\mu_k - \beta)(1 - \alpha\mu_k)}{(\beta\mu_i + 1)(\alpha + \mu_k)}, \quad 1 \leq k \leq 2n - i$$

with μ_k being the eigenvalues of the matrix S . It follows from $\lambda_k \neq 1$ that $(\mu_k - \beta)(1 - \alpha\mu_k) \neq 0$. For this case, we assume that $y_k = v_k$. Then

$$x_k + \frac{1 - \alpha\mu_k}{\alpha + \mu_k} v_k = \left[1 - \frac{(\mu_k - \beta)(1 - \alpha\mu_k)}{(\beta\mu_i + 1)(\alpha + \mu_k)} \right] x_k, \quad 1 \leq k \leq 2n - i,$$

that is

$$\frac{1 - \alpha\mu_k}{\alpha + \mu_k} v_k = -\frac{(\mu_k - \beta)(1 - \alpha\mu_k)}{(\beta\mu_i + 1)(\alpha + \mu_k)} x_k, \quad 1 \leq k \leq 2n - i,$$

which can be simplified as

$$x_k = \frac{\beta\mu_k + 1}{\beta - \mu_k} v_k, \quad 1 \leq k \leq 2n - i.$$

As a result, it holds that $\left[\frac{\beta\mu_k + 1}{\beta - \mu_k} v_k; v_k \right]$ is the eigenvector of the matrix G that corresponds to the nonunit eigenvalue $\frac{(\alpha + \beta)(\mu_k^2 + 1)}{\beta\mu_k^2 + (\alpha\beta + 1)\mu_k + \alpha}$. Since v_k ($1 \leq k \leq 2n - i$) are linearly independent, there are $2n - i$ linearly independent eigenvectors corresponding to the nonunit eigenvalues of G . Then the conclusions shown in (1) and (2) follow by the above analysis and Lemma 4.1 of [19], and the independence of the $2n$ eigenvectors of G is given by the nonsingularity of the matrix H defined as in (26).

In the sequel, we turn to discuss the condition number of the matrix H in (26). It follows from straightforward computations that

$$\begin{aligned} H &= \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} I_i & D\bar{V} \\ 0 & \bar{V} \end{pmatrix} = \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} I_{i-n} & 0 & 0 & 0 \\ 0 & I_{2n-i} & 0 & D\bar{V} \\ 0 & 0 & I_{i-n} & 0 \\ 0 & 0 & 0 & \bar{V} \end{pmatrix} \\ &= \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \begin{pmatrix} I_{i-n} & 0 & 0 & 0 \\ 0 & I_{2n-i} & 0 & D \\ 0 & 0 & I_{i-n} & 0 \\ 0 & 0 & 0 & I_{2n-i} \end{pmatrix} \begin{pmatrix} I_{i-n} & 0 & 0 & 0 \\ 0 & I_{2n-i} & 0 & 0 \\ 0 & 0 & I_{i-n} & 0 \\ 0 & 0 & 0 & \bar{V} \end{pmatrix}. \end{aligned}$$

By making use of the expression of the matrix H in the above equation we obtain

$$\begin{aligned} \kappa_2(H) &\leq \kappa_2 \left(\begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix} \right) \kappa_2 \left(\begin{pmatrix} I_{i-n} & 0 & 0 & 0 \\ 0 & I_{2n-i} & 0 & D \\ 0 & 0 & I_{i-n} & 0 \\ 0 & 0 & 0 & I_{2n-i} \end{pmatrix} \right) \\ &\quad \kappa_2 \left(\begin{pmatrix} I_{i-n} & 0 & 0 & 0 \\ 0 & I_{2n-i} & 0 & 0 \\ 0 & 0 & I_{i-n} & 0 \\ 0 & 0 & 0 & \bar{V} \end{pmatrix} \right) \end{aligned}$$

$$\begin{aligned}
 &= \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \kappa_2 \left(\begin{pmatrix} I_{i-n} & 0 & 0 & 0 \\ 0 & I_{2n-i} & 0 & D \\ 0 & 0 & I_{i-n} & 0 \\ 0 & 0 & 0 & I_{2n-i} \end{pmatrix} \right) \\
 &= \frac{\left[\bar{\mu}_{\max}^2 + 2 + \bar{\mu}_{\max} \sqrt{\bar{\mu}_{\max}^2 + 4} \right] \sqrt{\lambda_{\max}}}{2\sqrt{\lambda_{\min}}}
 \end{aligned}$$

with $\bar{\mu}_{\max}$ being the maximum value of the diagonal entry of D in modulus and defined as in (27). □

Remark 3.2 Theorem 3.3 shows that the PTS-preconditioned matrix has $2n$ linearly independent eigenvectors. As mentioned in [7] and shown in (25), if the matrix formed by the eigenvectors is well-conditioned in this case, at each additional iteration the degree increases by one and the accuracy can be quickly achieved. Thus a proper parameter β should be taken to make the upper bound of $\kappa_2(H)$ in Theorem 3.3 as small as possible.

The GMRES method will terminate when the degree of the minimum polynomial is attained. In particular, the degree of the minimum polynomial is equal to the dimension of the corresponding Krylov subspace [30]. In the sequel, some upper bounds of the dimension of the Krylov subspace $K(\mathcal{P}_{PTS}^{-1}\bar{A}, \mathcal{P}_{PTS}^{-1}\bar{b})$ are given.

Theorem 3.4 *Let W be symmetric positive definite and T be symmetric positive semi-definite, respectively, then the degree of the minimum polynomial of the PTS-preconditioned matrix is less than $n+1$, i.e., $K(\mathcal{P}_{PTS}^{-1}\bar{A}, \mathcal{P}_{PTS}^{-1}\bar{b}) \leq n+1$; in particular, if the matrix $(\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1}$ has q distinct nonunit eigenvalues, then the degree of the minimum polynomial of the PTS-preconditioned matrix is less than $q+1$, i.e., $K(\mathcal{P}_{PTS}^{-1}\bar{A}, \mathcal{P}_{PTS}^{-1}\bar{b}) \leq q+1$.*

Proof It follows from Theorem 3.3 that $\mathcal{P}_{PTS}^{-1}\bar{A}$ has $2n$ linearly independent eigenvectors, which means that $\mathcal{P}_{PTS}^{-1}\bar{A}$ is diagonalizable, and hence

$$p_{\min}(\mathcal{P}_{PTS}^{-1}\bar{A}) \leq p \left((x-1) \prod_{k=1}^n \left[\frac{(\alpha + \beta)(\mu_k^2 + 1)}{\beta\mu_k^2 + (\alpha\beta + 1)\mu_k + \alpha} \right] \right) = n + 1,$$

where $p_{\min}(A)$ and $p(g(x))$ represent the degrees of the minimal polynomial of the matrix A and $g(x)$, respectively. In addition, if the matrix $(\alpha + \beta)(\beta S + I)^{-1}(S^2 + I)(\alpha I + S)^{-1}$ has q distinct nonunit eigenvalues, then it has

$$p_{\min}(\mathcal{P}_{PTS}^{-1}\bar{A}) \leq p \left((x-1) \prod_{k=1}^q \left[\frac{(\alpha + \beta)(\mu_k^2 + 1)}{\beta\mu_k^2 + (\alpha\beta + 1)\mu_k + \alpha} \right] \right) = q + 1,$$

which completes the proof of this theorem. □

4 The minimum residual PTS (MRPTS) iteration method and its convergence properties

In order to further improve the efficiency of the PTS iteration method, we generalize the PTS iteration method and derive a minimum residual PTS (MRPTS) iteration method in this section.

In [38], the authors defined the minimize residual HSS (MRHSS) iteration method by introducing two parameters into the HSS iteration method. These parameters are adopted by minimizing the residual norms at each step of the HSS iteration scheme. Numerical experiments in [38] showed that the MRHSS method has advantages over the HSS one.

The PTS iteration method in (7) can be rewritten as

$$\begin{pmatrix} u^{(k+1)} \\ v^{(k+1)} \end{pmatrix} = \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix} + \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix}^{-1} \left(\bar{b} - \bar{A} \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix} \right). \tag{30}$$

By applying the idea of [38], we introduce a variable β_k into (30), which leads to a new iteration scheme of the form

$$\begin{pmatrix} u^{(k+1)} \\ v^{(k+1)} \end{pmatrix} = \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix} + \beta_k \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix}^{-1} \left(\bar{b} - \bar{A} \begin{pmatrix} u^{(k)} \\ v^{(k)} \end{pmatrix} \right). \tag{31}$$

As mentioned in [38], if proper parameter β_k is chosen, the convergence rate of (31) may be faster than that of (7). Next we discuss the optimal value of β_k in the MRPTS method. Inasmuch as (5) is a real linear system, it should be proper to adopt β_k in the real field \mathbb{R} here.

The residual form of the iteration scheme (31) can be written as

$$r^{(k+1)} = r^{(k)} - \beta_k \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}, \tag{32}$$

where $r^{(k)} = \bar{b} - \bar{A}(u^{(k)}; v^{(k)}) \in \mathbb{R}^{2n}$.

In view of Equation (8) in [38] and (32), $\|r^{(k+1)}\|_2^2$ can be briefly expressed as

$$\begin{aligned} \|r^{(k+1)}\|_2^2 &= (r^{(k+1)}, r^{(k+1)}) = (r^{(k)} - \beta_k \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}, r^{(k)} \\ &\quad - \beta_k \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}) \\ &= (r^{(k)}, r^{(k)}) - (r^{(k)}, \beta_k \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}) \\ &\quad - (\beta_k \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}, r^{(k)}) + \beta_k^2 \|\bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}\|_2^2 \\ &= \|r^{(k)}\|_2^2 - 2\beta_k (r^{(k)}, \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}) + \beta_k^2 \|\bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}\|_2^2. \end{aligned}$$

As a matter of fact, $(r^{(k)}, \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}) = (\bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}, r^{(k)})$ due to the facts that both $\bar{A} \mathcal{M}(\alpha, \beta)^{-1}$ and $r^{(k)}$ are real matrices and $(r^{(k)})^T \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)} = (\bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)})^T r^{(k)}$. One then obtains

$$\beta_k = \frac{(r^{(k)}, \bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)})}{\|\bar{A} \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}\|_2^2} \tag{33}$$

by solving the equation $\frac{\partial \|r^{(k+1)}\|_2^2}{\partial \beta_k} = 0$. Furthermore, we can easily check that

$$\frac{\partial^2 \|r^{(k+1)}\|_2^2}{\partial \beta_k^2} = 2\|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}r^{(k)}\|_2^2 \geq 0.$$

If $\|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}r^{(k)}\|_2^2 = 0$, then $r^{(k)} = 0$ by the nonsingularity of $\bar{\mathcal{A}}$ and $\mathcal{M}(\alpha, \beta)$. Hence $(u^{(k)}; v^{(k)})$ is the exact solution of (4), and we need not construct β_k to obtain $(u^{(k+1)}; v^{(k+1)})$ in this case. Then we assume that $r^{(k)} \neq 0$ for $k = 0, 1, 2, \dots$ throughout this paper. Under this assumption, it has

$$\frac{\partial^2 \|r^{(k+1)}\|_2^2}{\partial \beta_k^2} = 2\|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}r^{(k)}\|_2^2 > 0,$$

which implies that β_k defined as in (33) is the minimum point of the function $\|r^{(k+1)}\|_2$.

Therefore, we can form the following detailed implementation of the MRPTS iteration method:

The minimum residual PTS (MRPTS) iteration method Let $\alpha, \beta > 0$. Given $\varepsilon > 0$, an initial guess $(u^{(0)}; v^{(0)})$ and right-hand side vector $b = (p; q)$ of (4). For $k = 0, 1, 2, \dots$, until $(u^{(k)}; v^{(k)})$ converges,

Step 1: compute $\bar{r}^{(0)} = b - \mathcal{A}(u^{(0)}; v^{(0)})$, and divide $\bar{r}^{(0)}$ into $(\bar{r}_1^{(0)}; \bar{r}_2^{(0)})$ with $\bar{r}_1^{(0)}, \bar{r}_2^{(0)} \in \mathbb{R}^n$;

Step 2: compute $r_1^{(0)} = \alpha\bar{r}_1^{(0)} + \bar{r}_2^{(0)}$ and $r_2^{(0)} = \bar{r}_2^{(0)} - \beta\bar{r}_1^{(0)}$, and set $r^{(0)} = (r_1^{(0)}; r_2^{(0)})$ and $k = 0$;

Step 3: if $\frac{\|b - \mathcal{A}(u^{(k)}; v^{(k)})\|_2}{\|b\|_2} < \varepsilon$, then stop; otherwise, continue;

Step 4: solve $(\alpha W + T)t_1^{(k)} = r_1^{(k)}$;

Step 5: compute $t_2^{(k)} = r_2^{(k)} - (T - \beta W)t_1^{(k)}$;

Step 5: solve $(\beta T + W)t_3^{(k)} = t_2^{(k)}$;

Step 6: compute $t_4^{(k)} = (\alpha W + T)t_1^{(k)} + (W - \alpha T)t_3^{(k)}$ and $t_5^{(k)} = (T - \beta W)t_1^{(k)} + (\beta T + W)t_3^{(k)}$ and set $t^{(k)} = (t_4^{(k)}; t_5^{(k)})$;

Step 7: compute the value of β_k :

$$\beta_k = \frac{(r^{(k)}, t^{(k)})}{\|t^{(k)}\|_2^2} = \frac{(r_1^{(k)})^T t_4^{(k)} + (r_2^{(k)})^T t_5^{(k)}}{\|t_4^{(k)}\|_2^2 + \|t_5^{(k)}\|_2^2};$$

Step 8: compute $u^{(k+1)} = u^{(k)} + \beta_k t_1^{(k)}$, $v^{(k+1)} = v^{(k)} + \beta_k t_3^{(k)}$, $r_1^{(k+1)} = r_1^{(k)} - \beta_k t_4^{(k)}$ and $r_2^{(k+1)} = r_2^{(k)} - \beta_k t_5^{(k)}$; set $k = k + 1$ and return to Step 3.

Remark 4.1 When $\beta_k = 1$, the MRPTS iteration method automatically reduces to the PTS one. With a suitable choice of the parameter β_k the convergence rate of the MRPTS iteration method may be accelerated so as to be faster than the PTS one.

In what follows, the convergence properties of the MRPTS iteration method for solving the complex symmetric linear systems are discussed. To this end, we first demonstrate the following lemma.

Lemma 4.1 *Let $A = \text{diag}(a_1, \dots, a_n)$ and $B = \text{diag}(b_1, \dots, b_n)$ be two $n \times n$ real diagonal matrices, then*

$$\left\| \begin{pmatrix} AB & A \\ 0 & 0 \end{pmatrix} \right\|_2 = \max_{1 \leq i \leq n} \left\{ a_i \sqrt{1 + b_i^2} \right\}.$$

Proof Direct computations give

$$\left\| \begin{pmatrix} AB & A \\ 0 & 0 \end{pmatrix} \right\|_2 = \sqrt{\rho \left(\begin{pmatrix} AB & 0 \\ A & 0 \end{pmatrix} \begin{pmatrix} AB & A \\ 0 & 0 \end{pmatrix} \right)} = \sqrt{\rho \left(\begin{pmatrix} A^2 B^2 & A^2 B \\ A^2 B & A^2 \end{pmatrix} \right)} := \sqrt{\rho(H)}.$$

Note that the nonzero blocks of H are all diagonal matrices, then its eigenvalues are those of the matrices

$$H_i = \begin{pmatrix} a_i^2 b_i^2 & a_i^2 b_i \\ a_i^2 b_i & a_i^2 \end{pmatrix}, \quad 1 \leq i \leq n.$$

By direct calculations, we then immediately achieve the conclusion that we were proving. □

Theorem 4.1 *Let W be symmetric positive definite and T be symmetric positive semi-definite, respectively. Then the MRPTS iteration method is convergent if for any $\theta \in [0, \frac{\pi}{2}]$, the parameters α and β satisfy*

$$\begin{aligned} \max \left\{ \frac{1 - (\beta \mu_{\min} + 1) \sqrt{\frac{\lambda_{\min}}{\lambda_{\max}}} \cos \theta}{\mu_{\min}}, 0 \right\} < \alpha < \frac{1 + (\beta \mu_{\min} + 1) \sqrt{\frac{\lambda_{\min}}{\lambda_{\max}}} \cos \theta}{\mu_{\max}}, \\ \max \{ \mu_{\min} - (\alpha + \mu_{\max}) \tan \theta, 0 \} \leq \beta \leq \mu_{\min} + (\alpha + \mu_{\min}) \tan \theta. \end{aligned} \tag{34}$$

Proof Since β_k defined as in (33) is the minimum point of the function $\|r^{(k+1)}\|_2$, it holds that

$$\begin{aligned} \|r^{(k+1)}\|_2 &= \|r^{(k)} - \beta_k \bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1} r^{(k)}\|_2 \\ &\leq \|r^{(k)} - \bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1} r^{(k)}\|_2 \\ &= \|(I - \bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}) r^{(k)}\|_2 \\ &= \|\mathcal{N}(\alpha, \beta) \mathcal{M}(\alpha, \beta)^{-1} r^{(k)}\|_2 \\ &\leq \|\mathcal{N}(\alpha, \beta) \mathcal{M}(\alpha, \beta)^{-1}\|_2 \|r^{(k)}\|_2. \end{aligned}$$

Straightforward computations reveal that

$$\begin{aligned}
 & \mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1} \\
 &= \begin{pmatrix} 0 & \alpha T - W \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix}^{-1} \\
 &= \begin{pmatrix} 0 & \alpha T - W \\ 0 & 0 \end{pmatrix} \begin{pmatrix} (\alpha W + T)^{-1} & 0 \\ -(\beta T + W)^{-1}(T - \beta W)(\alpha W + T)^{-1} & (\beta T + W)^{-1} \end{pmatrix} \\
 &= \begin{pmatrix} (\alpha T - W)(\beta T + W)^{-1}(\beta W - T)(\alpha W + T)^{-1} & (\alpha T - W)(\beta T + W)^{-1} \\ 0 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} W^{\frac{1}{2}} & 0 \\ 0 & W^{\frac{1}{2}} \end{pmatrix} \begin{pmatrix} (\alpha S - I)(\beta S + I)^{-1}(\beta I - S)(\alpha I + S)^{-1} & (\alpha S - I)(\beta S + I)^{-1} \\ 0 & 0 \end{pmatrix} \\
 & \quad \begin{pmatrix} W^{-\frac{1}{2}} & 0 \\ 0 & W^{-\frac{1}{2}} \end{pmatrix}.
 \end{aligned}$$

Therefore, by making use of Lemma 4.1, we can conduct the estimate

$$\begin{aligned}
 & \|\mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1}\|_2 \\
 & \leq \sqrt{\frac{\lambda_{\min}}{\lambda_{\max}}} \left\| \begin{pmatrix} (\alpha S - I)(\beta S + I)^{-1}(\beta I - S)(\alpha I + S)^{-1} & (\alpha S - I)(\beta S + I)^{-1} \\ 0 & 0 \end{pmatrix} \right\|_2 \\
 & \leq \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}} \max_{1 \leq i \leq n} \left| \frac{\alpha \mu_i - 1}{\beta \mu_i + 1} \right| \sqrt{1 + \left| \frac{\beta - \mu_i}{\alpha + \mu_i} \right|^2} := \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) \tag{35}
 \end{aligned}$$

with λ_{\max} and λ_{\min} being the maximum and minimum eigenvalues of W , respectively. To get $\|\mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1}\|_2 < 1$, it is enough to have

$$\left| \frac{\alpha \mu_i - 1}{\beta \mu_i + 1} \right|^2 \left(1 + \left| \frac{\beta - \mu_i}{\alpha + \mu_i} \right|^2 \right) < \frac{\lambda_{\min}}{\lambda_{\max}} \tag{36}$$

for all μ_i ($1 \leq i \leq n$). For notational simplicity we denote by

$$\bar{c} = \left| \frac{\alpha \mu_i - 1}{\beta \mu_i + 1} \right|, \quad \bar{d} = \left| \frac{\beta - \mu_i}{\alpha + \mu_i} \right|, \quad g = \sqrt{\frac{\lambda_{\min}}{\lambda_{\max}}}.$$

If $\bar{c} < g \cos \theta$ and $\bar{d} < \tan \theta$ hold for any $\theta \in [0, \frac{\pi}{2}]$, then $\bar{c}^2(1 + \bar{d}^2) < g^2$, which is exactly (36). It follows from straightforward computations that a sufficient condition for guaranteeing $\bar{c} < g \cos \theta$ and $\bar{d} < \tan \theta$ is (34). Thus if (34) holds, we deduce that

$$\|r^{(k+1)}\|_2 \leq \|\mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1}\|_2 \|r^{(k)}\|_2 \leq \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) \|r^{(k)}\|_2 < \|r^{(k)}\|_2,$$

which results in

$$\lim_{k \rightarrow +\infty} \|r^{(k+1)}\|_2 \leq \lim_{k \rightarrow +\infty} \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) \|r^{(k)}\|_2$$

$$= \lim_{k \rightarrow +\infty} \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min})^k \|r^{(0)}\|_2 = 0,$$

i.e., the MRPTS iteration method is convergent. □

In the following, we study the choice for the parameters of the MRPTS iteration method, which can be used in practical computation. At each step of the MRPTS iteration method, two linear subsystems with the coefficient matrices $\alpha W + T$ and $\beta T + W$ require to be solved. By applying the idea of [20], the authors in [24] gave a strategy for choosing the parameters of the TSP iteration method. According to [24], the parameters α and β should satisfy

$$\begin{aligned} & \|(\alpha I + W^{-\frac{1}{2}}TW^{-\frac{1}{2}})^{-1}\|_2 \cdot \|\alpha I + W^{-\frac{1}{2}}TW^{-\frac{1}{2}}\|_2 \\ &= \|(\beta W^{-\frac{1}{2}}TW^{-\frac{1}{2}} + I)^{-1}\|_2 \cdot \|\beta W^{-\frac{1}{2}}TW^{-\frac{1}{2}} + I\|_2, \end{aligned}$$

which is equivalent to $(\alpha\beta - 1)(\mu_{\max} - \mu_{\min}) = 0$. Owing to the fact that $\mu_{\max} \neq \mu_{\min}$ holds in many practical problems, we consider to adopt proper parameters α and β satisfying $\alpha\beta = 1$ and the conditions of Theorem 3.1 in the MRPTS iteration method. The numerical results in Sect. 6 will illustrate the effectiveness of the MRPTS iteration method by using this strategy. Here, μ_{\max} and μ_{\min} denote the maximum and minimum eigenvalues of $W^{-\frac{1}{2}}TW^{-\frac{1}{2}}$, respectively.

5 The inexact MRPTS iteration method

The two half-steps at each step of the MRPTS iteration require finding solutions with the coefficient matrices $\alpha W + T$ and $\beta T + W$. If the problem size is very large, then using the direct methods like the Cholesky factorization to solve these problems is time consuming. To overcome this disadvantage, we prefer to utilize the inexact methods to solve these two linear equations. More specifically, we may employ the CG or PCG methods to solve the linear systems with coefficient matrices $\alpha W + T$ and $\beta T + W$, because they are symmetric positive definite. This yields the inexact MRPTS (IMRPTS) iteration for solving the linear system (4) as follows.

The inexact MRPTS (IMRPTS) iteration method Given an initial guess $(\bar{u}^{(0)}; \bar{v}^{(0)})$, for $k = 0, 1, 2, \dots$, until $(\bar{u}^{(k)}; \bar{v}^{(k)})$ converges, solve $(\bar{u}^{(k+1)}; \bar{v}^{(k+1)})$ approximately from

$$\begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} \bar{z}_1^{(k)} \\ \bar{z}_2^{(k)} \end{pmatrix} \approx \begin{pmatrix} r_1^{(k)} \\ r_2^{(k)} \end{pmatrix} = \bar{b} - \bar{\mathcal{A}} \begin{pmatrix} \bar{u}^{(k)} \\ \bar{v}^{(k)} \end{pmatrix}$$

by employing the CG (or PCG) method to solve the linear subsystems with the coefficient matrices $\alpha W + T$ and $\beta T + W$; then compute $(\bar{w}_1^{(k)}; \bar{w}_2^{(k)})$ by

$$\begin{pmatrix} \bar{w}_1^{(k)} \\ \bar{w}_2^{(k)} \end{pmatrix} = \begin{pmatrix} \alpha W + T & W - \alpha T \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} \bar{z}_1^{(k)} \\ \bar{z}_2^{(k)} \end{pmatrix},$$

and compute $(\bar{u}^{(k+1)}; \bar{v}^{(k+1)})$ by

$$\begin{pmatrix} \bar{u}^{(k+1)} \\ \bar{v}^{(k+1)} \end{pmatrix} = \begin{pmatrix} \bar{u}^{(k)} \\ \bar{v}^{(k)} \end{pmatrix} + \frac{(\bar{r}_1^{(k)})^T \bar{w}_1^{(k)} + (\bar{r}_2^{(k)})^T \bar{w}_2^{(k)}}{\|\bar{w}_1^{(k)}\|_2^2 + \|\bar{w}_2^{(k)}\|_2^2} \begin{pmatrix} \bar{z}_1^{(k)} \\ \bar{z}_2^{(k)} \end{pmatrix}, \tag{37}$$

where α and β are given positive constants.

To simplify numerical implementation and convergence analysis, we may rewrite the above IMRPTS iteration method as the following equivalent scheme.

Given an initial guess $(\bar{u}^{(0)}; \bar{v}^{(0)})$ and right-hand side vector $b = (p; q)$ of (4), for $k = 0, 1, 2, \dots$, until $(\bar{u}^{(k)}; \bar{v}^{(k)})$ converges,

1. compute $\bar{r}^{(k)} = b - \mathcal{A}(\bar{u}^{(k)}; \bar{v}^{(k)})$, and divide $\bar{r}^{(k)}$ into $(\bar{r}_1^{(k)}; \bar{r}_2^{(k)})$ with $\bar{r}_1^{(k)}, \bar{r}_2^{(k)} \in \mathbb{R}^n$;
2. compute $r_1^{(k)} = \alpha \bar{r}_1^{(k)} + \bar{r}_2^{(k)}$ and $r_2^{(k)} = \bar{r}_2^{(k)} - \beta \bar{r}_1^{(k)}$;
3. approximate the solution of $(\alpha W + T)\bar{z}_1^{(k)} = r_1^{(k)}$ by the CG (or PCG) method until $\bar{z}_1^{(k)}$ is such that the residual $\bar{p}^{(k)} = r_1^{(k)} - (\alpha W + T)\bar{z}_1^{(k)}$ satisfies $\|\bar{p}^{(k)}\| \leq \varepsilon_k \|\bar{r}_1^{(k)}\|$;
4. compute $\bar{t}^{(k)} = r_2^{(k)} - (T - \beta W)\bar{z}_1^{(k)}$;
5. solve $(\beta T + W)\bar{z}_2^{(k)} = \bar{t}^{(k)}$ by the CG (or PCG) method to compute the approximate solution $\bar{z}_2^{(k)}$ until it is such that the residual $\bar{q}^{(k)} = \bar{t}^{(k)} - (\beta T + W)\bar{z}_2^{(k)}$ satisfies $\|\bar{q}^{(k)}\| \leq \eta_k \|\bar{t}^{(k)}\|$;
6. compute $\bar{w}_1^{(k)} = (\alpha W + T)\bar{z}_1^{(k)} + (W - \alpha T)\bar{z}_2^{(k)}$ and $\bar{w}_2^{(k)} = (T - \beta W)\bar{z}_1^{(k)} + (\beta T + W)\bar{z}_2^{(k)}$;
7. compute the value of $\bar{\beta}_k$:

$$\bar{\beta}_k = \frac{(\bar{r}_1^{(k)})^T \bar{w}_1^{(k)} + (\bar{r}_2^{(k)})^T \bar{w}_2^{(k)}}{\|\bar{w}_1^{(k)}\|_2^2 + \|\bar{w}_2^{(k)}\|_2^2};$$

8. compute $\bar{u}^{(k+1)} = \bar{u}^{(k)} + \beta_k \bar{z}_1^{(k)}$, $\bar{v}^{(k+1)} = \bar{v}^{(k)} + \beta_k \bar{z}_2^{(k)}$, and $(\bar{r}_1^{(k+1)}; \bar{r}_2^{(k+1)}) = b - \mathcal{A}(\bar{u}^{(k+1)}; \bar{v}^{(k+1)})$.

Here, $\|\cdot\|$ is a norm of a vector, and $\{\varepsilon_k\}$ and $\{\eta_k\}$ are two prescribed tolerances.

Next, we analyze the convergence of the IMRPTS method. To this end, we consider a vector norm $\|x\|_{M_2} = \|M_2 x\|_2$ ($\forall x \in \mathbb{C}^n$) defined as in [14] and let $\delta_k = \max\{\varepsilon_k, \eta_k\}$.

Motivated by Theorem 3.1 of [14], we establish the following result related to the convergence property of the MRPTS iteration method.

Theorem 5.1 *Let $W, T \in \mathbb{R}^{n \times n}$ be symmetric positive definite and symmetric positive semi-definite, respectively, and let α and β be positive constants satisfying the condition (34), and $\bar{x}^{(k)} = (\bar{u}^{(k)}; \bar{v}^{(k)})$. If $\{\bar{x}^{(k)}\}$ is an iterative sequence generated by the IMRPTS iteration method and if $x^* \in \mathbb{C}^n$ is the exact solution of the system (4), then*

$$\|\bar{x}^{(k+1)} - x^*\|_{\bar{\mathcal{A}}} \leq \psi(k) \|\bar{x}^{(k)} - x^*\|_{\bar{\mathcal{A}}},$$

where

$$\psi(k) = \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) + \delta_k \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2.$$

In particular, if

$$\Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min})(1 + \delta_k) + \delta_k < 1, \tag{38}$$

then the sequence $\{\bar{x}^{(k)}\}$ converges to x^* , where $\delta_k = \max\{\varepsilon_k, \eta_k\}$.

Proof It follows from Steps 3 and 5 of the IMRPTS iteration method in the above and the definition of δ_k that

$$\begin{aligned} \mathcal{M}(\alpha, \beta) \begin{pmatrix} \bar{z}_1^{(k)} \\ \bar{z}_2^{(k)} \end{pmatrix} &= \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} \bar{z}_1^{(k)} \\ \bar{z}_2^{(k)} \end{pmatrix} = \begin{pmatrix} r_1^{(k)} \\ r_2^{(k)} \end{pmatrix} + \begin{pmatrix} \bar{p}^{(k)} \\ \bar{q}^{(k)} \end{pmatrix}, \\ \left\| \begin{pmatrix} \bar{p}^{(k)} \\ \bar{q}^{(k)} \end{pmatrix} \right\|_2 &= \left\| \begin{pmatrix} r_1^{(k)} \\ r_2^{(k)} \end{pmatrix} - \begin{pmatrix} \alpha W + T & 0 \\ T - \beta W & \beta T + W \end{pmatrix} \begin{pmatrix} \bar{z}_1^{(k)} \\ \bar{z}_2^{(k)} \end{pmatrix} \right\|_2 \leq \delta_k \left\| \begin{pmatrix} r_1^{(k)} \\ r_2^{(k)} \end{pmatrix} \right\|_2, \end{aligned}$$

which along with (37) results in

$$\begin{aligned} \bar{x}^{(k+1)} &= \bar{x}^{(k)} + \bar{\beta}_k(\bar{z}_1^{(k)}; \bar{z}_2^{(k)}) \\ &= \bar{x}^{(k)} + \bar{\beta}_k \mathcal{M}(\alpha, \beta)^{-1}[(r_1^{(k)}; r_2^{(k)}) + (\bar{p}^{(k)}; \bar{q}^{(k)})] \\ &= \bar{x}^{(k)} + \bar{\beta}_k \mathcal{M}(\alpha, \beta)^{-1}(r_1^{(k)}; r_2^{(k)}) + \bar{\beta}_k \mathcal{M}(\alpha, \beta)^{-1}(\bar{p}^{(k)}; \bar{q}^{(k)}) \end{aligned}$$

and

$$r^{(k+1)} = r^{(k)} - \bar{\beta}_k \bar{\mathcal{A}}(\bar{z}_1^{(k)}; \bar{z}_2^{(k)}).$$

The combination of the above two equations yields the following result

$$\begin{aligned} & \|\bar{x}^{(k+1)} - x^*\|_{\bar{\mathcal{A}}} \\ &= \|r^{(k+1)}\|_2 \\ &= \|r^{(k)} - \bar{\beta}_k \bar{\mathcal{A}}(\bar{z}_1^{(k)}; \bar{z}_2^{(k)})\|_2 \\ &\leq \|r^{(k)} - \bar{\mathcal{A}}(\bar{z}_1^{(k)}; \bar{z}_2^{(k)})\|_2 \\ &= \|r^{(k)} - \bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}(r_1^{(k)}; r_2^{(k)}) - \bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}(\bar{p}^{(k)}; \bar{q}^{(k)})\|_2 \\ &\leq \|r^{(k)} - \bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}(r_1^{(k)}; r_2^{(k)})\|_2 + \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}(\bar{p}^{(k)}; \bar{q}^{(k)})\|_2 \\ &\leq \|\mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1}\|_2 \|r^{(k)}\|_2 + \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}(\bar{p}^{(k)}; \bar{q}^{(k)})\|_2 \\ &\leq \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) \|r^{(k)}\|_2 + \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2 \|(\bar{p}^{(k)}; \bar{q}^{(k)})\|_2 \\ &\leq \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) \|r^{(k)}\|_2 + \delta_k \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2 \|r^{(k)}\|_2 \\ &= (\Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) + \delta_k \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2) \|r^{(k)}\|_2 \end{aligned}$$

$$= (\Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) + \delta_k \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2) \|\bar{x}^{(k)} - x^*\|_{\bar{\mathcal{A}}}.$$

Under the condition (34), it has $\Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) < 1$. If $\{\delta_k\}$ is chosen such that

$$\Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}) + \delta_k \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2 < 1,$$

then the iterative sequence $\{\bar{x}^{(k)}\}$ converges to the exact solution x^* of (4). Moreover, in view of (35) we deduce that

$$\begin{aligned} \|\bar{\mathcal{A}}\mathcal{M}(\alpha, \beta)^{-1}\|_2 &= \|I - \mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1}\|_2 \\ &\leq \|I\|_2 + \|\mathcal{N}(\alpha, \beta)\mathcal{M}(\alpha, \beta)^{-1}\|_2 \\ &\leq 1 + \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min}), \end{aligned}$$

from which one gives the following result

$$\psi(k) \leq \Phi(\alpha, \beta, \lambda_{\max}, \lambda_{\min})(1 + \delta_k) + \delta_k.$$

Then a sufficient condition for guaranteeing $\psi(k) < 1$ is (38). Up to now, the proof has been completed. \square

Remark 5.1 Theorem 5.1 can be utilized for the inexact PTS (IPTS) iteration method due to the fact that it is the special case of the IMRPTS one.

6 Numerical experiments

In this section, we present the results of numerical experiments aimed at validating the effectiveness of the proposed methods and comparing their numerical behaviors with those of some known ones. Further, we exhibit the numerical advantages of the PTS preconditioner over the PMHSS, additive block diagonal (ABD) [12], RBLT and block triangular (BT) [29] ones. Here, the preconditioned matrix V in the PMHSS iteration method is taken as W . For the tested iteration methods, their parameters are the experimentally found optimal ones which minimize the total number of iterations. In all tables, the parameters α_{pre} and β_{pre} are the ones that satisfy $\alpha\beta = 1$ and minimize the IT of the MRPTS iteration method. Numerical results are compared in terms of both the number of iteration steps (denoted by “IT”) and the elapsed CPU time in seconds (denoted by “CPU”).

All experiments are carried out using MATLAB (version R2016b) on a personal computer with Intel (R) Pentium (R) CPU G3240T 2.870 GHz, 16.0 GB memory and Windows 10 system. In actual computations, the initial vector is taken to be the zero vector. All iterations are terminated once the current relative residual (denoted by “RES”) norm is reduced by a factor of 10^{12} or the number of iteration steps exceeds 500, or the CPU times are over 3600 s. The latter two are indicated by “–” in the numerical tables.

Table 1 Numerical results for different iteration methods for Example 6.1 when $(\varpi, \mu) = (\pi, 0.02)$

Method	m	128	256	512	1024
PMHSS	α_{exp}	0.9	0.9	0.9	0.9
	IT	77	77	77	77
	CPU	1.2922	5.3227	45.5343	211.9940
	RES	$6.94\text{e-}13$	$7.51\text{e-}13$	$7.69\text{e-}13$	$7.73\text{e-}13$
SCSP	ω_{exp}	1.37	1.38	1.39	1.40
	IT	80	78	77	75
	CPU	0.6274	2.9585	21.2163	99.5001
	RES	$7.80\text{e-}13$	$9.35\text{e-}13$	$7.81\text{e-}13$	$8.93\text{e-}13$
CRI	α_{exp}	1	1	1	1
	IT	27	26	24	23
	CPU	0.5145	2.2184	15.9938	70.0917
	RES	$4.63\text{e-}13$	$4.13\text{e-}13$	$9.27\text{e-}13$	$8.38\text{e-}13$
TMIT	α_{exp}	0.62	0.62	0.62	0.62
	IT	19	18	17	17
	CPU	0.3046	1.3647	6.1151	29.4480
	RES	$4.73\text{e-}13$	$6.51\text{e-}13$	$9.03\text{e-}13$	$3.25\text{e-}13$
MRTMIT	α_{exp}	0.62	0.61	0.60	0.60
	IT	19	17	16	17
	CPU	0.2591	1.1849	6.1693	34.3332
	RES	$1.86\text{e-}13$	$6.99\text{e-}13$	$2.73\text{e-}13$	$4.74\text{e-}13$
TTSCSP	α_{exp}	0.45	0.45	0.45	0.45
	β_{exp}	0.12	0.12	0.12	0.12
	IT	19	19	19	18
	CPU	0.3545	1.7036	11.1996	51.6492
PTS	RES	$8.82\text{e-}13$	$6.52\text{e-}13$	$6.44\text{e-}13$	$5.47\text{e-}13$
	α_{exp}	0.45	0.45	0.45	0.45
	β_{exp}	0.10	0.10	0.10	0.10
	IT	20	19	19	18
MRPTS	CPU	0.3045	1.4996	7.7722	35.3665
	RES	$6.16\text{e-}13$	$7.93\text{e-}13$	$3.04\text{e-}13$	$7.14\text{e-}13$
	α_{exp}	1	1	1	1
	β_{exp}	0.94	0.94	0.97	0.955
MRPTS	IT	17	16	15	15
	CPU	0.2827	1.3486	6.4089	29.4228
	RES	$2.65\text{e-}13$	$3.39\text{e-}13$	$5.01\text{e-}13$	$6.91\text{e-}13$
	α_{pre}	1.5	1.4	1.4	1.4
MRPTS	β_{pre}	0.6667	0.7143	0.7143	0.7143
	IT	17	17	18	18
	CPU	0.2899	1.4722	7.6980	39.1278
	RES	$4.82\text{e-}13$	$2.91\text{e-}13$	$8.58\text{e-}13$	$1.76\text{e-}13$

Table 2 When $(\sigma_1, \sigma_2) = (10, 100)$, numerical results for different iteration methods for Example 6.2

Method	m	128	256	512	1024
PMHSS	α_{exp}	1	1	1	1
	IT	80	81	81	81
	CPU	1.1685	5.7358	46.3507	230.1828
	RES	$9.58e-13$	$8.31e-13$	$8.87e-13$	$9.04e-13$
SCSP	ω_{exp}	1.36	1.38	1.39	1.39
	IT	90	89	87	86
	CPU	0.6972	3.3356	24.8934	124.1379
	RES	$8.58e-13$	$8.94e-13$	$9.99e-13$	$7.95e-13$
CRI	α_{exp}	1	1	1	1
	IT	36	34	33	31
	CPU	0.6013	2.6913	21.0440	92.8868
	RES	$5.27e-13$	$7.50e-13$	$5.32e-13$	$7.57e-13$
TMIT	α_{exp}	0.64	0.64	0.63	0.63
	IT	20	19	18	18
	CPU	0.2767	1.3588	6.4789	34.9853
	RES	$6.19e-13$	$8.04e-13$	$8.40e-13$	$2.98e-13$
MRTMIT	α_{exp}	0.57	0.59	0.60	0.60
	IT	19	17	19	19
	CPU	0.2943	1.3628	8.2398	37.1222
	RES	$3.26e-13$	$9.39e-13$	$8.24e-13$	$4.20e-13$
TTSCSP	α_{exp}	0.45	0.45	0.45	0.45
	β_{exp}	0.12	0.11	0.11	0.11
	IT	22	22	21	21
	CPU	0.3701	1.7815	12.4429	57.7445
PTS	RES	$8.48e-13$	$4.65e-13$	$7.39e-13$	$5.93e-13$
	α_{exp}	0.45	0.45	0.45	0.45
	β_{exp}	0.12	0.10	0.10	0.10
	IT	23	22	21	21
MRPTS	CPU	0.3561	1.7389	8.0597	38.9969
	RES	$7.01e-13$	$6.82e-13$	$9.26e-13$	$6.54e-13$
	α_{exp}	1	1	1	1
	β_{exp}	0.925	0.96	0.96	0.96
MRPTS	IT	17	17	17	16
	CPU	0.2802	1.3915	6.8732	31.1029
	RES	$4.95e-13$	$5.52e-13$	$8.97e-13$	$8.24e-13$
	α_{pre}	1.1	1.1	1.03	1.03
MRPTS	β_{pre}	0.9091	0.9091	0.9709	0.9709
	IT	17	17	17	16
	CPU	0.2919	1.4525	7.3702	32.9961
	RES	$3.76e-13$	$9.78e-13$	$3.39e-13$	$9.25e-13$

Table 3 Numerical results for different iteration methods for Example 6.3

Method	m	128	256	512	1024
PMHSS	α_{exp}	0.7	0.8	1	1.1
	IT	61	62	65	69
	CPU	1.3952	7.0130	60.2707	290.3031
	RES	$7.40\text{e-}13$	$9.18\text{e-}13$	$8.86\text{e-}13$	$8.28\text{e-}13$
SCSP	ω_{exp}	1.13	1.01	0.95	0.93
	IT	82	122	165	198
	CPU	0.8134	6.9190	74.9659	452.8848
	RES	$9.63\text{e-}13$	$8.97\text{e-}13$	$8.48\text{e-}13$	$9.42\text{e-}13$
CRI	α_{exp}	1	1	1	1
	IT	36	36	36	35
	CPU	0.8589	4.5805	35.5885	163.6716
	RES	$9.75\text{e-}13$	$7.46\text{e-}13$	$5.28\text{e-}13$	$7.52\text{e-}13$
TMIT	α_{exp}	0.68	0.77	0.80	0.82
	IT	23	26	27	28
	CPU	0.5068	3.0231	14.9228	76.4157
	RES	$5.24\text{e-}13$	$5.88\text{e-}13$	$6.47\text{e-}13$	$7.74\text{e-}13$
MRTMIT	α_{exp}	0.62	0.59	0.66	0.66
	IT	21	23	25	27
	CPU	0.4260	2.6903	14.3862	78.0856
	RES	$3.41\text{e-}13$	$6.17\text{e-}13$	$7.25\text{e-}13$	$9.01\text{e-}13$
TTSCSP	α_{exp}	0.36	0.24	0.17	0.14
	β_{exp}	0.22	0.22	0.22	0.22
	IT	22	27	31	34
	CPU	0.5998	3.5273	28.3587	151.4894
PTS	RES	$5.68\text{e-}13$	$8.65\text{e-}13$	$8.98\text{e-}13$	$6.17\text{e-}13$
	α_{exp}	0.37	0.24	0.17	0.14
	β_{exp}	0.23	0.22	0.22	0.22
	IT	22	27	32	36
MRPTS	CPU	0.5012	3.2822	18.3237	101.2192
	RES	$6.85\text{e-}13$	$6.82\text{e-}13$	$6.38\text{e-}13$	$5.13\text{e-}13$
	α_{exp}	1.05	1.03	1.02	1.02
	β_{exp}	0.92	0.935	0.935	0.935
MRPTS	IT	15	16	17	17
	CPU	0.3879	2.1892	10.7612	53.4980
	RES	$6.81\text{e-}13$	$4.09\text{e-}13$	$1.07\text{e-}13$	$9.37\text{e-}13$
	α_{pre}	1.08	1.07	1.07	1.05
MRPTS	β_{pre}	0.9259	0.9346	0.9346	0.9524
	IT	16	16	18	20
	CPU	0.4031	2.1111	12.1748	68.3381
	RES	$5.39\text{e-}13$	$9.95\text{e-}13$	$2.76\text{e-}13$	$4.21\text{e-}13$

Table 4 Numerical results for different iteration methods for Example 6.4

Method	m	128	256	512	1024
PMHSS	α_{exp}	1.4	1.4	1.4	1.4
	IT	43	43	43	43
	CPU	2.0576	9.6673	49.9618	232.8513
	RES	$8.21\text{e-}13$	$7.68\text{e-}13$	$7.38\text{e-}13$	$7.24\text{e-}13$
SCSP	ω_{exp}	0.6	0.6	0.6	0.6
	IT	21	21	21	21
	CPU	0.3891	1.5661	8.1872	38.2184
	RES	$4.99\text{e-}13$	$6.55\text{e-}13$	$7.61\text{e-}13$	$8.26\text{e-}13$
CRI	α_{exp}	1	1	1	1
	IT	41	41	41	41
	CPU	1.2365	6.0289	30.5952	145.7394
	RES	$8.93\text{e-}13$	$9.04\text{e-}13$	$9.08\text{e-}13$	$9.09\text{e-}13$
TMIT	α_{exp}	0.62	0.63	0.64	0.64
	IT	22	22	22	22
	CPU	0.5672	2.4507	12.6579	67.2756
	RES	$5.38\text{e-}13$	$2.94\text{e-}13$	$5.10\text{e-}13$	$5.28\text{e-}13$
MRTMIT	α_{exp}	0.67	0.72	0.68	0.65
	IT	16	17	17	16
	CPU	0.4417	2.0956	10.9334	48.0135
	RES	$7.89\text{e-}13$	$3.26\text{e-}13$	$3.32\text{e-}13$	$9.56\text{e-}13$
TTSCSP	α_{exp}	0.35	0.35	0.35	0.35
	β_{exp}	1.1	1.1	1.1	1.1
	IT	9	9	9	9
	CPU	0.3271	1.4220	7.3943	37.1068
PTS	RES	$5.46\text{e-}13$	$8.65\text{e-}13$	$1.90\text{e-}13$	$1.23\text{e-}13$
	α_{exp}	0.38	0.38	0.38	0.38
	β_{exp}	1.12	1.1	1.1	1.1
	IT	10	9	9	9
MRPTS	CPU	0.3139	1.2316	5.8765	27.3067
	RES	$3.75\text{e-}14$	$9.86\text{e-}13$	$9.23\text{e-}13$	$8.88\text{e-}13$
	α_{exp}	1	1	1	1
	β_{exp}	3.6	3.6	3.6	3.6
MRPTS	IT	9	9	9	9
	CPU	0.2897	1.2349	6.3891	28.5775
	RES	$8.96\text{e-}13$	$8.53\text{e-}13$	$6.29\text{e-}13$	$4.11\text{e-}13$
	α_{pre}	0.60	0.60	0.60	0.60
MRPTS	β_{pre}	1.6667	1.6667	1.6667	1.6667
	IT	9	9	9	10
	CPU	0.3110	1.3474	7.0470	34.5838
	RES	$9.58\text{e-}13$	$6.85\text{e-}14$	$8.98\text{e-}14$	$1.90\text{e-}13$

Table 5 Numerical results for the tested preconditioned GMRES methods for Example 6.1 when $(\varpi, \mu) = (\pi, 0.02)$

Method	m	128	256	512	1024
PMHSS-GMRES	α_{exp}	10	10	10	10
	IT	22	22	22	22
	CPU	0.6463	2.0176	9.8752	47.1681
	RES	6.26e-14	6.32e-14	6.09e-14	6.17e-14
ABD-GMRES	α_{exp}	2	2	2	2
	IT	15	15	15	15
	CPU	0.3068	1.5090	7.6157	36.5362
	RES	3.82e-13	3.57e-13	3.43e-13	3.36e-13
RBLT-GMRES	α_{exp}	1	1	1	1
	IT	12	12	12	12
	CPU	0.2440	1.1996	5.8368	26.5757
	RES	7.45e-13	7.62e-13	7.27e-13	6.83e-13
BT-GMRES	α_{exp}	1	1	1	1
	IT	8	8	9	9
	CPU	0.2799	1.3940	7.7004	36.2910
	RES	5.86e-13	6.75e-13	5.24e-14	7.26e-14
PTS-GMRES	α_{exp}	25	25	25	25
	β_{exp}	0.1	0.1	0.1	0.1
	IT	7	7	7	7
	CPU	0.1914	0.9720	4.9159	22.5076
	RES	2.94e-14	3.96e-14	4.76e-14	5.67e-14

Example 6.1 We consider the following complex symmetric linear system [9,10]:

$$[(-\varpi^2 M + K) + i(\varpi C_V + C_H)]x = b,$$

where M and K are the inertia and the stiffness matrices, C_V and C_H are the viscous and the hysteretic damping matrices, respectively, and ϖ is the driving circular frequency. We take $C_H = \mu K$ with μ a damping coefficient, $M = I$, $C_V = 10I$, and K the five-point centered difference matrix approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square $[0, 1] \times [0, 1]$ with the mesh-size $h = 1/(m + 1)$. The matrix $K \in \mathbb{R}^{n \times n}$ possesses the tensor-product form $K = I \otimes V_m + V_m \otimes I$, with $V_m = h^{-2} \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$. Hence, K is an $n \times n$ block-tridiagonal matrix, with $n = m^2$. In addition, we set the right-hand side vector b to be $b = (1 + i)A\mathbf{1}$, with $\mathbf{1}$ being the vector of all entries equal to 1. The linear system is normalized by multiplying both sides with h^2 .

Example 6.2 Consider the complex Helmholtz equation [22,36,39]:

$$-\Delta u + \sigma_1 u + i\sigma_2 u = f$$

Table 6 When $(\sigma_1, \sigma_2) = (10, 100)$, numerical results for different tested preconditioned GMRES methods for Example 6.2

Method	m	128	256	512	1024
PMHSS-GMRES	α_{exp}	10	10	10	10
	IT	27	27	27	27
	CPU	0.5379	2.4658	12.0660	60.7561
	RES	7.95e-13	8.69e-13	8.50e-13	8.01e-13
ABD-GMRES	α_{exp}	3	3	3	3
	IT	19	19	19	19
	CPU	0.3383	1.8114	9.4279	44.8177
	RES	3.02e-13	1.98e-13	1.57e-13	2.08e-13
RBLT-GMRES	α_{exp}	1	1	1	1
	IT	16	16	16	16
	CPU	0.3231	1.5035	7.0927	34.0243
	RES	9.21e-14	1.03e-13	1.03e-13	9.77e-14
BT-GMRES	α_{exp}	0.2	0.25	0.25	0.26
	IT	11	11	11	11
	CPU	0.3452	1.7248	8.8147	42.4493
	RES	4.65e-14	5.12e-14	5.60e-13	4.76e-13
PTS-GMRES	α_{exp}	10	10	10	10
	β_{exp}	0.2	0.2	0.2	0.18
	IT	9	9	9	8
	CPU	0.2122	1.0711	5.0687	21.7177
	RES	5.63e-14	5.41e-14	8.74e-13	6.64e-13

with σ_1 and σ_2 being real coefficient functions. Here, u satisfies Dirichlet boundary conditions in the square $D = [0, 1] \times [0, 1]$. By discretizing this equation with finite differences on an $m \times m$ grid with mesh size $h = 1/(m + 1)$, we obtain a complex linear system

$$[(K + \sigma_1 I) + i\sigma_2 I]x = b,$$

where the matrix $K \in \mathbb{R}^{m \times m}$ possesses the tensor-product form

$$K = I \otimes B_m + B_m \otimes I \text{ with } B_m = \frac{1}{h^2} \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}.$$

Actually, K is the five-point centered difference matrix approximating the negative Laplacian operator $L = -\Delta$. In our tests, let the right-hand side vector $b = (1 + i)A\mathbf{1}$ with $\mathbf{1}$ being the vector of all entries are equal to 1. As before, we normalize the complex linear system by multiplying both sides by h^2 .

Table 7 Numerical results for the tested preconditioned GMRES methods for Example 6.3

Method	m	128	256	512	1024
PMHSS-GMRES	α_{exp}	10	1	1	1
	IT	27	27	27	28
	CPU	0.7345	3.7177	18.3317	94.7583
	RES	6.47e-13	5.50e-13	9.92e-13	5.03e-13
ABD-GMRES	α_{exp}	3	3	3	3
	IT	24	32	44	58
	CPU	0.7191	4.7093	36.6458	222.9774
	RES	1.57e-13	6.78e-13	3.30e-13	7.06e-13
RBLT-GMRES	α_{exp}	1	1	1	1
	IT	14	14	14	15
	CPU	0.4030	1.9504	9.5837	49.6874
	RES	6.19e-13	3.99e-13	7.56e-13	9.45e-14
BT-GMRES	α_{exp}	0.3	0.5	0.6	0.8
	IT	11	13	15	16
	CPU	0.4833	2.9832	17.4067	79.3832
	RES	2.55e-13	3.84e-13	3.69e-13	5.46e-13
PTS-GMRES	α_{exp}	10	5	3	2
	β_{exp}	0.5	1	1	1
	IT	9	11	13	15
	CPU	0.2994	1.7480	9.7903	53.9726
	RES	8.12e-13	5.00e-13	5.03e-13	1.70e-13

Example 6.3 [9,10] Consider the linear system of equations $(W + iT)x = b$, with

$$T = I \otimes V + V \otimes I \text{ and } W = 10(I \otimes V_c + V_c \otimes I) + 9(e_1 e_m^T + e_m e_1^T) \otimes I,$$

where $V = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$, $V_c = V - e_1 e_m^T - e_m e_1^T \in \mathbb{R}^{m \times m}$ and e_1 and e_m are the first and last unit vectors in \mathbb{R}^m , respectively. We take the right-hand side vector b to be $b = (1 + i)A\mathbf{1}$, with $\mathbf{1}$ being the vector of all entries equal to 1.

Here T and W correspond to the five-point centered difference matrices approximating the negative Laplacian operator with homogeneous Dirichlet boundary conditions and periodic boundary conditions, respectively, on a uniform mesh in the unit square $[0, 1] \times [0, 1]$ with the mesh-size $h = \frac{1}{m+1}$.

Example 6.4 [9,10] Consider the linear system of equations

$$\left[\left(K + \frac{3 - \sqrt{3}}{\tau} I \right) + i \left(K + \frac{3 + \sqrt{3}}{\tau} I \right) \right] x = b,$$

where τ is the time step-size and K is the five-point centered difference matrix approximating the negative Laplacian operator $L = -\Delta$ with homogeneous Dirich-

Table 8 Numerical results for the tested preconditioned GMRES methods for Example 6.4

Method	m	128	256	512	1024
PMHSS-GMRES	α_{exp}	0.8	1	1	1
	IT	19	21	21	21
	CPU	0.8336	4.0443	19.1729	86.8263
	RES	9.15e-13	2.15e-13	3.62e-13	3.26e-13
ABD-GMRES	α_{exp}	1	0.9	0.7	0.45
	IT	36	38	39	40
	CPU	1.6088	9.3486	41.7325	189.9543
	RES	8.05e-13	4.61e-13	9.53e-13	2.99e-13
RBLT-GMRES	α_{exp}	1	1	1	1
	IT	11	11	11	11
	CPU	0.4322	1.8226	8.7787	40.4093
	RES	8.04e-14	1.97e-13	2.62e-13	1.83e-13
BT-GMRES	α_{exp}	1.5	1.5	1.5	1.5
	IT	10	10	10	10
	CPU	0.5510	2.4560	11.6316	53.5778
	RES	5.32e-14	1.86e-13	5.34e-13	8.96e-13
PTS-GMRES	α_{exp}	1.5	1.4	1.4	1.4
	β_{exp}	6.6	8	8	8.3
	IT	8	8	8	8
	CPU	0.3598	1.6274	8.3472	34.8529
	RES	8.56e-13	6.40e-13	9.08e-13	9.78e-13

let boundary conditions, on a uniform mesh in the unit square $[0, 1] \times [0, 1]$ with the mesh-size $h = \frac{1}{m+1}$. The matrix $K \in \mathbb{R}^{n \times n}$ possesses the tensor-product form $K = I \otimes V_m + V_m \otimes I$, with $V_m = h^{-2} \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$. Hence, K is an $n \times n$ block-tridiagonal matrix, with $n = m^2$. We take

$$W = K + \frac{3 - \sqrt{3}}{\tau} I \text{ and } T = K + \frac{3 + \sqrt{3}}{\tau} I,$$

and the right-hand side vector b with its j th entry b_j being given by

$$b_j = \frac{(1 - i)j}{\tau(j + 1)^2}, \quad j = 1, 2, \dots, n.$$

In our tests, we take $\tau = h$. Besides, we normalize coefficient matrix and right-hand side by multiplying both by h^2 .

For all the tested exact iteration methods, we apply the Cholesky factorization incorporated with the symmetric approximate minimum degree reordering [30] for solving the sub-systems. To do so, the `symamd.m` command of Matlab [32] is used here.

Table 9 Numerical results for the tested inexact iteration methods for Example 6.1 when $(\varpi, \mu) = (\pi, 0.02)$

Method	m	128	256	512	1024
IPMHSS	α_{exp}	1	1	1	0.9
	IT	78	79	79	79
	CPU	1.5407	6.1557	43.7370	229.8173
	RES	7.69e-13	7.55e-13	7.68e-13	8.81e-13
ISCSP	ω_{exp}	1.45	1.49	1.50	1.50
	IT	69	68	69	71
	CPU	0.5260	2.7028	31.0392	170.0237
	RES	9.97e-13	7.02e-13	7.99e-13	7.10e-13
IPGSOR	α_{exp}	0.85	0.85	0.85	0.85
	ω_{exp}	1.7	1.7	1.65	1.75
	IT	18	18	18	18
	CPU	0.4626	2.4784	14.7452	95.8220
	RES	1.94e-13	3.85e-13	7.83e-13	2.92e-13
ITMIT	α_{exp}	0.64	0.63	0.63	0.64
	IT	20	19	19	18
	CPU	0.5998	2.8607	20.4515	113.6930
	RES	5.49e-13	8.15e-13	3.32e-13	7.62e-13
ITTSCSP	α_{exp}	0.49	0.5	0.5	0.5
	β_{exp}	0.14	0.14	0.15	0.15
	IT	24	24	25	25
	CPU	0.8768	5.9890	72.8336	518.9691
	RES	6.24e-13	6.66e-13	8.93e-13	7.50e-13
IPTS	α_{exp}	0.47	0.52	0.54	0.55
	β_{exp}	0.135	0.14	0.15	0.15
	IT	24	26	26	26
	CPU	0.7987	4.3849	40.3469	346.9023
	RES	9.17e-13	4.00e-13	6.82e-13	6.45e-13
IMRPTS	α_{exp}	1.95	1.93	2	2.1
	β_{exp}	0.9	0.93	0.95	0.98
	IT	18	19	19	19
	CPU	0.3837	2.0781	13.6492	72.2002
	RES	3.14e-13	6.92e-13	6.19e-13	9.10e-13

The experimentally found optimal parameters, IT and CPU times of the tested exact iteration methods for Examples 6.1–6.4 with respect to different problem sizes are listed in Tables 1, 2, 3, 4, 5, 6, 7 and 8. From Tables 1, 2, 3 and 4, we observe that the PTS iteration method outperforms the PMHSS, SCSP and CRI ones, and the advantage of the PTS one becomes more pronounced as the system size increases. By comparing with the PTS and the TTSCSP iteration methods, we find that their IT are almost the

Table 10 When $(\sigma_1, \sigma_2) = (10, 100)$, numerical results for different inexact iteration methods for Example 6.2

Method	m	128	256	512	1024
IPMHSS	α_{exp}	1.1	0.9	1	1
	IT	81	82	83	84
	CPU	0.9684	4.3253	30.9399	168.8756
	RES	9.68e-13	9.63e-13	9.38e-13	7.93e-13
ISCSP	ω_{exp}	1.39	1.39	1.43	1.45
	IT	83	85	83	82
	CPU	0.4926	2.0362	22.4343	114.9036
	RES	8.56e-13	9.32e-13	7.79e-13	8.56e-13
IPGSOR	α_{exp}	0.85	0.85	0.85	0.85
	ω_{exp}	1.65	1.65	1.65	1.65
	IT	18	18	18	18
	CPU	0.2906	1.5107	10.0339	51.4547
	RES	3.53e-13	6.42e-13	4.17e-13	8.22e-13
ITMIT	α_{exp}	0.63	0.63	0.63	0.63
	IT	21	20	19	19
	CPU	0.3499	1.8590	12.7252	72.6571
	RES	6.91e-13	9.58e-13	8.75e-13	4.06e-13
ITTSCSP	α_{exp}	0.43	0.43	0.43	0.44
	β_{exp}	0.1	0.1	0.1	0.1
	IT	24	25	25	25
	CPU	0.4688	2.4933	27.5653	158.0234
	RES	4.94e-13	4.27e-13	8.22e-13	7.73e-13
IPTS	α_{exp}	0.43	0.43	0.43	0.44
	β_{exp}	0.1	0.1	0.1	0.1
	IT	25	25	25	25
	CPU	0.4692	2.2133	14.1735	71.4201
	RES	7.69e-13	6.31e-13	5.04e-13	7.64e-13
IMRPTS	α_{exp}	1.9	1.7	1.7	1.7
	β_{exp}	0.95	0.9	0.85	0.82
	IT	19	19	19	20
	CPU	0.2782	1.4461	8.9909	49.7253
	RES	7.88e-13	7.08e-13	9.61e-13	6.80e-13

same. This is owing to the fact that they have the same optimal convergence factors as remarked in Sect. 2. While the PTS iteration method requires less CPU times than the latter one. The reason is that in the TTSCSP iteration method we deal with complex arithmetic whereas in the PTS one we deal only with real arithmetic. The MRPTS iteration method performs better than the PMHSS, SCSP, CRI and PTS ones, and its IT is almost unchanged even decreases with respect to m . Meanwhile, the numerical

Table 11 Numerical results for different inexact iteration methods for Example 6.3

Method	m	128	256	512	1024
IPMHSS	α_{exp}	0.7	0.7	0.9	1
	IT	65	66	68	70
	CPU	2.8161	16.7937	151.8701	962.7204
	RES	$8.77\text{e-}13$	$9.90\text{e-}13$	$8.52\text{e-}13$	$9.78\text{e-}13$
ISCSP	ω_{exp}	1.23	1.17	1.14	1.10
	IT	81	95	101	120
	CPU	2.3751	16.2250	229.4278	1.3692e+03
	RES	$7.02\text{e-}13$	$7.51\text{e-}13$	$9.26\text{e-}13$	$8.24\text{e-}13$
IPGSOR	α_{exp}	0.85	0.85	0.83	0.81
	ω_{exp}	1.3	1.25	1.25	1.25
	IT	19	20	22	23
	CPU	1.2541	7.5418	66.5938	419.1550
	RES	$4.02\text{e-}13$	$6.22\text{e-}13$	$7.13\text{e-}13$	$9.25\text{e-}13$
ITMIT	α_{exp}	0.68	0.77	0.81	0.84
	IT	21	22	24	24
	CPU	1.3878	8.4091	68.9586	419.0472
	RES	$6.12\text{e-}13$	$8.97\text{e-}13$	$3.45\text{e-}13$	$5.43\text{e-}13$
ITTSCSP	α_{exp}	0.4	0.34	0.32	0.31
	β_{exp}	0.17	0.17	0.17	0.17
	IT	21	23	24	25
	CPU	2.1806	12.2523	172.6473	915.1399
	RES	$7.19\text{e-}13$	$6.89\text{e-}13$	$7.75\text{e-}13$	$5.50\text{e-}13$
IPTS	α_{exp}	0.41	0.28	0.26	0.22
	β_{exp}	0.17	0.18	0.2	0.2
	IT	22	30	37	43
	CPU	1.6197	13.4279	147.9300	944.9223
	RES	$9.06\text{e-}13$	$8.32\text{e-}13$	$9.31\text{e-}13$	$6.88\text{e-}07$
IMRPTS	α_{exp}	1.7	1.8	1	1
	β_{exp}	0.95	1	0.94	0.96
	IT	19	21	23	23
	CPU	1.3415	9.4876	65.7000	408.5504
	RES	$6.97\text{e-}13$	$8.15\text{e-}13$	$8.17\text{e-}13$	$6.82\text{e-}13$

behaviors of the MRPTS, MRTMIT (The minimum residual version of TMIT) and TMIT iteration methods are comparable for Examples 6.1 and 6.2, while the former is the most efficient method for Examples 6.3 and 6.4. In addition, it is apparent that the experimentally found optimal parameters of the MRPTS method are stable, which brings great convenience for us to find those for large systems. As the numerical results of Tables 5, 6, 7 and 8 show, the computing efficiency of the PTS preconditioner is superior to the other ones in terms of IT and CPU times.

Table 12 Numerical results for different inexact iteration methods for Example 6.4

Method	m	128	256	512	1024
IPMHSS	α_{exp}	1.1	1.1	1.2	1.2
	IT	44	43	43	43
	CPU	0.4967	1.8970	11.1667	49.3969
	RES	$7.36\text{e-}13$	$8.85\text{e-}13$	$6.96\text{e-}13$	$7.92\text{e-}13$
ISCSP	ω_{exp}	0.61	0.61	0.62	0.62
	IT	21	21	21	21
	CPU	0.1365	0.5524	5.1019	23.2132
	RES	$5.79\text{e-}13$	$6.69\text{e-}13$	$6.91\text{e-}13$	$8.07\text{e-}13$
IPGSOR	α_{exp}	0.85	0.85	0.85	0.85
	ω_{exp}	1.02	1.02	1.02	1.02
	IT	18	18	18	17
	CPU	0.1865	0.7953	4.5397	19.1206
	RES	$3.44\text{e-}13$	$2.29\text{e-}13$	$2.52\text{e-}13$	$6.50\text{e-}13$
ITMIT	α_{exp}	0.62	0.62	0.62	0.62
	IT	25	26	26	26
	CPU	0.3397	1.4859	7.9846	31.9640
	RES	$7.28\text{e-}13$	$5.76\text{e-}13$	$4.40\text{e-}13$	$7.71\text{e-}13$
ITTSCSP	α_{exp}	0.34	0.34	0.34	0.34
	β_{exp}	1.12	1.12	1.05	1.05
	IT	10	10	10	10
	CPU	0.1342	0.5078	4.7300	22.2256
	RES	$2.66\text{e-}13$	$8.12\text{e-}13$	$5.74\text{e-}13$	$3.10\text{e-}13$
IPTS	α_{exp}	0.36	0.36	0.36	0.36
	β_{exp}	1	1	1	1
	IT	14	14	14	14
	CPU	0.1682	0.6762	3.7657	16.8961
	RES	$1.39\text{e-}13$	$2.13\text{e-}13$	$3.24\text{e-}13$	$4.88\text{e-}13$
IMRPTS	α_{exp}	0.98	0.98	0.98	0.98
	β_{exp}	3.3	3.3	3.3	3.3
	IT	13	13	13	13
	CPU	0.1674	0.6586	3.7577	16.9447
	RES	$8.81\text{e-}14$	$2.12\text{e-}13$	$5.28\text{e-}13$	$3.85\text{e-}13$

The numerical performances of the MRPTS iteration method with the parameters α_{pre} and β_{pre} and with the experimentally found optimal parameters are comparable. Thus, the strategy for choosing the parameters α_{pre} and β_{pre} of the MRPTS iteration method can be applicable for the practical problems.

Besides, Tables 9, 10, 11 and 12 report the numerical results of the tested inexact iteration methods. We adopt the PCG method as the inner solver for the tested inexact methods. More concretely, we employ the modified incomplete Cholesky factor-

ization (Matlab code: `ichol(C, struct('michol', 'on', 'type', 'ict', 'droptol', 1e-1))`) as the preconditioner in the PCG method, as it was used in [32]. The stopping tolerance used for the inner PCG iteration method is 10^{-1} . Note that the parameters adopted in Table 9 are the ones that minimize the IT of the tested methods without the condition that the diagonal elements of the matrix is nonpositive when applying the function `ichol` of Matlab for Example 6.1.

By comparing the results in Tables 9, 10, 11 and 12, we can see that, the IPTS iteration method is more efficient than the ITTSCSP one for most cases from the CPU times point of view. For Examples 6.1 and 6.2, the IMRPTS iteration method returns better numerical results than the other tested ones. The exception is that the IT of the ITMIT and the IPGSOR iteration methods are less than that of the IMRPTS one, while the latter outperforms the ITMIT and the IPGSOR ones with respect to the CPU times. For Examples 6.3 and 6.4, the IMRPTS iteration method performs the best among the tested ones. Another important fact which can be pointed out is that the convergence behavior of the IMRPTS iteration method is insensitive to m and almost independent to the problem size.

7 Concluding remarks

In the current paper, we first establish a preconditioned triangular splitting (PTS) iteration method, which avoids involving complex arithmetic compared with the TTSCSP one derived in [32]. Besides, by using the minimum residual technique put forward in [38], we construct a minimum residual PTS (MRPTS) iteration method which exhibits higher calculation efficiency than the PTS one. The inexact version of the MRPTS iteration method is also derived. We establish the convergence theories for the proposed iteration methods and investigate the spectral properties of the PTS-preconditioned matrix in detail. Finally, numerical experiments validate the effectiveness of the proposed methods, and they are superior to some known ones in terms of the iterations and CPU times.

However, in this work the parameters of the proposed iteration methods and preconditioner in the numerical experiments are the experimentally found optimal ones or the proper ones satisfying $\alpha\beta = 1$. The choice of the optimal parameters of them has not been derived, which needs further discussions in our future work.

Acknowledgements We would like to express our sincere thanks to the editor and the anonymous reviewer for their valuable suggestions and constructive comments which greatly improved the presentation of this paper.

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. Axelsson, O., Neytcheva, M.G., Ahmad, B.: A comparison of iterative methods to solve complex valued linear algebraic systems. *Numer. Algorithms* **66**, 811–841 (2014)

3. Axelsson, O., Salkuyeh, D.K.: A new version of a preconditioning method for certain two-by-two block matrices with square blocks. *BIT Numer. Math.* (2018). <https://doi.org/10.1007/s10543-018-0741-x>
4. Bai, Z.-Z.: Sharp error bounds of some Krylov subspace methods for non-Hermitian linear systems. *Appl. Math. Comput.* **109**, 273–285 (2000)
5. Bai, Z.-Z.: Block preconditioners for elliptic PDE-constrained optimization problems. *Computing* **91**, 379–395 (2011)
6. Bai, Z.-Z.: Rotated block triangular preconditioning based on PMHSS. *Sci. China Math.* **56**, 2523–2538 (2013)
7. Bai, Z.-Z.: Motivations and realizations of Krylov subspace methods for large sparse linear systems. *J. Comput. Appl. Math.* **283**, 71–78 (2015)
8. Bai, Z.-Z.: On preconditioned iteration methods for complex linear systems. *J. Eng. Math.* **93**, 41–60 (2015)
9. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. *Computing* **87**, 93–111 (2010)
10. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. *Numer. Algorithms* **56**, 297–317 (2011)
11. 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.* **33**, 343–369 (2013)
12. Bai, Z.-Z., Chen, F., Wang, Z.-Q.: Additive block diagonal preconditioning for block two-by-two linear systems of skew-Hamiltonian coefficient matrices. *Numer. Algorithms* **62**, 655–675 (2013)
13. 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)
14. 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)
15. Bai, Z.-Z., Parlett, B.N., Wang, Z.-Q.: On generalized successive overrelaxation methods for augmented linear systems. *Numer. Math.* **102**, 1–38 (2005)
16. Bai, Z.-Z., Wang, Z.-Q.: On parameterized inexact Uzawa methods for generalized saddle point problems. *Linear Algebra Appl.* **428**, 2900–2932 (2008)
17. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. *Acta Numer.* **14**, 1–137 (2005)
18. Bertaccini, D.: Efficient preconditioning for sequences of parametric complex symmetric linear systems. *Electron. Trans. Numer. Anal.* **18**, 49–64 (2004)
19. Cao, Y., Jiang, M.-Q., Zheng, Y.-L.: A splitting preconditioner for saddle point problems. *Numer. Linear Algebra Appl.* **18**, 875–895 (2011)
20. Chen, F.: On choices of iteration parameter in HSS method. *Appl. Math. Comput.* **271**, 832–837 (2015)
21. Dehghan, M., Dehghani-Madiseh, M., Hajarian, M.: A generalized preconditioned MHSS method for a class of complex symmetric linear systems. *Math. Model. Anal.* **18**, 561–576 (2013)
22. 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.* **22**, 761–776 (2015)
23. Hezari, D., Salkuyeh, D.K., Edalatpour, V.: A new iterative method for solving a class of complex symmetric system of linear equations. *Numer. Algorithms* **73**, 927–955 (2016)
24. Huang, Z.-G., Wang, L.-G., Xu, Z., Cui, J.-J.: An efficient two-step iterative method for solving a class of complex symmetric linear systems. *Comput. Math. Appl.* **75**, 2473–2498 (2018)
25. Huang, Z.-G., Wang, L.-G., Xu, Z., Cui, J.-J.: Preconditioned accelerated generalized successive overrelaxation method for solving complex symmetric linear systems. *Comput. Math. Appl.* **77**, 1902–1916 (2019)
26. Li, X., Yang, A.-L., Wu, Y.-J.: Lopsided PMHSS iteration method for a class of complex symmetric linear systems. *Numer. Algorithms* **66**, 555–568 (2014)
27. Li, X.-A., Zhang, W.-H., Wu, Y.-J.: On symmetric block triangular splitting iteration method for a class of complex symmetric system of linear equations. *Appl. Math. Lett.* **79**, 131–137 (2018)
28. Liang, Z.-Z., Zhang, G.-F.: On SSOR iteration method for a class of block two-by-two linear systems. *Numer. Algorithms* **71**, 655–671 (2016)
29. Liao, L.-D., Zhang, G.-F.: A note on block diagonal and block triangular preconditioners for complex symmetric linear systems. *Numer. Algorithms* **80**, 1143–1154 (2019)
30. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia (2003)

31. Salkuyeh, D.K., Hezari, D., Edalatpour, V.: Generalized successive overrelaxation iterative method for a class of complex symmetric linear system of equations. *Int. J. Comput. Math.* **92**, 802–815 (2015)
32. Salkuyeh, D.K., Siahkolaei, T.S.: Two-parameter TSCSP method for solving complex symmetric system of linear equations. *Calcolo* **55**, 8 (2018)
33. Wang, T., Lu, L.-Z.: Alternating-directional PMHSS iteration method for a class of two-by-two block linear systems. *Appl. Math. Lett.* **58**, 159–164 (2016)
34. Wang, T., Zheng, Q.-Q., Lu, L.-Z.: A new iteration method for a class of complex symmetric linear systems. *J. Comput. Appl. Math.* **325**, 188–197 (2017)
35. Wu, S.-L., Li, C.-X.: A splitting method for complex symmetric indefinite linear system. *J. Comput. Appl. Math.* **313**, 343–354 (2017)
36. Xiao, X.-Y., Wang, X.: A new single-step iteration method for solving complex symmetric linear systems. *Numer. Algorithms* **78**, 643–660 (2018)
37. Xiao, X.-Y., Wang, X., Yin, H.-W.: Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems. *Comput. Math. Appl.* **74**, 2269–2280 (2017)
38. Yang, A.-L., Cao, Y., Wu, Y.-J.: Minimum residual Hermitian and skew-Hermitian splitting iteration method for non-Hermitian positive definite linear systems. *BIT Numer. Math.* **59**, 299–319 (2019)
39. Zeng, M.-L., Ma, C.-F.: A parameterized SHSS iteration method for a class of complex symmetric system of linear equations. *Comput. Math. Appl.* **71**, 2124–2131 (2016)
40. Zhang, J.-H., Dai, H.: A new block preconditioner for complex symmetric indefinite linear systems. *Numer. Algorithms* **74**, 889–903 (2017)
41. Zhang, J.-H., Wang, Z.-W., Zhao, J.: Preconditioned symmetric block triangular splitting iteration method for a class of complex symmetric linear systems. *Appl. Math. Lett.* **86**, 95–102 (2018)
42. 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.* **73**, 91–97 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.