Inexact rotated block triangular preconditioners for a class of block two-by-two matrices

Chao Lang · Zhi-Ru Ren

Received: 13 June 2013 / Accepted: 2 October 2013 / Published online: 27 November 2013

Abstract Based on rotated block triangular preconditioners proposed by Bai in (Sci China Math 56, 2013), we give a class of inexact rotated block triangular preconditioners for block two-by-two matrices of real square blocks W and T, which avoid inversion of the matrix $\alpha W+T$ exactly at each step of solving the linear systems. The spectral properties of the corresponding preconditioned matrices are analyzed. Numerical results show that these inexact rotated block triangular preconditioners are more effective than the exact ones when they are used to accelerate Krylov subspace iteration methods for solving block two-by-two linear systems.

Keywords Block two-by-two matrix · Block triangular preconditioning · Inexact preconditioning · Incomplete factorization

1 Introduction

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

$$Ax \equiv \begin{pmatrix} W & -T \\ T & W \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \equiv b, \tag{1.1}$$

where the matrix $A \in \mathbb{R}^{2m\times 2m}$ has a block two-by-two structure with $W \in \mathbb{R}^{m\times m}$ and $T \in \mathbb{R}^{m\times m}$. Note that the matrix A is nonsingular if and only if $\text{null}(W)\cap\text{null}(T) = \{0\}$ and $\pm\mathrm{i}$ is not a generalized eigenvalue of the matrix pair (W, T) (i.e., $Tx \neq \mp\mathrm{i}Wx$ for some $x \neq 0$), where $\text{null}(\cdot)$ denotes the nullspace of the corresponding matrix and $\mathrm{i} = \sqrt{-1}$ is the imaginary unit. Many practical problems arising from scientific computing and engineering applications may require the solution of a linear system of the form (1.1), for example, computational electrodynamics [1,2], optimization [3–7], optimal control [8–10], and real equivalent formulations of complex linear systems [11–13]; see also [14–17].

A number of iterative methods have been proposed for the block two-by-two linear system (1.1) such as C-to-R iteration methods [11], alternating splitting iteration methods [18–20], preconditioned Krylov subspace iteration

C. Lang · Z.-R. Ren (✉)
State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P. O. Box 2719,
Beijing 100190, People's Republic of China
e-mail: renzr@lsec.cc.ac.cn

methods [21], and so on; see also [22–24]. To solve the block two-by-two linear system (1.1) effectively, Bai et al., in [9], established a class of *preconditioned and modified Hermitian and skew-Hermitian splitting* (PMHSS) iteration methods by modifying and preconditioning the *Hermitian and skew-Hermitian splitting* (HSS) iteration method [18,25,26]. The PMHSS methods are essentially a kind of splitting iteration method, and the splitting matrices

$$F(\alpha) = \left[1 + \frac{1}{\alpha}\right] Q \begin{pmatrix} \alpha W + T & 0 \\ 0 & \alpha W + T \end{pmatrix}, \quad \text{with} \quad Q = \frac{1}{2}\begin{pmatrix} I & -I \\ I & I \end{pmatrix}, \tag{1.2}$$

can be used as preconditioners called PMHSS preconditioners for the matrix A in (1.1), where α is a given positive constant and I is the identity matrix. The authors established the convergence theory of these PMHSS methods and analyzed the spectral properties of the PMHSS preconditioned matrix $F(\alpha)^{-1}A$ when the matrices W and T were symmetric positive semidefinite in [9]. Numerical experiments showed that the PMHSS preconditioners were quite competitive when used to precondition Krylov subspace iteration methods such as the generalized minimal residual (GMRES) method.

Based on structures of the PMHSS preconditioners, Bai in [12] recently constructed a class of rotated block triangular preconditioners, called *rotated block lower triangular* (RBLT), *rotated block upper triangular* (RBUT), and *rotated block triangular product* (RBTP) preconditioners, for a block two-by-two linear system of the form (1.1). These rotated block triangular preconditioners are applicable not only to symmetric positive semidefinite matrices W and T but also in cases where either W or T is nonsymmetric. The author analyzed the eigenproperties and derived bounds for the degrees of the minimal polynomials of the preconditioned matrices. It was shown that the GMRES iteration method incorporated with these rotated block triangular precondtioners could be competitive with and even more efficient than that incorporated with the PMHSS preconditioner.

Note that the rotated block triangular preconditioning processes in [12] involve solving linear subsystems of the coefficient matrix $\alpha W + T$. Thus, it is necessary to invert $\alpha W + T$ at each step of the preconditioning processes. In many practical applications, the matrices W and T are very large and sparse, and much more computing time is required to invert the matrix $\alpha W + T$ exactly. By making use of the idea of inexact preconditioning [25,27,28], in this paper we present a class of inexact rotated block triangular preconditioners for a block two-by-two linear system of the form (1.1), which avoids inversion of the matrix $\alpha W + T$ exactly at each step of solving the linear subsystems. We can take advantage of the structures of the matrices W and T to solve the linear subsystems inexactly. For example, we use incomplete Cholesky factorization when the matrix $\alpha W + T$ is symmetric positive definite or incomplete LU factorization when the matrix $\alpha W + T$ is nonsymmetric. Then the spectral properties of these inexact preconditioned matrices are analyzed, and the effectiveness of the GMRES iteration method incorporated with these inexact rotated block triangular preconditioners is shown when compared with the exact ones.

The outline of this paper is as follows. In Sect. 2 we present the inexact rotated block triangular preconditioners and describe procedures for computing the generalized residual equations with these preconditioners. The spectral properties of the preconditioned matrices are analyzed in Sect. 3. In Sect. 4, we use numerical results to show the effectiveness of these inexact preconditioners. Finally, in Sect. 5, we end the paper with some concluding remarks.

2 Inexact rotated triangular preconditioning

Let α be a real constant. In [12] the author constructed rotated block triangular preconditioners for the block two-by-two matrix A in (1.1). When the matrices W and T are large and sparse, it is costly to compute the inverse of the matrix $\alpha W + T$ exactly. Assume that the matrix $\alpha\widetilde{W + T}$, an approximation to the matrix $\alpha W + T$, is nonsingular. To save computing time, we present three inexact rotated block triangular preconditioners by following the approach in [12], which are an *inexact rotated block lower triangular* (IRBLT) preconditioner

$$\widetilde{L}(\alpha) = \frac{1}{\sqrt{\alpha^2 + 1}} G \begin{pmatrix} \alpha\widetilde{W + T} & 0 \\ \alpha T - W & \alpha\widetilde{W + T} \end{pmatrix}, \tag{2.1}$$

an *inexact rotated block upper triangular* (IRBUT) preconditioner

$$\widetilde{U}(\alpha) = \frac{1}{\sqrt{\alpha^2+1}} G \begin{pmatrix} \widetilde{\alpha W + T} & W - \alpha T \\ 0 & \widetilde{\alpha W + T} \end{pmatrix}, \tag{2.2}$$

and an *inexact rotated block triangular product* (IRBTP) preconditioner

$$\widetilde{P}(\alpha) = \frac{1}{\sqrt{\alpha^2+1}} G \begin{pmatrix} \widetilde{\alpha W + T} & 0 \\ \alpha T - W & \widetilde{\alpha W + T} \end{pmatrix} \begin{pmatrix} \widetilde{\alpha W + T} & 0 \\ 0 & \widetilde{\alpha W + T} \end{pmatrix}^{-1} \begin{pmatrix} \widetilde{\alpha W + T} & W - \alpha T \\ 0 & \widetilde{\alpha W + T} \end{pmatrix} \tag{2.3}$$

for the matrix A, where the matrix

$$G = \frac{1}{\sqrt{2}} \begin{pmatrix} I & -I \\ I & I \end{pmatrix}$$

is a block Givens rotation. Note that $G = \sqrt{2} Q$, with Q being the matrix defined in (1.2). It is expected that these inexact preconditioners will be more effective than the exact ones in [12].

When these inexact rotated block triangular preconditioners are used to accelerate the convergence rate of Krylov subspace iteration methods, it is necessary to solve sequences of generalized residual equations of the forms

$$\widetilde{L}(\alpha) v = r, \quad \widetilde{U}(\alpha) v = r, \quad \text{and} \quad \widetilde{P}(\alpha) v = r, \tag{2.4}$$

where $r = (r_a^{\mathrm{T}}, r_b^{\mathrm{T}})^{\mathrm{T}}$ and $v = (v_a^{\mathrm{T}}, v_b^{\mathrm{T}})^{\mathrm{T}}$ are the current and generalized residual vectors, respectively. Here and in the sequel, $(\cdot)^{\mathrm{T}}$ denotes the transpose of a vector or a matrix. As these inexact preconditioners $\widetilde{L}(\alpha)$, $\widetilde{U}(\alpha)$, and $\widetilde{P}(\alpha)$ are multiplicative structures, the generalized residual equations in (2.4) can be accomplished by solving only subsystems of the same coefficient matrix $\widetilde{\alpha W + T}$. Thus, we can compute the generalized residual vectors v in (2.4) by the following procedures.

Procedures for Computing the Generalized Residuals

Set $\tilde{r}_a = \sqrt{\frac{\alpha^2+1}{2}}(r_a + r_b)$ and $\tilde{r}_b = \sqrt{\frac{\alpha^2+1}{2}}(-r_a + r_b)$.

- Computing v from $\widetilde{L}(\alpha) v = r$:
  - Solve v_a from $(\widetilde{\alpha W + T}) v_a = \tilde{r}_a$.
  - Solve v_b from $(\widetilde{\alpha W + T}) v_b = (W - \alpha T) v_a + \tilde{r}_b$.
- Computing v from $\widetilde{U}(\alpha) v = r$:
  - Solve v_b from $(\widetilde{\alpha W + T}) v_b = \tilde{r}_b$.
  - Solve v_a from $(\widetilde{\alpha W + T}) v_a = (\alpha T - W) v_b + \tilde{r}_a$.
- Computing v from $\widetilde{P}(\alpha) v = r$:
  - Solve $\tilde{v}_a$ from $(\widetilde{\alpha W + T}) \tilde{v}_a = \tilde{r}_a$.
  - Solve v_b from $(\widetilde{\alpha W + T}) v_b = (W - \alpha T) \tilde{v}_a + \tilde{r}_b$.
  - Solve v_a from $(\widetilde{\alpha W + T}) v_a = (\alpha T - W) v_b + \tilde{r}_a$.

From the foregoing procedures for computing the generalized residual equations, we see that the only difference between our proposed inexact preconditioners and the exact ones in [12] is that we solve the subsystems of the coefficient matrix $\widetilde{\alpha W + T}$ instead of $\alpha W + T$. Because the matrix $\widetilde{\alpha W + T}$ is an approximation to the matrix $\alpha W + T$, we may choose it by making use of the special structure of $\alpha W + T$. Here, we adopt incomplete triangular factorization methods [21] to implement the inexact preconditioning processes. More specifically, we use incomplete LU factorization when $\alpha W + T$ is nonsymmetric and incomplete Cholesky factorization when it is symmetric positive definite.

3 Eigenvalue properties of preconditioned matrices

In this section, we analyze the eigenvalue properties of the preconditioned matrices $\widetilde{L}(\alpha)^{-1}A$, $\widetilde{U}(\alpha)^{-1}A$, and $\widetilde{P}(\alpha)^{-1}A$.

We first introduce some necessary notations that will be used in the subsequent discussions. Denote the matrices

$$M(\alpha) = (\alpha\widetilde{W + T})^{-1}(\alpha W + T), \quad \widetilde{V}(\alpha) = (\alpha\widetilde{W + T})^{-1}(W - \alpha T),$$

and

$$\widetilde{R}(\alpha) = \begin{pmatrix} M(\alpha) & \widetilde{V}(\alpha) \\ \widetilde{V}(\alpha)(M(\alpha) - I) & M(\alpha) + \widetilde{V}(\alpha)^2 \end{pmatrix},$$

$$\widetilde{S}(\alpha) = \begin{pmatrix} M(\alpha) + \widetilde{V}(\alpha)^2 & \widetilde{V}(\alpha)(I - M(\alpha)) \\ -\widetilde{V}(\alpha) & M(\alpha) \end{pmatrix},$$

$$\widetilde{R}_s(\alpha) = \begin{pmatrix} M(\alpha) + \widetilde{V}(\alpha)^2(I - M(\alpha)) & \widetilde{V}(\alpha) - \widetilde{V}(\alpha)^3 - \widetilde{V}(\alpha)M(\alpha) \\ \widetilde{V}(\alpha)(M(\alpha) - I) & M(\alpha) + \widetilde{V}(\alpha)^2 \end{pmatrix}.$$

It then follows from

$$\begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix} \widetilde{R}(\alpha) \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} = \widetilde{S}(\alpha)$$

that $\widetilde{R}(\alpha)$ and $\widetilde{S}(\alpha)$ are orthogonally similar matrices. In addition, we introduce

$$I(\alpha) = \frac{1}{\sqrt{2(\alpha^2+1)}} \begin{pmatrix} (\alpha+1)I & (\alpha-1)I \\ (1-\alpha)I & (\alpha+1)I \end{pmatrix},$$

which is a block two-by-two orthogonal matrix, and the corresponding spectral decomposition is

$$I(\alpha) = \Phi(\alpha)\Lambda(\alpha)\Phi(\alpha)^*,$$

where

$$\Phi(\alpha) = \frac{1}{\sqrt{2}} \begin{pmatrix} I & \mathrm{i}I \\ \mathrm{i}I & I \end{pmatrix} \quad \text{and} \quad \Lambda(\alpha) = \frac{1}{\sqrt{2(\alpha^2+1)}} \begin{pmatrix} ((\alpha+1) + \mathrm{i}(\alpha-1))I & 0 \\ 0 & ((\alpha+1) - \mathrm{i}(\alpha-1))I \end{pmatrix}.$$

Here and in the sequel, $(\cdot)^*$ and $\|\cdot\|$ are used to represent the conjugate transpose and the Euclidean norm of a vector or a matrix, respectively. Let

$$D(\alpha) = \begin{pmatrix} M(\alpha) & 0 \\ 0 & M(\alpha) \end{pmatrix} \tag{3.1}$$

and

$$\widetilde{D}(\alpha) = D(\alpha)\Lambda(\alpha) = \frac{1}{\sqrt{2(\alpha^2+1)}} \begin{pmatrix} ((\alpha+1) + \mathrm{i}(\alpha-1))M(\alpha) & 0 \\ 0 & ((\alpha+1) - \mathrm{i}(\alpha-1))M(\alpha) \end{pmatrix} \tag{3.2}$$

be two block-diagonal matrices.

In what follows, we derive equivalent expressions for the preconditioned matrices $\widetilde{L}(\alpha)^{-1}A$, $\widetilde{U}(\alpha)^{-1}A$, and $\widetilde{P}(\alpha)^{-1}A$ and demonstrate the corresponding spectral properties.

Theorem 3.1 *Let W and T be two real square matrices such that $\widetilde{\alpha W + T}$ is nonsingular, with α being a real constant. Then, for the preconditioning matrices $\widetilde{L}(\alpha)$, $\widetilde{U}(\alpha)$, and $\widetilde{P}(\alpha)$ defined in* (2.1), (2.2), *and* (2.3), *it holds that*

(i) $\widetilde{L}(\alpha)^{-1}A = \widetilde{R}(\alpha)I(\alpha)$, $\widetilde{U}(\alpha)^{-1}A = \widetilde{S}(\alpha)I(\alpha)$, *and* $\widetilde{P}(\alpha)^{-1}A = \widetilde{R}_s(\alpha)I(\alpha)$*;*

(ii) *If* $M(\alpha)$ *is diagonalizable, i.e., there exist a diagonal matrix* $\Omega(\alpha) = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n)$ *and a nonsingular matrix* $Q(\alpha)$ *such that* $M(\alpha) = Q(\alpha)^{-1}\Omega(\alpha)Q(\alpha)$*, then*

(ii$_1$) *The eigenvalues of* $\widetilde{L}(\alpha)^{-1}A$ *and* $\widetilde{U}(\alpha)^{-1}A$ *are located respectively within the same union of circles having its center* $\widetilde{c}_{\pm j}(\alpha)$ *and radius* $\widetilde{\delta}(\alpha)$ *on the complex plane, where*

$$\widetilde{c}_{\pm j}(\alpha) = \frac{1}{\sqrt{2(\alpha^2+1)}}((\alpha+1) \pm \mathrm{i}(\alpha-1))\mu_j$$

and

$$\widetilde{\delta}(\alpha) = \|Q^{-1}(\alpha)\|\|Q(\alpha)\|\|\widetilde{V}(\alpha)\| \left\{\sqrt{1+\|\widetilde{V}(\alpha)\|^2} + \|M(\alpha) - I\|\right\};$$

(ii$_2$) *The eigenvalues of* $\widetilde{P}(\alpha)^{-1}A$ *are located respectively within the union of circles having its center* $\widetilde{c}_{\pm j}(\alpha)$ *and radius* $\widetilde{\delta}_s(\alpha)$ *on the complex plane, where*

$$\widetilde{c}_{\pm j}(\alpha) = \frac{1}{\sqrt{2(\alpha^2+1)}}((\alpha+1) \pm \mathrm{i}(\alpha-1))\mu_j$$

and

$$\widetilde{\delta}_s(\alpha) = \|Q^{-1}(\alpha)\|\|Q(\alpha)\| \left\{\|\widetilde{V}(\alpha)\|^2\sqrt{1+\|\widetilde{V}(\alpha)\|^2} + \|\widetilde{V}(\alpha)\|\|M(\alpha) - I\| \left(\|\widetilde{V}(\alpha)\| + 1\right)\right\}.$$

Proof We first prove (i). We only demonstrate the product expression of the matrix $\widetilde{L}(\alpha)^{-1}A$, and those about the IRBUT and the IRBTP preconditioning matrices $\widetilde{U}(\alpha)$ and $\widetilde{P}(\alpha)$ can be derived analogously to $\widetilde{L}(\alpha)$. To this end, we define the matrix

$$J(\alpha) = \frac{1}{\sqrt{\alpha^2+1}}\begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix}.$$

By straightforward computation, we obtain $J(\alpha)^{-1} = J(\alpha)^{\mathrm{T}}$, $J(\alpha)^{-1}G^{\mathrm{T}} = I(\alpha)$, and

$$AJ(\alpha) = \frac{1}{\sqrt{\alpha^2+1}}\begin{pmatrix} \alpha W + T & W - \alpha T \\ \alpha T - W & \alpha W + T \end{pmatrix}. \tag{3.3}$$

Because the matrices A and G^{T} satisfy $AG^{\mathrm{T}} = G^{\mathrm{T}}A$ and

$$\widetilde{L}(\alpha)^{-1} = \sqrt{\alpha^2+1}\begin{pmatrix} (\alpha \widetilde{W+T})^{-1} & 0 \\ \widetilde{V}(\alpha)(\alpha \widetilde{W+T})^{-1} & (\alpha \widetilde{W+T})^{-1} \end{pmatrix} G^{\mathrm{T}}, \tag{3.4}$$

it follows from (3.3) and (3.4) that

$$\begin{aligned} \widetilde{L}(\alpha)^{-1}A &= \sqrt{\alpha^2+1}\begin{pmatrix} (\alpha \widetilde{W+T})^{-1} & 0 \\ \widetilde{V}(\alpha)(\alpha \widetilde{W+T})^{-1} & (\alpha \widetilde{W+T})^{-1} \end{pmatrix} AG^{\mathrm{T}} \\ &= \sqrt{\alpha^2+1}\begin{pmatrix} (\alpha \widetilde{W+T})^{-1} & 0 \\ \widetilde{V}(\alpha)(\alpha \widetilde{W+T})^{-1} & (\alpha \widetilde{W+T})^{-1} \end{pmatrix} (AJ(\alpha))(J(\alpha)^{-1}G^{\mathrm{T}}) \\ &= \begin{pmatrix} (\alpha \widetilde{W+T})^{-1} & 0 \\ \widetilde{V}(\alpha)(\alpha \widetilde{W+T})^{-1} & (\alpha \widetilde{W+T})^{-1} \end{pmatrix} \begin{pmatrix} \alpha W + T & W - \alpha T \\ \alpha T - W & \alpha W + T \end{pmatrix} I(\alpha) \end{aligned}$$

$$
\begin{aligned}
&= \begin{pmatrix} M(\alpha) & \widetilde{V}(\alpha) \\ \widetilde{V}(\alpha)(M(\alpha)-I) & M(\alpha)+\widetilde{V}(\alpha)^2 \end{pmatrix} I(\alpha) \\
&= \widetilde{R}(\alpha)I(\alpha).
\end{aligned}
$$

This shows the validity of (i).

Now we demonstrate (ii). We only prove the eigenvalue properties of the IRBLT preconditioned matrix $\widetilde{L}(\alpha)^{-1}A$ in (ii$_1$) because those of the IRBUT preconditioned matrix $\widetilde{U}(\alpha)^{-1}A$ can be obtained similarly. From (3.1) we have

$$
\widetilde{R}(\alpha) = D(\alpha) + \widetilde{Y}(\alpha),
$$

with

$$
\widetilde{Y}(\alpha) = \begin{pmatrix} 0 & \widetilde{V}(\alpha) \\ \widetilde{V}(\alpha)(M(\alpha)-I) & \widetilde{V}(\alpha)^2 \end{pmatrix} = \begin{pmatrix} \widetilde{V}(\alpha) & 0 \\ 0 & \widetilde{V}(\alpha) \end{pmatrix} \begin{pmatrix} 0 & I \\ M(\alpha)-I & \widetilde{V}(\alpha) \end{pmatrix}.
$$

Thus, it can be obtained from (i) that

$$
\widetilde{L}(\alpha)^{-1}A = \widetilde{R}(\alpha)I(\alpha) = D(\alpha)I(\alpha) + \widetilde{Y}(\alpha)I(\alpha)
$$

and

$$
\begin{aligned}
\Phi(\alpha)^*\widetilde{L}(\alpha)^{-1}A\Phi(\alpha) &= \Phi(\alpha)^*D(\alpha)I(\alpha)\Phi(\alpha) + \Phi(\alpha)^*\widetilde{Y}(\alpha)I(\alpha)\Phi(\alpha) \\
&= \Phi(\alpha)^*D(\alpha)\Phi(\alpha)\Phi(\alpha)^*I(\alpha)\Phi(\alpha) + \Phi(\alpha)^*\widetilde{Y}(\alpha)I(\alpha)\Phi(\alpha) \\
&= D(\alpha)\Lambda(\alpha) + \Phi(\alpha)^*\widetilde{Y}(\alpha)I(\alpha)\Phi(\alpha) \\
&= \widetilde{D}(\alpha) + \Phi(\alpha)^*\widetilde{Y}(\alpha)I(\alpha)\Phi(\alpha).
\end{aligned}
$$

Here we have used the fact that $\Phi(\alpha)$ is unitary and $\Phi(\alpha)^*D(\alpha)\Phi(\alpha) = D(\alpha)$. Because $I(\alpha)$ is orthogonal, we can further obtain

$$
\begin{aligned}
\|\Phi(\alpha)^*\widetilde{Y}(\alpha)I(\alpha)\Phi(\alpha)\| &= \|\widetilde{Y}(\alpha)\| \\
&\leq \|\widetilde{V}(\alpha)\| \left\| \begin{pmatrix} 0 & I \\ M(\alpha)-I & \widetilde{V}(\alpha) \end{pmatrix} \right\| \\
&\leq \|\widetilde{V}(\alpha)\| \left\{ \left\| \begin{pmatrix} 0 & I \\ 0 & \widetilde{V}(\alpha) \end{pmatrix} \right\| + \left\| \begin{pmatrix} 0 & 0 \\ M(\alpha)-I & 0 \end{pmatrix} \right\| \right\} \\
&\leq \|\widetilde{V}(\alpha)\| \left\{ \sqrt{1+\|\widetilde{V}(\alpha)\|^2} + \|M(\alpha)-I\| \right\}.
\end{aligned}
$$

Therefore,

$$
\|\Phi(\alpha)^*\widetilde{L}(\alpha)^{-1}A\Phi(\alpha) - \widetilde{D}(\alpha)\| \leq \|\widetilde{V}(\alpha)\| \left\{ \sqrt{1+\|\widetilde{V}(\alpha)\|^2} + \|M(\alpha)-I\| \right\}.
$$

Let λ be an eigenvalue of the matrix $\widetilde{L}(\alpha)^{-1}A$. Because $\Phi(\alpha)$ is a unitary matrix, λ is also an eigenvalue of the matrix $\Phi(\alpha)^*\widetilde{L}(\alpha)^{-1}A\Phi(\alpha)$. If $M(\alpha)$ is diagonalizable, i.e., $M(\alpha) = Q(\alpha)^{-1}\Omega(\alpha)Q(\alpha)$, then from (3.2) we obtain

$$
\begin{aligned}
\widetilde{D}(\alpha) = &\frac{1}{\sqrt{2(\alpha^2+1)}} \begin{pmatrix} Q(\alpha)^{-1} & 0 \\ 0 & Q(\alpha)^{-1} \end{pmatrix} \\
&\times \begin{pmatrix} ((\alpha+1)+\mathrm{i}(\alpha-1))\Omega(\alpha) & 0 \\ 0 & ((\alpha+1)-\mathrm{i}(\alpha-1))\Omega(\alpha) \end{pmatrix} \begin{pmatrix} Q(\alpha) & 0 \\ 0 & Q(\alpha) \end{pmatrix},
\end{aligned}
$$

where $\Omega(\alpha) = \mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n)$. Denoting $\widetilde{c}_{\pm j}(\alpha) = ((\alpha + 1) \pm \mathrm{i}(\alpha - 1))\mu_j/\sqrt{2(\alpha^2 + 1)}$, we know that $\widetilde{c}_{\pm j}(\alpha)$ are eigenvalues of the matrix $\widetilde{D}(\alpha)$. From the Bauer–Fike theorem [29,30] we have

$$\begin{aligned}
|\lambda - \widetilde{c}_{\pm j}(\alpha)| &\leq \|Q(\alpha)^{-1}\|\|Q(\alpha)\|\|\Phi(\alpha)^*\widetilde{L}(\alpha)^{-1}A\Phi(\alpha) - \widetilde{D}(\alpha)\| \\
&\leq \|Q(\alpha)^{-1}\|\|Q(\alpha)\|\|\widetilde{V}(\alpha)\| \left\{\sqrt{1 + \|\widetilde{V}(\alpha)\|^2} + \|M(\alpha) - I\|\right\} = \widetilde{\delta}(\alpha).
\end{aligned}$$

This verifies the validity of (ii$_1$).

Finally, we prove (ii$_2$). By rewriting $\widetilde{R}_s(\alpha) = D(\alpha) + Y_s(\alpha)$ with

$$Y_s(\alpha) = \begin{pmatrix} \widetilde{V}(\alpha)^2(I - M(\alpha)) & \widetilde{V}(\alpha) - \widetilde{V}(\alpha)^3 - \widetilde{V}(\alpha)M(\alpha) \\ \widetilde{V}(\alpha)(M(\alpha) - I) & \widetilde{V}(\alpha)^2 \end{pmatrix},$$

from (i) we obtain

$$\widetilde{P}(\alpha)^{-1}A = \widetilde{R}_s(\alpha)I(\alpha) = D(\alpha)I(\alpha) + \widetilde{Y}_s(\alpha)I(\alpha)$$

and, thereby,

$$\begin{aligned}
\Phi(\alpha)^*\widetilde{P}(\alpha)^{-1}A\Phi(\alpha) &= \Phi(\alpha)^*D(\alpha)I(\alpha)\Phi(\alpha) + \Phi(\alpha)^*\widetilde{Y}_s(\alpha)I(\alpha)\Phi(\alpha) \\
&= D(\alpha)\Lambda(\alpha) + \Phi(\alpha)^*\widetilde{Y}_s(\alpha)I(\alpha)\Phi(\alpha) \\
&= \widetilde{D}(\alpha) + \Phi(\alpha)^*\widetilde{Y}_s(\alpha)I(\alpha)\Phi(\alpha). \qquad (3.5)
\end{aligned}$$

Moreover, it holds that

$$\begin{aligned}
\|\Phi(\alpha)^*\widetilde{Y}_s(\alpha)I(\alpha)\Phi(\alpha)\| &= \|\widetilde{Y}_s(\alpha)\| \\
&\leq \left\|\begin{pmatrix} 0 & -\widetilde{V}(\alpha)^3 \\ 0 & \widetilde{V}(\alpha)^2 \end{pmatrix}\right\| + \left\|\begin{pmatrix} \widetilde{V}(\alpha)^2(I - M(\alpha)) & \widetilde{V}(\alpha)(I - M(\alpha)) \\ \widetilde{V}(\alpha)(M(\alpha) - I) & \widetilde{V}(\alpha) \end{pmatrix}\right\| \\
&\leq \|\widetilde{V}(\alpha)\|^2\sqrt{1 + \|\widetilde{V}(\alpha)\|^2} + \left\|\begin{pmatrix} \widetilde{V}(\alpha)(I - M(\alpha)) & 0 \\ 0 & \widetilde{V}(\alpha)(I - M(\alpha)) \end{pmatrix}\begin{pmatrix} \widetilde{V}(\alpha) & I \\ -I & 0 \end{pmatrix}\right\| \\
&\leq \|\widetilde{V}(\alpha)\|^2\sqrt{1 + \|\widetilde{V}(\alpha)\|^2} + \|\widetilde{V}(\alpha)\|\|M(\alpha) - I\| \left\|\begin{pmatrix} \widetilde{V}(\alpha) & I \\ -I & 0 \end{pmatrix}\right\| \\
&\leq \|\widetilde{V}(\alpha)\|^2\sqrt{1 + \|\widetilde{V}(\alpha)\|^2} + \|\widetilde{V}(\alpha)\|\|M(\alpha) - I\| \left\{\left\|\begin{pmatrix} \widetilde{V}(\alpha) & 0 \\ 0 & 0 \end{pmatrix}\right\| + \left\|\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}\right\|\right\} \\
&\leq \|\widetilde{V}(\alpha)\|^2\sqrt{1 + \|\widetilde{V}(\alpha)\|^2} + \|\widetilde{V}(\alpha)\|\|M(\alpha) - I\| \left(\|\widetilde{V}(\alpha)\| + 1\right). \qquad (3.6)
\end{aligned}$$

Therefore, from (3.5) and (3.6) we have

$$\|\Phi(\alpha)^*\widetilde{P}(\alpha)^{-1}A\Phi(\alpha) - \widetilde{D}(\alpha)\| \leq \|\widetilde{V}(\alpha)\|^2\sqrt{1 + \|\widetilde{V}(\alpha)\|^2} + \|\widetilde{V}(\alpha)\|\|M(\alpha) - I\| \left(\|\widetilde{V}(\alpha)\| + 1\right) = \widetilde{\delta}_s(\alpha). \quad (3.7)$$

Analogously to the derivation of (ii$_1$), from (3.7) we immediately obtain the validity of (ii$_2$). □

For the special case $\widetilde{\alpha W + T} = \alpha W + T$, i.e., $M(\alpha) = I$, the inexact rotated block triangular preconditioners change into the exact ones, and the results given in Theorem 3.1 are the same as those in [12]. We may guarantee that all the eigenvalues of these inexact preconditioned matrices are located in the right half or left half of the complex plane by selecting a suitable parameter α and approximate matrix $\widetilde{\alpha W + T}$, though this is not an easy task. In applications, we choose $\widetilde{\alpha W + T}$ in accordance with the special structures of the matrices W and T.

4 Numerical results

In this section, we test the effectiveness of our proposed inexact rotated block triangular preconditioners. To this end, we apply the GMRES iteration method, incorporated with the IRBLT, the IRBUT, and the IRBTP preconditioners, to the system of linear equations (1.1).

The real block two-by-two linear system (1.1) is equivalently reformulated from the complex linear system

$$(W + \mathrm{i}T)\mathbf{x} = \mathbf{h}, \quad \text{with } \mathbf{x} = y + \mathrm{i}z \quad \text{and} \quad \mathbf{h} = f + \mathrm{i}g. \tag{4.1}$$

In the following two examples [12,13], we need to solve the linear system (4.1).

Example 4.1 (Complex-Shifted Linear System) Consider the parabolic partial differential equation in [12,13]. By discretizing it with a finite-element method of bilinear elements on a uniform rectangular mesh, we obtain a linear system of the form

$$(K + \mathrm{i}\omega\tau M)\mathbf{x} = \mathbf{h},$$

where ω is a positive parameter, τ is the temporal discretization step size, and $K = M + \tau L \in \mathbb{R}^{m\times m}$, with $M \in \mathbb{R}^{m\times m}$ being the mass matrix and $L \in \mathbb{R}^{m\times m}$ the discrete negative Laplace operator. Here, the exact solution of the parabolic partial differential equation and the right-hand vector $\mathbf{h}$ of the discretized linear system are given as in [12]; see also [11,13].

In Example 4.1, let $W = K$ and $T = \omega\tau M$. Then we obtain the linear system (4.1). It is easy to verify that the matrices W and T are symmetric positive definite, as is $\alpha W + T$.

Example 4.2 (Nonsymmetric BBC Problem) A complex linear system is of the form $(W + \mathrm{i}T)\mathbf{z} = \mathbf{h}$, with

$$W = I \otimes L + L \otimes I \quad \text{and} \quad T = 10(I \otimes L_c + L_c \otimes I) + 9(e_1 e_l^{\mathrm{T}} + e_l e_1^{\mathrm{T}}) \otimes I,$$

where $L = \mathrm{tridiag}(-1-\theta, 2, -1+\theta) \in \mathbb{R}^{l\times l}$ is a tridiagonal matrix, $\theta = \nu/[2(l+1)]$ with ν a positive constant, $L_c = L - e_1 e_l^{\mathrm{T}} - e_l e_1^{\mathrm{T}}$, and e_1 and e_l are the first and last column vectors of the identity matrix $I \in \mathbb{R}^{l\times l}$, respectively. The right-hand-side vector $\mathbf{h}$ is defined as in [12]. See also [13,18,19].

In our implementations, the initial guess is taken to be zero and the iteration process is terminated once the current residual $r^{(j)}$ satisfies

$$\frac{\|r^{(j)}\|_2}{\|r^{(0)}\|_2} \le 10^{-6}.$$

The parameter α is taken to be 1.0 because it is observed that in our implementations the results do not rely on α sensitively. From [12] we see that it also holds true in the exact preconditioned GMRES methods. In addition, we implement the inexact preconditioning processes by incomplete Cholesky factorization in Example 4.1 and by incomplete LU factorization in Example 4.2. All codes were written in MATLAB R2008a and all experiments were performed on a personal computer with 1.86 G memory.

In Tables 1 and 2, we list the numbers of iteration steps (IT) and CPU times in seconds (CPU) of the IRBLT, IRBUT, and IRBTP preconditioned GMRES methods, termed briefly as IRBLT-GMRES, IRBUT-GMRES and IRBTP-GMRES, for Examples 4.1 and 4.2 with respect to different values of the problem parameter ω or ν and the problem size m, respectively. Here, the CPU times are shown in parentheses. In Tables 3 and 4, the speed-up ratios of the CPU times between the exact preconditioned GMRES method in [12] and the inexact ones in this paper, i.e., $\mathrm{CPU}_{\mathrm{exact}}/\mathrm{CPU}_{\mathrm{inexact}}$, are listed for Examples 4.1 and 4.2 with respect to various values of the problem parameter ω or ν and the problem size m, respectively.

From Tables 1 and 2 we see that all of these three inexact preconditioned GMRES methods can successfully compute satisfactory approximations to the exact solutions of Examples 4.1 and 4.2 in a few iteration steps. Moreover, the iteration steps remain almost invariant for each method and for any fixed problem parameter ω or ν when the

Table 1 Iteration step and CPU time [IT (CPU)] for Example 4.1

Method	ω	$m \times m$				
		625 × 625	900 × 900	1,764 × 1,764	1,936 × 1,936	2,025 × 2,025
IRBLT-GMRES	1	5 (0.22)	5 (0.43)	5 (1.57)	5 (1.85)	5 (2.01)
	10^2	9 (0.28)	9 (0.55)	10 (2.05)	10 (2.44)	10 (2.67)
	10^3	8 (0.25)	8 (0.50)	9 (1.98)	9 (2.32)	9 (2.56)
	10^4	8 (0.26)	8 (0.52)	8 (1.87)	8 (2.16)	8 (2.41)
IRBUT-GMRES	1	4 (0.18)	4 (0.36)	4 (1.32)	4 (1.57)	4 (1.72)
	10^2	9 (0.23)	9 (0.48)	9 (1.71)	9 (1.99)	9 (2.21)
	10^3	8 (0.22)	8 (0.43)	8 (1.62)	8 (1.91)	8 (2.05)
	10^4	7 (0.22)	8 (0.43)	8 (1.56)	8 (1.89)	8 (2.05)
IRBTP-GMRES	1	4 (0.20)	4 (0.41)	4 (1.46)	4 (1.74)	4 (1.89)
	10^2	9 (0.27)	9 (0.56)	9 (1.97)	9 (2.37)	9 (2.55)
	10^3	8 (0.25)	8 (0.50)	8 (1.83)	8 (2.19)	8 (2.42)
	10^4	7 (0.25)	8 (0.52)	8 (1.87)	8 (2.18)	8 (2.38)

Table 2 Iteration step and CPU time [IT (CPU)] for Example 4.2

Method	ν	$m \times m$				
		625 × 625	900 × 900	1,225 × 1,225	1,600 × 1,600	2,025 × 2,025
IRBLT-GMRES	0.1	6 (0.26)	6 (0.50)	7 (0.96)	7 (1.57)	6 (2.29)
	1	7 (0.27)	7 (0.53)	7 (0.97)	7 (1.61)	7 (2.46)
	10	7 (0.28)	7 (0.55)	8 (0.99)	8 (1.70)	8 (2.56)
	100	5 (0.23)	5 (0.48)	5 (0.78)	5 (1.37)	5 (2.12)
IRBUT-GMRES	0.1	6 (0.23)	6 (0.46)	7 (0.86)	7 (1.41)	7 (2.16)
	1	7 (0.24)	7 (0.47)	7 (0.83)	7 (1.37)	7 (2.15)
	10	7 (0.25)	7 (0.47)	8 (0.89)	8 (1.46)	8 (2.29)
	100	5 (0.20)	5 (0.43)	5 (0.75)	5 (1.23)	5 (1.92)
IRBTP-GMRES	0.1	5 (0.23)	6 (0.51)	6 (0.92)	6 (1.50)	6 (2.29)
	1	6 (0.25)	6 (0.52)	7 (0.94)	7 (1.59)	6 (2.29)
	10	7 (0.29)	7 (0.56)	7 (0.95)	7 (1.57)	7 (2.45)
	100	5 (0.23)	5 (0.46)	5 (0.83)	5 (1.35)	5 (2.16)

problem size m increases. It can be noticed from Table 1 that of the three methods, the IRBLT-GMRES method needs the most iteration steps and CPU time. The IRBUT-GMRES and the IRBTP-GMRES methods require almost the same number of iteration steps, but the IRBTP-GMRES method costs more CPU time because it requires solving three subsystems of the coefficient matrix $\alpha \widetilde{W + T}$ and the same two subsystems for the IRBUT-GMRES method in each preconditioning process. We see also from Table 2 that the IRBUT-GMRES method requires the least CPU time, and the IRBTP-GMRES method sometimes requires the fewest number of iteration steps in Example 4.2.

Comparing the results in Tables 1 and 2 with those in [12] for Examples 4.1 and 4.2, we observe that for the same parameters α, ω or ν, and m, the iteration steps of the IRBLT-GMRES, IRBUT-GMRES, and IRBTP-GMRES methods remains almost the same as those of the RBLT-GMRES, RBUT-GMRES, and RBTP-GMRES methods. The CPU times of these inexact preconditioned GMRES methods are much less than the corresponding exact ones because the inexact preconditioning processes cost less CPU time than the exact ones. More specifically, in

Example 4.1 we perform incomplete Cholesky factorization with threshold and pivoting (ICTP) [21] by making use of the sparse structure of the matrix $\alpha W + T$, denoted by $\alpha W + T \approx \widetilde{L}\widetilde{L}^T$, and let $\widetilde{\alpha W + T} = \widetilde{L}\widetilde{L}^T$. Because $\widetilde{L}\widetilde{L}^T$ is a good approximation of $\alpha W + T$, and $\widetilde{L}$ is a very sparse matrix, the cost to invert $\widetilde{\alpha W + T}$ is very cheap in the inexact preconditioning processes. In Example 4.2 we use incomplete LU factorization with threshold and pivoting (ILUTP) for the matrix $\alpha W + T$ for the same reasons. Therefore, we say that the IRBLT, IRBUT, and IRBTP preconditioners are superior to the exact ones in [12] for accelerating the convergence rate of the GMRES method in Examples 4.1 and 4.2.

From Tables 3 and 4 we see that the speed-up ratios of $\mathrm{CPU}_{\mathrm{exact}}/\mathrm{CPU}_{\mathrm{inexact}}$ are greater than 1 for any fixed problem parameter ω or ν and problem size m, which indicates that the IRBLT-GMRES, IRBUT-GMRES, and IRBTP-GMRES methods cost less CPU time than the exact ones, respectively. This shows the advantages of these

Table 3 Speed-up ($\mathrm{CPU}_{\mathrm{exact}}/\mathrm{CPU}_{\mathrm{inexact}}$) for Example 4.1

Method	ω	$m \times m$				
		625 × 625	900 × 900	1,764 × 1,764	1,936 × 1,936	2,025 × 2,025
IRBLT-GMRES	1	1.22	1.22	1.31	1.32	1.48
	10^2	1.41	1.32	1.44	1.44	1.61
	10^3	1.45	1.36	1.42	1.44	1.63
	10^4	1.40	1.30	1.42	1.48	1.60
IRBUT-GMRES	1	1.24	1.23	1.35	1.36	1.49
	10^2	1.63	1.37	1.49	1.55	1.70
	10^3	1.51	1.42	1.50	1.53	1.76
	10^4	1.42	1.42	1.54	1.63	1.88
IRBTP-GMRES	1	1.42	1.36	1.45	1.51	1.72
	10^2	1.75	1.53	1.65	1.66	1.95
	10^3	1.91	1.61	1.70	1.68	1.94
	10^4	1.62	1.57	1.66	1.68	1.97

Table 4 Speed-up ($\mathrm{CPU}_{\mathrm{exact}}/\mathrm{CPU}_{\mathrm{inexact}}$) for Example 4.2

Method	ν	$m \times m$				
		625 × 625	900 × 900	1,225 × 1,225	1,600 × 1,600	2,025 × 2,025
IRBLT-GMRES	0.1	1.01	1.07	1.12	1.19	1.27
	1	1.00	1.08	1.10	1.17	1.27
	10	1.01	1.10	1.15	1.17	1.29
	100	1.01	1.04	1.18	1.20	1.31
IRBUT-GMRES	0.1	1.09	1.04	1.16	1.21	1.31
	1	1.02	1.12	1.19	1.24	1.33
	10	1.03	1.10	1.17	1.23	1.29
	100	1.08	1.07	1.15	1.23	1.35
IRBTP-GMRES	0.1	1.10	1.10	1.18	1.25	1.35
	1	1.09	1.09	1.21	1.24	1.36
	10	1.04	1.08	1.20	1.26	1.35
	100	1.09	1.14	1.21	1.31	1.35

inexact preconditioners over the exact preconditioners in [12]. And the larger the speed-up ratio is, the more apparent the superiority of the inexact preconditioners over the exact ones. Further, the speed-up ratios also increase when the parameter m grows (i.e., the size of the matrices becomes large) in both Examples 4.1 and 4.2. In particular, when $m = 2,025$, the speed-up ratios are the largest for all cases listed in Tables 3 and 4. Hence, we may conclude that when matrices grow in size, their structures become more sparse and the inexact preconditioners are more suitable for solving the systems of linear equations in Examples 4.1 and 4.2. Moreover, the speed-up ratios of the IRBTP-GMRES method is generally greater than that of the IRBLT-GMRES and IRBUT-GMRES methods. Thus, it is obvious that the inexact preconditioners proposed in this paper are more effective than the exact ones in [12] based on the speed-up ratios of the CPU times.

5 Concluding remarks

IRBLT, IRBUT, and IRBTP preconditioning matrices were proposed to precondition linear systems, when the coefficient matrix is a block two-by-two matrix of real square blocks based on the RBLT, RBUT, and RBTP preconditioners in [12]. We analyzed the eigenvalue properties of these three inexact preconditioned matrices and showed the effectiveness of the IRBLT, IRBUT, and IRBTP preconditioners at accelerating the convergence rates of Krylov subspace iteration methods such as GMRES for solving the linear system (1.1) on the basis of both theoretical analysis and numerical results.

In Sect. 2 we proved that the eigenvalues of the IRBLT, IRBUT, and IRBTP preconditioned matrices are respectively located within the union of some circles. It is difficult to provide details about the eigenvalues and eigenvectors of these preconditioned matrices like the exact ones in [12] in the general case. In practical applications, we may choose the approximate matrix $\alpha \widetilde{W + T}$ by splitting the matrix $\alpha W + T$. More specifically, let $\alpha W + T = B(\alpha) - C(\alpha)$ be a splitting of $\alpha W + T$. Then we can take $\alpha \widetilde{W + T} = B(\alpha)$ or

$$\alpha \widetilde{W + T} = (\alpha W + T)\left[I - \left(B(\alpha)^{-1}C(\alpha)\right)^m\right]^{-1} = B(\alpha)\left[\sum_{j=0}^{m-1}\left(B(\alpha)^{-1}C(\alpha)\right)^j\right]^{-1}, \tag{5.1}$$

with m being a positive integer; see [31]. It is expected that in future work, when the approximation matrix $\alpha \widetilde{W + T}$ is specially defined as in (5.1), more eigenproperties of the preconditioned matrices will be derived.

Acknowledgments This work was supported by the National Natural Science Foundation (11301521), P. R. China.

References

1. Hiptmair R (2002) Finite elements in computational electromagnetism. Acta Numer 11:237–339
2. van Rienen U (2001) Numerical methods in computational electrodynamics: linear systems in practical applications. Springer, Berlin
3. Gill PE, Murray W, Wright MH (1984) Practical optimization. Academic Press, New York
4. Gould NIM, Hribar ME, Nocedal J (2001) On the solution of equality constrained quadratic programming problems arising in optimization. SIAM J Sci Comput 23:1375–1394
5. Keller C, Gould NIM, Wathen AJ (2000) Constrained preconditioning for indefinite linear systems. SIAM J Matrix Anal Appl 21:1300–1317
6. Rees T, Dollar HS, Wathen AJ (2010) Optimal solvers for PDE-constraint optimization. SIAM J Sci Comput 32:271–298
7. Rees T, Stoll M (2010) Block-triangular preconditioners for PDE-constrained optimization. Numer Linear Algebra Appl 17:977–996
8. Bai Z-Z (2011) Block preconditioners for elliptic PDE-constrained optimization problems. Computing 91:379–395
9. Bai Z-Z, Benzi M, Chen F, Wang Z-Q (2013) 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
10. Lions JL (1968) Optimal control of systems governed by partial differential equations. Spring, Berlin
11. Axelsson O, Kucherov A (2000) Real valued iterative methods for solving complex symmetric linear systems. Numer Linear Algebra Appl 7:197–218

12. Bai Z-Z (2013) Rotated block triangular preconditioning based on PMHSS. Sci China Math doi:10.1007/s11425-013-4695-9
13. Bai Z-Z (2013) On preconditioned iteration methods for complex linear systems. J Eng Math doi:10.1007/s10665-013-9670-5
14. Bai Z-Z (2006) Structured preconditioners for nonsingular matrices of block two-by-two structures. Math Comput 75:791–815
15. Benzi M, Bertaccini D (2008) Block preconditioning of real-valued iterative algorithms for complex linear systems. IMA J Numer Anal 28:598–618
16. Guo X-X, Wang S (2012) Modified HSS iteration methods for a class of non-Hermitian positive-definite linear systems. Appl Math Comput 218:10122–10128
17. Xu W-W (2013) A generalization of preconditioned MHSS iteration method for complex symmetric indefinite linear systems. Appl Math Comput 219:10510–10517
18. Bai Z-Z, Benzi M, Chen F (2010) Modified HSS iteration methods for a class of complex symmetric linear systems. Computing 87:93–111
19. Bai Z-Z, Benzi M, Chen F (2011) On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer Algorithms 56:297–317
20. Li X, Yang A-L, Wu Y-J (2013) Lopsided PMHSS iteration method for a class of complex symmetric linear systems. Numer Algorithms. doi:10.1007/s11075-013-9748-1
21. Saad Y (1996) Iterative methods for sparse linear systems. PWS Publishing Company, Boston
22. Axelsson O, Neytcheva MG, Ahmad B (2013) A comparison of iterative methods to solve complex valued linear algebraic systems. TR 2013–005, Institute for Information Technology, Uppsala University.
23. Benzi M, Ferragut L, Pennacchio M, Simoncini V (2010) Solution of linear systems from an optimal control problem arising in wind simulation. Numer Linear Algebra Appl 17:895–915
24. Yang A-L, Wu Y-J, Xu Z-J (2013) The semi-convergence properties of MHSS method for a class of complex nonsymmetric singular linear systems. Numer Algorithms. doi:10.1007/s11075-013-9755-2
25. Bai Z-Z, Golub GH, Ng MK (2003) Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J Matrix Anal Appl 24:603–626
26. Bai Z-Z, Golub GH, Pan J-Y (2004) Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer Math 98:1–32
27. Bai Z-Z, Golub GH, Ng MK (2008) On ineaxct Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. Linear Algebra Appl 428:413–440
28. Bai Z-Z, Li G-Q (2003) Restrictively preconditioned conjugate gradient methods for systems of linear equations. IMA J Numer Anal 23:561–580
29. Golub GH, Van Loan CF (2013) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore
30. Parlett BN (1998) The symmetric eigenvalue problem. SIAM, Philadelphia
31. Bai Z-Z, Wang Z-Q (2006) Restrictive preconditioners for conjugate gradient methods for symmetric positive definite linear systems. J Comput Appl Math 187:202–226