Journal of Applied Analysis and Computation
Volume 16, Number 3, June 2026, 1535–1551

Website:http://www.jaac-online.com
DOI:10.11948/20230287

ON COMPLEX ROTATED BLOCK TRIANGULAR PRECONDITIONED ITERATION METHODS FOR A CLASS OF BLOCK TWO-BY-TWO COMPLEX LINEAR SYSTEMS

Jian-Mei Ni[1] and Hai-Long Shen[1,†]

Abstract For the optimal control problems bound by the time-periodic eddy current equation, we build a class of complex rotating block triangular preconditioners based on the BAS preconditioning matrix. The corresponding preconditioned matrices' spectrum characteristics are examined. The application of these complicated rotated block triangular preconditioners to quicken Krylov subspace iteration techniques demonstrates their potential to be competitive with and even more effective than the BAS preconditioner.

Keywords PDE-constrained optimization, block two-by-two linear system, finite element, rotated block triangular preconditioner.

MSC(2010) 65F08, 65F10.

1. Introduction

Consider the following distributed control problem [8,9]: Find the state $y\left(x,t\right)$ and the control $u\left(x,t\right)$ which minimize the functional

$$J\left(y,u\right)=\frac{\beta}{2}\int_0^T\int_{\Omega}\left|u\left(x,t\right)\right|^2\mathrm{d}x\mathrm{d}t+\frac{1}{2}\int_0^T\int_{\Omega}\left|y\left(x,t\right)-y_d\left(x,t\right)\right|^2\mathrm{d}x\mathrm{d}t, \tag{1.1}$$

subject to the time-periodic problem [12,13]

$$\begin{cases}\sigma\dfrac{\partial}{\partial t}y\left(x,t\right)+\operatorname{curl}\left(v\operatorname{curl}\left(y\left(x,t\right)\right)\right)+\varepsilon y\left(x,t\right)=u\left(x,t\right), & \text{in } Q_T,\\ y\left(x,t\right)\times n=0, \quad \text{on } \Sigma_T, & \\ u\left(x,0\right)=u\left(x,T\right), \quad \text{in } \Omega, & \\ y\left(x,0\right)=y\left(x,T\right), \quad \text{in } \Omega. & \end{cases} \tag{1.2}$$

Ω denotes an open and bounded domain in $\mathbb{R}^d$ for $d\in\{2,3\}$ and its boundary Γ is Lipschitz-continuous. The lateral surface of the space-time cylinder $Q_T=\Omega\times(0,T)$ is $\Sigma_T=\partial\Omega\times(0,T)$. Meanwhile, $v\in L^{\infty}\left(\Omega\right)$, $\sigma\in L^{\infty}\left(\Omega\right)$ and β denote the magnetic reluctivity, electrical conductivity and additional regularization parameter, respectively. It frequently results from PDE constrained optimization problems that have been discretized using finite elements. We direct readers to [8, 11, 14] for more background information on this class of linear system of

†The corresponding author.
[1]Department of Mathematics, College of Sciences, Northeastern University, Shenyang, Liaoning 110819, China
Email: 17862895953@163.com(J.-M. Ni), hailong_shen@126.com(H.-L. Shen)

equations. Furthermore, it is presumed in numerous practical situations that $y_d(x,t)$ is time-harmonic [7], i.e.,

$$y_d(x,t) = \widehat{y}_d(x)\, e^{i\widetilde{\omega}t}$$

with $\widetilde{\omega} = \frac{2\pi k}{T}$ for some $k \in \mathbb{Z}$. So the solution $y(x,t)$ and the control functions $u(x,t)$ also have this property. They satisfy the following equations

$$y(x,t) = \widehat{y}(x)\, e^{i\widetilde{\omega}t}, \quad u(x,t) = \widehat{u}(x)\, e^{i\widetilde{\omega}t}.$$

By replacing the above equations, we can obtain problem as follows:

$$\begin{aligned} &\min_{y,u} \frac{\beta}{2} \int_{\Omega} |\widehat{u}(x)|^2 \mathrm{d}x + \frac{1}{2}\int_{\Omega} |\widehat{y}(x) - \widehat{y}_d(x)|^2 \mathrm{d}x, \\ &s.t. \begin{cases} i\widetilde{\omega}\sigma\widehat{y}(x) + \mathrm{curl}\,(\mathrm{vcurl}\,(\widehat{y}(x))) + \varepsilon\widehat{y}(x) = \widehat{u}(x), & \text{in } \Omega, \\ \widehat{y}(x) \times n = 0, \quad \text{on } \Gamma. \end{cases} \end{aligned} \tag{1.3}$$

We use the discretize-then-optimize framework [2,15,21] to discretize the constraints and obtain a finite dimensional solution. A simplified complex linear system is obtained as follows:

$$\begin{aligned} &\min_{y,u} \frac{\beta}{2} u^* M u + \frac{1}{2}(y - y_{\mathrm{d}})^* M (y - y_{\mathrm{d}}), \\ &s.t. \quad i\widetilde{\omega}\sigma M y + K y - M u = 0, \end{aligned} \tag{1.4}$$

where the real matrix $M = [M_{ij}]_{n\times n}$ is the mass matrix and the real matrix $K = [K_{ij}]_{n\times n}$ is the stiffness matrix. Also, the vectors y, y_d, u represent the coefficient vectors of the corresponding finite element functions $\widehat{y}, \widehat{y_d}, \widehat{u}$, respectively. $(\cdot)^*$ denotes conjugate transpose. The Lagrangian functional is presented as

$$L(y,u,p) = \frac{1}{2}(y - y_{\mathrm{d}})^* M (y - y_{\mathrm{d}}) + \frac{\beta}{2} u^* M u + p^* (i\sigma\widetilde{\omega} M y + K y - M u) \tag{1.5}$$

where p denotes the Lagrangian multiplier associated with the constraint. Applying the stationary condition, i.e., $\nabla L(y,u,p) = 0$, leads to the following Karush-Kuhn-Tucker (KKT) system

$$\begin{bmatrix} M & 0 & K - i\omega M \\ 0 & \beta M & -M \\ K + i\omega M & -M & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ p \end{bmatrix} = \begin{bmatrix} M y_d \\ 0 \\ 0 \end{bmatrix}, \tag{1.6}$$

where $\omega = \sigma\widetilde{\omega}$. The second equation yields $u = \frac{1}{\beta}p$. Using it to eliminate p to reduce (6) to

$$\begin{bmatrix} M & \beta(K - i\omega M) \\ (K + i\omega M) & -M \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} M y_d \\ 0 \end{bmatrix}. \tag{1.7}$$

Then using a simple scaling technique $\widehat{u} = -\sqrt{\beta}u$ or $\overline{u} = \sqrt{\beta}u$, simplify (7) to following complex systems:

$$\mathscr{A}x = \begin{bmatrix} M & -\sqrt{\beta}(K - i\omega M) \\ \sqrt{\beta}(K + i\omega M) & M \end{bmatrix} \begin{bmatrix} y \\ \widehat{u} \end{bmatrix} = \begin{bmatrix} M y_d \\ 0 \end{bmatrix} = b \tag{1.8}$$

or

$$\mathscr{A}_{\mathrm{I}} x_{\mathrm{I}}=\begin{bmatrix} M & \sqrt{\beta}\,(K-i\omega M) \\ \sqrt{\beta}\,(K+i\omega M) & -M \end{bmatrix}\begin{bmatrix} y \\ \overline{u} \end{bmatrix}=\begin{bmatrix} My_d \\ 0 \end{bmatrix}=b \tag{1.9}$$

where M and K are symmetric matrix.

Obviously, the generalized saddle-point issue includes linear systems (8) as a particular instance. Bai in [6] proposed the the Hermitian and skew-Hermitian splitting (HSS) iterative method which is unconditionally convergent and easy to implementation. Subsequently, many iterative solution methods [4, 5] have emerged on the basis of HSS iterative method for the following complex linear system of the form

$$Ax=f,\ x,\ f\in\mathbb{C}^{n\times n}, \tag{1.10}$$

with $A=W+iT\in\mathbb{C}^{n\times n}$. The real matrics $W,T\in\mathbb{R}^{n\times n}$ are symmetric and positive definite matrices. The PMHSS preconditioner for the system (10) has the following structure

$$F(\alpha)=\frac{\alpha+1}{\sqrt{2}\alpha}G_0\begin{pmatrix} \alpha W+T & 0 \\ 0 & \alpha W+T \end{pmatrix},\ \text{with } G_0=\frac{1}{\sqrt{2}}\begin{pmatrix} I & -I \\ I & I \end{pmatrix} \tag{1.11}$$

where α is a prescribed real constant.

As long as matrices W and T possess the feature of being symmetric positive semidefinite, the convergence theory for these PMHSS algorithms was developed. The authors examined the spectral characteristics of the PMHSS preconditioned matrix $F^{-1}(\alpha)A$. The PMHSS preconditioner was quite competitive, according to numerical experiments, when it is utilized for generalized minimum residual (GMRES) iter. The convergence theory for these PMHSS algorithms was developed by the authors.ation techniques in Krylov subspace.

Bai in [3] has derived a class of rotated block triangular preconditioners, which have the following form:

$$L(\alpha)=\frac{1}{\sqrt{\alpha^2+1}}G\begin{pmatrix} \alpha W+T & 0 \\ \alpha T-W & \alpha W+T \end{pmatrix}, \tag{1.12}$$

$$U(\alpha)=\frac{1}{\sqrt{\alpha^2+1}}G\begin{pmatrix} \alpha W+T & W-\alpha T \\ 0 & \alpha W+T \end{pmatrix}, \tag{1.13}$$

$$P(\alpha)=\frac{1}{\sqrt{\alpha^2+1}}G\begin{pmatrix} \alpha W+T & 0 \\ \alpha T-W & \alpha W+T \end{pmatrix}\begin{pmatrix} \alpha W+T & 0 \\ 0 & \alpha W+T \end{pmatrix}^{-1}\begin{pmatrix} \alpha W+T & W-\alpha T \\ 0 & \alpha W+T \end{pmatrix}. \tag{1.14}$$

In [3, 5], Bai showed that the eigenvalues of $F^{-1}(\alpha)A$. Bai also demonstrated the eigenvalues of $L^{-1}(\alpha)A$, $U^{-1}(\alpha)A$ and $P^{-1}(\alpha)A$ are clustered within a complex disk centered at 1. Numerical results in [3] have shown that the PMHSS preconditioner cannot be outperformed by the aforementioned three preconditioners.

To solve the system (9), a huge and sparse complex linear system, iterative approaches must be used in conjunction with appropriate preconditioners. The real block diagonal preconditioner and the alternative indefinite preconditioner were proposed by Krendl in [9] and they have the following forms

$$P_{BD}=\begin{bmatrix}\left(1+\sqrt{\beta}\omega\right)M+\sqrt{\beta}K & 0\\ 0 & \left(1+\sqrt{\beta}\omega\right)M+\sqrt{\beta}K\end{bmatrix}, \tag{1.15}$$

$$P_{AI}=\begin{pmatrix}0 & M+\sqrt{\beta}\left(K-i\omega M\right)\\ M+\sqrt{\beta}\left(K+i\omega M\right) & -M\end{pmatrix}. \tag{1.16}$$

It is shown that the spectrum of $P_{BD}^{-1}\mathscr{A}_I$ is contained in the set

$$\left[-1,-\frac{\sqrt{3}}{3}\right]\cup\left[\frac{\sqrt{3}}{3},1\right].$$

Inspired by the PMHSS in [5], Zheng et al. constructed the block alternating splitting (BAS) iteration method to solve the system (8) in [23]. The details of the BAS process are given next. Firstly, define the rotation matrix

$$P_1=\frac{1}{1+\omega^2\beta}\begin{bmatrix}I & -i\omega\sqrt{\beta}I\\ i\omega\sqrt{\beta}I & I\end{bmatrix},\quad P_2=\begin{bmatrix}0 & I\\ I & 0\end{bmatrix}. \tag{1.17}$$

Afterwards, pre- multiply P_1,P_2 to system (9) and use the HSS splitting method in [6]. Then the BAS iterative method is stated as

$$\begin{cases}\left(\alpha V+H_1\right)x^{\left(k+\frac{1}{2}\right)}=\left(\alpha V-S_1\right)x^{(k)}+P_1b,\\ \left(\alpha V+H_2\right)x^{(k+1)}=\left(\alpha V-S_2\right)x^{\left(k+\frac{1}{2}\right)}+P_2b,\end{cases} \tag{1.18}$$

where V is a symmetric positive definite matrix, and

$$H_1=\begin{bmatrix}M & 0\\ 0 & M\end{bmatrix},\quad S_1=\begin{bmatrix}-\frac{i\omega\beta}{1+\omega^2\beta}K & \frac{\sqrt{\beta}}{1+\omega^2\beta}K\\ -\frac{\sqrt{\beta}}{1+\omega^2\beta}K & \frac{i\omega\beta}{1+\omega^2\beta}K\end{bmatrix},$$
$$H_2=\begin{bmatrix}\sqrt{\beta}K & 0\\ 0 & \sqrt{\beta}K\end{bmatrix},\quad S_2=\begin{bmatrix}i\omega\sqrt{\beta}M & M\\ M & -i\omega\sqrt{\beta}M\end{bmatrix}.$$

The corresponding BAS preconditioner was also derived as follows

$$P_{BAS}\left(\alpha\right)=\left(1+\alpha\right)P\left(\alpha\right)\begin{bmatrix}\alpha M+\sqrt{\beta}K & 0\\ 0 & \alpha M+\sqrt{\beta}K\end{bmatrix} \tag{1.19}$$

with

$$P\left(\alpha\right)=\frac{1}{\alpha\left(2+\omega^2\beta\right)}\begin{bmatrix}I & \left(1+\omega^2\beta-i\omega\sqrt{\beta}\right)I\\ \left(1+\omega^2\beta+i\omega\sqrt{\beta}\right)I & -I\end{bmatrix}.$$

Meanwhile, Numerical experiments proved that $P_{BAS}(\alpha)$ outperforms the previous preconditioner P_{BD} and P_{AI} which Krendl et al. presented in [9]. We found that

$$P(\alpha)\cdot P(\alpha)=\frac{1+\beta\omega^2}{\alpha^2(2+\beta\omega^2)}\begin{bmatrix} I & 0\\ 0 & I\end{bmatrix}.$$

By this property, we can easily obtain the inverse matrix of $P(\alpha)$ as $P^{-1}(\alpha)=\frac{\alpha^2(2+\beta\omega^2)}{1+\beta\omega^2}P(\alpha)$. The algorithmic implementation of $P_{BAS}(\alpha)$ in both a straightforward splitting iteration process and the Krylov acceleration is then taken into consideration. We must resolve the following linear system in an implementation: $P_{BAS}(\alpha)z=r$ with $r=\left(r_a^T,r_b^T\right)^T$ and $z=\left(z_a^T,z_b^T\right)^T$. Hence, the following algorithm can be stated for computing the vector z:

Solve $\left(\alpha M+\sqrt{\beta}K\right)z_1=\frac{\alpha^2}{(1+\alpha)\cdot(1+\beta\omega^2)}\left[r_1+\left(1+\beta\omega^2-i\omega\sqrt{\beta}\right)r_2\right]$.

Slove $\left(\alpha M+\sqrt{\beta}K\right)z_2=\frac{\alpha^2}{(1+\alpha)\cdot(1+\beta\omega^2)}\left[\left(1+\beta\omega^2+i\omega\sqrt{\beta}\right)r_1-r_2\right]$.

Zeng [19] presented a respectively scaled splitting-like (RSS-like) iteration method and induced the corresponding preconditioner for (9):

$$\begin{aligned}&P_{RSS-\text{like}}(\alpha)\\ =&\frac{1}{\alpha}\left[\alpha\sqrt{1+\beta\omega^2}M+\sqrt{\beta}K\right]\left(\sqrt{\beta}K-\sqrt{1+\beta\omega^2}GM\right)^{-1}\left[\alpha\sqrt{\beta}K+\sqrt{1+\beta\omega^2}M\right],\end{aligned}\tag{1.20}$$

where $G=\frac{1}{\sqrt{1+\beta\omega^2}}\begin{pmatrix} i\sqrt{\beta}\omega I & -I\\ I & -i\sqrt{\beta}\omega I\end{pmatrix}$. Recently, Salkuyeh [17] stated a block alternating splitting iteration (BASI) method and derived a preconditioner of the form:

$$P_{BASI}(\alpha)=\frac{1}{\alpha}\left[\alpha I+\left(1+\beta\omega^2\right)M\right](I-S)^{-1}\left[\alpha I+\sqrt{\beta\left(1+\beta\omega^2\right)}K\right],\tag{1.21}$$

where $S=\frac{1}{\sqrt{1+\beta\omega^2}}\begin{pmatrix} i\sqrt{\beta}\omega I & I\\ -I & i\sqrt{\beta}\omega I\end{pmatrix}$. When they are practically applied to the Krylov subspace method, it is usually necessary to solve for $P_{RSS-\text{like}}(\alpha)z=r$ and $P_{BASI}(\alpha)z=r$. Based on the above two preconditioners, we have the following algorithmic implementation respectively:

1. Computing z from $P_{RSS-\text{like}}(\alpha)z=r$.

Solve $\left(\alpha\sqrt{1+\beta\omega^2}M+\sqrt{\beta}K\right)h_1=\alpha r_1$.

Solve $\left(\alpha\sqrt{1+\beta\omega^2}M+\sqrt{\beta}K\right)h_2=\alpha r_1$.

Solve $\left(\alpha\sqrt{\beta}K+\sqrt{1+\beta\omega^2}M\right)z_1=\sqrt{\beta}(K-i\omega M)h_1+Mh_2$.

Solve $\left(\alpha\sqrt{\beta}K+\sqrt{1+\beta\omega^2}M\right)z_2=-Mh_1+\sqrt{\beta}(K+i\omega M)h_2$.

2. Computing z from $P_{BASI}(\alpha)z=r$.

Solve $\left[\alpha I+\left(1+\beta\omega^2\right)M\right]h_1=\alpha r_1$.

Solve $\left[\alpha I+\left(1+\beta\omega^2\right)M\right]h_2=\alpha r_2$.

Solve $\left[\alpha I+\sqrt{\beta\left(1+\beta\omega^2\right)}K\right]z_1=\left(1+\frac{i\omega\sqrt{\beta}}{\sqrt{1+\beta\omega^2}}\right)h_1-\frac{1}{\sqrt{1+\beta\omega^2}}h_2$.

Solve $\left[\alpha I+\sqrt{\beta\left(1+\beta\omega^2\right)}K\right]z_2=\frac{1}{\sqrt{1+\beta\omega^2}}h_1-\left(1-\frac{i\omega\sqrt{\beta}}{\sqrt{1+\beta\omega^2}}\right)h_2$.

As we have discussed, two systems with the coefficient matrices $\alpha M+\sqrt{\beta}K$ can be solved for the GMRES method in conjunction with BAS preconditioner. However, during the execution of the BASI preconditioner and RSS-like preconditioner we need to solve four subsystems,respectively.

By rewriting the system (9) as the 4-by-4 block real system, Zeng [18] presented a respectively scaled splitting (RSS) iteration method. The Alternating Symmetric Positive Definite and Scaled Symmetric Positive Semidefinite Splitting (ASSS) for System (9) was presented by Salkuyeh in [16]. These two methods can be seen as an application of RSS-like methods and BASI methods to fourth-order linear systems.

Inspired by the structure and effectiveness of the BAS preconditioner and the rotated block preconditioning matrices in [3, 23], we establish a class of complex rotated block triangular preconditioners for the complex linear system (9) in this paper. These preconditioning matrices avoid solving systems with complex coefficient matrices and have the rotated matrix that is easy to invert.

The structure of this essay is as follows. We introduce the complex rotated block preconditioning matrices and outline the computing process for using these matrices to solve the generalized residual equations in section 2. In section 3, we explore the spectral characteristics of the preconditioned matrices. When compared to the related BAS preconditioners, numerical experiments show that these preconditioners are more effective in section 4.

2. Complex rotated triangular preconditioning

Inspired by the construction and efficiency of BAS preconditioner, we are able to develop a class of complex rotated block triangular preconditioners for the coefficient matrix $\mathscr{A}_1$ defined in (9) by following the approach in [10, 20, 22]:

$$\mathcal{P}_L=G\begin{pmatrix}\alpha tM+\sqrt{\beta}K & 0\\ -\left(\alpha\sqrt{\beta}K-tM\right) & \alpha tM+\sqrt{\beta}K\end{pmatrix},\tag{2.1}$$

$$\mathcal{P}_U=G\begin{pmatrix}\alpha tM+\sqrt{\beta}K & -\left(tM-\alpha\sqrt{\beta}K\right)\\ 0 & \alpha tM+\sqrt{\beta}K\end{pmatrix},\tag{2.2}$$

and rotated block triangular product preconditioner

$$\begin{aligned}\mathcal{P}_{TP}=&G\begin{pmatrix}\alpha tM+\sqrt{\beta}K & 0\\ -\left(\alpha\sqrt{\beta}K-tM\right) & \alpha tM+\sqrt{\beta}K\end{pmatrix}\begin{pmatrix}\alpha tM+\sqrt{\beta}K & 0\\ 0 & \alpha tM+\sqrt{\beta}K\end{pmatrix}^{-1}\\ &\times\begin{pmatrix}\alpha tM+\sqrt{\beta}K & -\left(tM-\alpha\sqrt{\beta}K\right)\\ 0 & \alpha tM+\sqrt{\beta}K\end{pmatrix},\end{aligned}\tag{2.3}$$

where $t=1+\beta\omega^2$. The matrix $G=\frac{1}{\sqrt{1+\beta\omega^2}}\begin{pmatrix}I & -i\sqrt{\beta}\omega I\\ i\sqrt{\beta}\omega I & -I\end{pmatrix}$ is a block Givens rotation. It is easy to get $G^{-1}=G$. We assume that M is symmetric positive definite (SPD) and K is symmetric semi-positive definite (SPSD). Thus the matrix $\alpha tM+\sqrt{\beta}K$ is nonsingular.

In an effort to hasten the convergence of the Krylov subspace iterative method, a sequence of generalized residual equations of the following form must be solved when using these complex rotating block triangular preconditioners:

$$\mathcal{P}_L z = r, \qquad \mathcal{P}_U z = r, \qquad \mathcal{P}_{TP} z = r, \tag{2.4}$$

where $r = \left(r_a^T, r_b^T\right)^T$ and $z = \left(z_a^T, z_b^T\right)^T$ are the current and generalized residual vectors, respectively.

The following is how we can deduce the specific implementation approaches for computing the generalized residuals.

1. Computing z from $\mathcal{P}_L z = r$.
Solve $\left(\alpha tM + \sqrt{\beta}K\right) z_1 = \frac{1}{\sqrt{t}}\left(r_1 - i\omega\sqrt{\beta}r_2\right)$.
Slove $\left(\alpha tM + \sqrt{\beta}K\right) z_2 = \frac{1}{\sqrt{t}}\left(i\omega\sqrt{\beta}r_1 - r_2\right) + \left(\alpha\sqrt{\beta}K - tM\right) z_1$.
2. Computing z from $\mathcal{P}_U z = r$.
Solve $\left(\alpha tM + \sqrt{\beta}K\right) z_2 = \frac{1}{\sqrt{t}}\left(i\omega\sqrt{\beta}r_1 - r_2\right)$.
Slove $\left(\alpha tM + \sqrt{\beta}K\right) z_1 = \frac{1}{\sqrt{t}}\left(r_1 - i\omega\sqrt{\beta}r_2\right) + \left(tM - \alpha\sqrt{\beta}K\right) z_2$.
3. Computing z from $\mathcal{P}_{TP} z = r$.
Solve $\left(\alpha tM + \sqrt{\beta}K\right) g = \frac{1}{\sqrt{t}}\left(r_1 - i\omega\sqrt{\beta}r_2\right)$.
Slove $\left(\alpha tM + \sqrt{\beta}K\right) z_2 = \frac{1}{\sqrt{t}}\left(i\omega\sqrt{\beta}r_1 - r_2\right) + \left(\alpha\sqrt{\beta}K - tM\right) g$.
Solve $\left(\alpha tM + \sqrt{\beta}K\right) z_1 == \frac{1}{\sqrt{t}}\left(r_1 - i\omega\sqrt{\beta}r_2\right) + \left(tM - \alpha\sqrt{\beta}K\right) z_2$.

The solution of these three preconditioning matrices requires just the linear subsystems of the same coefficient matrix $\alpha tM + \sqrt{\beta}K$. By introducing the complex rotated matrix G into the preconditioners, it avoids solving the complex linear system, and it is easy to obtain that the inverse matrix of G is itself, both of which reduce the computational effort. More importantly, the block matrix is approximately estimated by the block triangular matrix more accurately than it would be estimated by the block diagonal matrix. Comparing with the BAS preconditioner in (19), the computation of the excess matrix product. i.e., $\left(\alpha\sqrt{\beta}K - tM\right) z_1$ and $\left(tM - \alpha\sqrt{\beta}K\right) z_2$ in the implementation of the algorithm is negligible since both M and K are large sparse matrices. On the basis of the above analysis, we expect better effect of the complex rotated block triangular preconditioners.

3. Analysis of eigenvalue properties

The eigenvalue properties of $\mathcal{P}_L^{-1}\mathscr{A}_1$, $\mathcal{P}_U^{-1}\mathscr{A}_1$ and $\mathcal{P}_{TP}^{-1}\mathscr{A}_1$ are derived in this section. First, we explain certain crucial notations that will be utilized in the discussions that follow. Defining the matrices

$$V(\alpha) = \left(\alpha tM + \sqrt{\beta}K\right)^{-1}\left(\alpha\sqrt{\beta}K - tM\right), \tag{3.1}$$

$$M(\alpha) = \beta i\omega\left(\alpha tM + \sqrt{\beta}K\right)^{-1} K, \tag{3.2}$$

and

$$\varphi(\alpha) = \frac{1}{\sqrt{2}}\begin{pmatrix} I & iI \\ iI & I \end{pmatrix}. \tag{3.3}$$

In addition, we introduce an orthogonal matrix

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

The resulting spectral decomposition of matrix $I(\alpha)$ is obtained as follows:

$$\varphi^*(\alpha) \begin{pmatrix} \alpha I & -I \\ I & \alpha I \end{pmatrix} \varphi(\alpha) = \begin{pmatrix} (\alpha+i) I & 0 \\ 0 & (\alpha-i) I \end{pmatrix} = \Lambda(\alpha).$$

The following spectrum characteristics apply tothe proposed complex rotated block triangular preconditioners.

Theorem 3.1. *Let $M \in R^{n\times n}$ and $K \in R^{n\times n}$ being SPD and SPSD, respectively. Then, for the preconditioning matrices $\mathcal{P}_L$, $\mathcal{P}_U$ and $\mathcal{P}_{TP}$ defined in (20), (21), and (22), it holds that*

(i) the eigenvalues λ of $\mathcal{P}_L^{-1}\mathscr{A}_1$ and $\mathcal{P}_U^{-1}\mathscr{A}_1$ are contained, respectively, within the complex disk centered at $c_{\pm j}(\alpha)$ with radius $\delta_1(\alpha)$, where $c_{\pm j}(\alpha) = \alpha \pm i$ and $\delta_1(\alpha) = \sqrt{1+\alpha^2} \cdot \|V(\alpha)\| \sqrt{1+\|V(\alpha)\|^2} + \|M(\alpha)\| \cdot (1+\|V(\alpha)\|)$;

(ii) the eigenvalues γ of $\mathcal{P}_{TP}^{-1}\mathscr{A}_1$ are contained within within the complex disk centered at $c_{\pm j}(\alpha)$ with radius $\delta_2(\alpha)$, where $c_{\pm j}(\alpha) = \alpha \pm i$ and $\delta_2(\alpha) = \sqrt{1+\alpha^2} \cdot \|V(\alpha)\|^2 \sqrt{1+\|V(\alpha)\|^2} + \|M(\alpha)\| + \|V(\alpha)\| \cdot \|M(\alpha)\| + \|V(\alpha)\| \cdot \sqrt{1+\|V(\alpha)\|^2} \cdot \|M(\alpha)\|$;

Proof. We first prove (i). Through direct computations, we obtain

$$\begin{aligned}
\mathcal{P}_L^{-1}\mathscr{A}_1 &= \begin{pmatrix} \left(\alpha t M + \sqrt{\beta}K\right)^{-1} & 0 \\ \left(\alpha t M + \sqrt{\beta}K\right)^{-1}\left(\alpha\sqrt{\beta}K - tM\right)\left(\alpha t M + \sqrt{\beta}K\right)^{-1} & \left(\alpha t M + \sqrt{\beta}K\right)^{-1} \end{pmatrix} G\mathscr{A} \\
&= \begin{pmatrix} \left(\alpha t M + \sqrt{\beta}K\right)^{-1} & 0 \\ V(\alpha)\left(\alpha t M + \sqrt{\beta}K\right)^{-1} & \left(\alpha t M + \sqrt{\beta}K\right)^{-1} \end{pmatrix} \begin{pmatrix} tM - \beta i\omega K & \sqrt{\beta}K \\ -\sqrt{\beta}K & tM + \beta i\omega K \end{pmatrix} \\
&= (\mathcal{M}_1 + \mathcal{N}_1),
\end{aligned}$$

where

$$\mathcal{M}_1 = \begin{pmatrix} \left(\alpha t M + \sqrt{\beta}K\right)^{-1} & 0 \\ V(\alpha)\left(\alpha t M + \sqrt{\beta}K\right)^{-1} & \left(\alpha t M + \sqrt{\beta}K\right)^{-1} \end{pmatrix} \begin{pmatrix} tM & \sqrt{\beta}K \\ -\sqrt{\beta}K & tM \end{pmatrix}$$

and

$$\mathcal{N}_1 = \begin{pmatrix} I & 0 \\ V(\alpha) & I \end{pmatrix} \begin{pmatrix} -M(\alpha) & 0 \\ 0 & M(\alpha) \end{pmatrix}.$$

By introducing the $I(\alpha) I(\alpha)^T = I$, we can get

$$\mathcal{M}_1 = \frac{1}{\alpha^2+1} \begin{pmatrix} \left(\alpha t M + \sqrt{\beta}K\right)^{-1} & 0 \\ V(\alpha)\left(\alpha t M + \sqrt{\beta}K\right)^{-1} & \left(\alpha t M + \sqrt{\beta}K\right)^{-1} \end{pmatrix}$$

$$
\begin{aligned}
&\times \begin{pmatrix} tM & \sqrt{\beta}K \\ -\sqrt{\beta}K & tM \end{pmatrix} \begin{pmatrix} \alpha I & -I \\ I & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix} \\
&= \frac{1}{\alpha^2+1} \begin{pmatrix} I & V(\alpha) \\ 0 & I+V^2(\alpha) \end{pmatrix} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix} \\
&= \frac{1}{\alpha^2+1} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix} + \frac{1}{\alpha^2+1} \begin{pmatrix} 0 & V(\alpha) \\ 0 & V^2(\alpha) \end{pmatrix} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix}.
\end{aligned}
$$

It follows that

$$
\mathcal{P}_{L}^{-1}\mathscr{A}_1 = \frac{1}{\alpha^2+1} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix} + \frac{1}{\alpha^2+1} \begin{pmatrix} 0 & V(\alpha) \\ 0 & V^2(\alpha) \end{pmatrix} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix} + \begin{pmatrix} I & 0 \\ V(\alpha) & I \end{pmatrix} \begin{pmatrix} -M(\alpha) & 0 \\ 0 & M(\alpha) \end{pmatrix}.
$$

Next, it is simple to deduce

$$
\varphi^*(\alpha) \cdot \mathcal{P}_{L}^{-1}\mathscr{A}_1 \cdot \varphi(\alpha) = D(a) + Y_1(\alpha),
$$

where

$$
D(a) = \frac{1}{\alpha^2+1} \Lambda(\alpha)
$$

and

$$
Y_1(\alpha) = \varphi^*(\alpha) \cdot \left[\frac{1}{\alpha^2+1} \cdot \begin{pmatrix} 0 & V(\alpha) \\ 0 & V^2(\alpha) \end{pmatrix} \begin{pmatrix} \alpha I & I \\ -I & \alpha I \end{pmatrix} + \begin{pmatrix} I & 0 \\ V(\alpha) & I \end{pmatrix} \begin{pmatrix} -M(\alpha) & 0 \\ 0 & M(\alpha) \end{pmatrix} \right] \varphi(\alpha).
$$

According to the fact that $\varphi^*(\alpha) \cdot \varphi(\alpha) = I$, we can obtain

$$
\begin{aligned}
&\|Y_1(\alpha)\| \\
\leqslant &\frac{1}{\alpha^2+1} \cdot \left\| \begin{pmatrix} 0 & V(\alpha) \\ 0 & V^2(\alpha) \end{pmatrix} \Lambda(\alpha) \right\| + \left\| \begin{pmatrix} -M(\alpha) & 0 \\ 0 & M(\alpha) \end{pmatrix} \right\| + \left\| \begin{pmatrix} 0 & 0 \\ V(\alpha) & 0 \end{pmatrix} \begin{pmatrix} -M(\alpha) & 0 \\ 0 & M(\alpha) \end{pmatrix} \right\| \\
\leqslant &\frac{1}{\sqrt{\alpha^2+1}} \|V(\alpha)\| \sqrt{1+\|V(\alpha)\|^2} + \|M(\alpha)\| \cdot (1+\|V(\alpha)\|) \\
= &\delta_1(\alpha).
\end{aligned}
$$

It follows that $\left\|\varphi^*(\alpha) \cdot \mathcal{P}_{L}^{-1}\mathscr{A}_1 \cdot \varphi(\alpha) - D(a)\right\| \leqslant \|Y_1(\alpha)\|$. Then we can get

$$
|\lambda - c_{\pm j}(\alpha)| \leqslant \delta_1(\alpha),
$$

where $c_{\pm j}(\alpha) = \frac{\alpha \pm i}{\alpha^2+1}$.

By means of the analogous way as above, we obtain that the eigenvalues of $\mathcal{P}_{L}^{-1}\mathscr{A}_1$ and $\mathcal{P}_{U}^{-1}\mathscr{A}_1$ are contained within the same union of two complex disks centered at $c_{\pm j}(\alpha)$ with radius $\delta_1(\alpha)$.

Now we demonstrate (ii) in a similar way. Throug straightforward calculation, we get

$$
\mathcal{P}_{TP}^{-1}\mathscr{A}_1 = I(\alpha)\, \mathcal{P}_{L}^{-1}\mathscr{A}_1,
$$

where

$$I(\alpha)=\begin{pmatrix} I & -V(\alpha) \\ 0 & I \end{pmatrix}.$$

Utilizing the outcomes of Theorem 4.2 (i), we have

$$\varphi^*(\alpha)\cdot\mathcal{P}_{TP}^{-1}\mathscr{A}_1\cdot\varphi(\alpha)=D(\alpha)+Y_2(\alpha),$$

where

$$\begin{aligned} Y_2(\alpha)=&\varphi^*(\alpha)\left[\frac{1}{\alpha^2+1}\begin{pmatrix} 0 & -V^3(\alpha) \\ 0 & V^2(\alpha)\end{pmatrix}\begin{pmatrix}\alpha I & I \\ -I & \alpha I\end{pmatrix}\right. \\ &\left.+\begin{pmatrix} I & -V(\alpha) \\ 0 & I\end{pmatrix}\begin{pmatrix} I & 0 \\ V(\alpha) & I\end{pmatrix}\begin{pmatrix}-M(\alpha) & 0 \\ 0 & M(\alpha)\end{pmatrix}\right]\varphi(\alpha). \end{aligned}$$

Analogously to the derivation of (i), we have

$$\begin{aligned} \|Y_3(\alpha)\|\leqslant&\frac{1}{\sqrt{\alpha^2+1}}\cdot\|V(\alpha)\|^2\sqrt{1+\|V(\alpha)\|^2}+\|M(\alpha)\|+\|V(\alpha)\|\cdot\|M(\alpha)\| \\ &+\|V(\alpha)\|\cdot\sqrt{1+\|V(\alpha)\|^2}\cdot\|M(\alpha)\| \\ =&\delta_2(\alpha). \end{aligned}$$

Hence, we can derive

$$|\gamma-c_{\pm j}(\alpha)|\leqslant\delta_2(\alpha).$$

□

Theorem 3.2. *Assume that $K\in R^{n\times n}$ is SPSD and $M\in R^{n\times n}$ is SPD. Thus, for the preconditioned matrices $\mathcal{P}_L^{-1}\mathscr{A}_1$, $\mathcal{P}_U^{-1}\mathscr{A}_1$, and $\mathcal{P}_{TP}^{-1}\mathscr{A}_1$, it holds that*

(i) the eigenvalues of the preconditioned matrix $\mathcal{P}_L^{-1}\mathscr{A}_1$ and $\mathcal{P}_U^{-1}\mathscr{A}_1$ are

$$\lambda_{\pm j}=b+\frac{ak}{2}\pm\frac{1}{2}\sqrt{\left(4+k^2-2\omega^2\beta\right)a^2+4kab+2b^2},\ j=1,2,\cdots,n,$$

(ii) the eigenvalues of the preconditioned matrix $\mathcal{P}_{TP}^{-1}\mathscr{A}_1$ are

$$\begin{aligned}&\sigma_{\pm j}\\ =&b+\frac{ak-cn}{2}\pm\frac{1}{2}\sqrt{\left(4+k^2-2\omega^2\beta\right)a^2+4kab+2b^2+cn\left(cn+4-4b-2ak\right)},\ j=1,2,\cdots,n.\end{aligned}$$

Proof. (i) Since M is SPD, we can define

$$\mathcal{P}_1=\begin{pmatrix} M^{-\frac{1}{2}} & 0 \\ 0 & M^{-\frac{1}{2}}\end{pmatrix}. \tag{3.4}$$

Then, the preconditioned matrix $\mathcal{P}_L^{-1}\mathscr{A}_1$ is similar to ${\mathcal{P}_1}^{-1}\mathcal{P}_L^{-1}\mathscr{A}_1\mathcal{P}_1$, which is

$${\mathcal{P}_1}^{-1}\mathcal{P}_L^{-1}\mathscr{A}\mathcal{P}_1=\begin{pmatrix}\widehat{M_1(\alpha)} & \widehat{U(\alpha)} \\ \widehat{V(\alpha)}\widehat{M_1(\alpha)}-\widehat{U(\alpha)} & \sqrt{\beta}V(\alpha)\left(\alpha tI+\sqrt{\beta}\widehat{K}\right)^{-1}K+\widehat{M_2(\alpha)}\end{pmatrix}$$

with

$$\begin{aligned}
&\widehat{K} = M^{-\frac{1}{2}} K M^{-\frac{1}{2}},\ \widehat{V(\alpha)} = \left(\alpha t I + \sqrt{\beta}\widehat{K}\right)^{-1}\left(\alpha\sqrt{\beta}\widehat{K} - tI\right),\\
&\widehat{M_1(\alpha)} = \left(\alpha t I + \sqrt{\beta}\widehat{K}\right)^{-1}\left(tI - \beta i\omega\widehat{K}\right),\\
&\widehat{M_2(\alpha)} = \left(\alpha t I + \sqrt{\beta}\widehat{K}\right)^{-1}\left(tI + \beta i\omega\widehat{K}\right),\ \widehat{U(\alpha)} = \sqrt{\beta}\left(\alpha t I + \sqrt{\beta}\widehat{K}\right)^{-1}\widehat{K}.
\end{aligned}$$

Denote

$$\mathcal{P}_2 = \begin{pmatrix} U^T & 0 \\ 0 & U^T \end{pmatrix}, \tag{3.5}$$

where $U \in R^{n\times n}$ is an orthogonal matrix. Suppose the eigenvalue decomposition of the symmetric matrix $\widehat{K}$ is

$$\widehat{K} = U^T \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} U,$$

where $\Lambda = \operatorname{diag}\{\mu_1, \mu_2, \cdots, \mu_r\})$ is a diagonal matrix with $\mu_1 \geqslant \mu_2 \geqslant \cdots \geqslant \mu_r > 0$ being the nonzero eigenvalues of the matrix $\widehat{K}$ and $r = \operatorname{rank}\left(\widehat{K}\right) \leqslant n$.

We know that ${\mathcal{P}_1}^{-1}\mathcal{P}_L^{-1}\mathscr{A}_1\mathcal{P}_1$ is similar to ${\mathcal{P}_2}^{-1}{\mathcal{P}_1}^{-1}\mathcal{P}_L^{-1}\mathscr{A}_1\mathcal{P}_1\mathcal{P}_2$, where

$${\mathcal{P}_2}^{-1}{\mathcal{P}_1}^{-1}\mathcal{P}_L^{-1}\mathscr{A}_1\mathcal{P}_1\mathcal{P}_2 = \begin{pmatrix} \widetilde{M_1(\alpha)} & 0 & \widetilde{U(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \\ \widetilde{V(\alpha)}\widetilde{M_1(\alpha)} - \widetilde{U(\alpha)} & 0 & \widetilde{V(\alpha)}\widetilde{U(\alpha)} + \widetilde{M_2(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.6}$$

with

$$\begin{aligned}
&\widetilde{V(\alpha)} = \left(\alpha t I + \sqrt{\beta}\Lambda\right)^{-1}\left(\alpha\sqrt{\beta}\Lambda - tI\right),\quad \widetilde{M_1(\alpha)} = \left(\alpha t I + \sqrt{\beta}\Lambda\right)^{-1}(tI - \beta i\omega\Lambda),\\
&\widetilde{M_2(\alpha)} = \left(\alpha t I + \sqrt{\beta}\Lambda\right)^{-1}(tI + \beta i\omega\Lambda),\quad \widetilde{U(\alpha)} = \sqrt{\beta}\left(\alpha t I + \sqrt{\beta}\Lambda\right)^{-1}\Lambda.
\end{aligned}$$

Denote

$$T = \begin{pmatrix} b - \sqrt{\beta}i\omega a & a \\ k\left(b - \sqrt{\beta}i\omega a\right) - a & ka + \left(b + \sqrt{\beta}i\omega a\right) \end{pmatrix}, \tag{3.7}$$

where

$$b = \frac{t}{\left(\alpha t + \sqrt{\beta}\mu\right)},\ a = \frac{\sqrt{\beta}\mu}{\left(\alpha t + \sqrt{\beta}\mu\right)},\ k = \frac{\left(\alpha\sqrt{\beta}u - t\right)}{\left(\alpha t + \sqrt{\beta}\mu\right)}.$$

Then, it is simple to confirm that the eigenvalues $\lambda_{\pm j}$ are

$$\lambda_{\pm j} = b + \frac{ak}{2} \pm \frac{1}{2}\sqrt{\left(4 + k^2 - 2\omega^2\beta\right)a^2 + 4kab + 2b^2},\ j = 1, 2, \cdots, n. \tag{3.8}$$

(ii) Finally, we prove (ii) according to (i).

It is straightforward to validate that $\mathcal{P}_{TP}^{-1}\mathscr{A}_1 = I(\alpha)\mathcal{P}_L^{-1}\mathscr{A}_1$ is similar to ${\mathcal{P}_1}^{-1}I(\alpha)\mathcal{P}_L^{-1}\mathscr{A}_1\mathcal{P}_1$, where

$$ {\mathcal{P}_1}^{-1}I(\alpha)\mathcal{P}_L^{-1}\mathscr{A}_1\mathcal{P}_1 = \begin{pmatrix} I & -\widehat{V(\alpha)} \\ 0 & I \end{pmatrix} \begin{pmatrix} \widehat{M_1(\alpha)} & \widehat{U(\alpha)} \\ \widehat{V(\alpha)}\widehat{M_1(\alpha)} - \widehat{U(\alpha)} & V(\alpha)\widehat{U(\alpha)} + \widehat{M_2(\alpha)} \end{pmatrix}. $$

Then, based on (28), we have

$$ \begin{aligned} & {\mathcal{P}_2}^{-1}{\mathcal{P}_1}^{-1}\widehat{I(\alpha)}\mathcal{P}_L^{-1}\mathscr{A}\mathcal{P}_1\mathcal{P}_2 \\ = & \begin{pmatrix} I & 0 & -\widetilde{V(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \widetilde{M_1(\alpha)} & 0 & \widetilde{U(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \\ \widetilde{V(\alpha)}\widetilde{M_1(\alpha)} - \widetilde{U(\alpha)} & 0 & \widetilde{V(\alpha)}\widetilde{U(\alpha)} + \widetilde{M_2(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ = & \begin{pmatrix} \widetilde{M_1(\alpha)} - \widetilde{V(\alpha)}\left(\widetilde{V(\alpha)}\widetilde{M_1(\alpha)} - \widetilde{U(\alpha)}\right) & 0 & \widetilde{U(\alpha)} - \widetilde{V(\alpha)}\left(\widetilde{V(\alpha)}\widetilde{U(\alpha)} + \widetilde{M_2(\alpha)}\right) & 0 \\ 0 & 0 & 0 & 0 \\ \widetilde{V(\alpha)}\widetilde{M_1(\alpha)} - \widetilde{U(\alpha)} & 0 & \widetilde{V(\alpha)}\widetilde{U(\alpha)} + \widetilde{M_2(\alpha)} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} $$

Denote

$$ T_{TP} = \begin{pmatrix} \left(b - \sqrt{\beta}i\omega a\right) - k\left(k\left(b - \sqrt{\beta}i\omega a\right) - a\right) & a - k\left(ka + b + \sqrt{\beta}i\omega a\right) \\ k\left(b - \sqrt{\beta}i\omega a\right) - a & ka + b + \sqrt{\beta}i\omega a \end{pmatrix}. \tag{3.9} $$

By straightforward computations, We have the nonzero eigenvalue

$$ \begin{aligned} \sigma_{\pm j} = & b + \frac{ak - cn}{2} \\ & \pm \frac{1}{2}\sqrt{\left(4 + k^2 - 2\omega^2\beta\right)a^2 + 4kab + 2b^2 + cn\left(cn + 4 - 4b - 2ak\right)}, \; j = 1, 2, \cdots, n \end{aligned} \tag{3.10} $$

with

$$ c = \left(k - \sqrt{\beta}i\omega\right)a + b; n = -k^2 b + k^2\sqrt{\beta}i\omega a + ka. $$

□

4. Numerical results

We use the results of numerical examples to compare a class of the complex block triangular preconditioner and BAS preconditioner (P-BAS) for solving system (9). In order to achieve this, the GMRES iteration method is used in conjunction with the $\mathcal{P}_L$, $\mathcal{P}_U$ and $\mathcal{P}_{TP}$ preconditioner. In the implementation, we choose the optimal parameters of BAS preconditioned iteration methods for comparison. Several experiments are performed for varying parameters ω, ε and different levels of grid refinement.The maximum iteration steps is set to 1000 and the tolerance is set to 10^{-6}. The computer was configured with Intel(R) Core(TM) i5-8250U CPU (1.60GHz 1.80 GHz). All runs are carried out in the MATLAB R2018a.

Example 4.1. [1] The numerical test are performed with $\varOmega = [0,1] \times [0,1] \in \mathbb{R}^{n\times n}$. The area $\varOmega$ is divided into two parts, one part is $\varOmega_1 = \{x \in \varOmega | \, x_1 > x_2\}$ and the other part is $\varOmega_2 = \varOmega \backslash \varOmega_1$. The various parameters are set as $\sigma = 1$, $\nu = 1$, $\beta = 10^{-8}$. Target state $\widehat{y}_d\,(x)$ is selected as

$$\widehat{y}_d\,(x)|_{\varOmega_1} = \begin{bmatrix} \sin\left(2\pi x_1\right) + 2\pi\cos\left(2\pi x_1\right)\left(x_2 - x_1\right) \\ \sin\left(\left(x_2 - x_1\right)^2\left(x_1 - 1\right)^2 x_2\right) - \sin\left(2\pi x_1\right) \end{bmatrix}$$

and $\widehat{y}_d\,(x)|_{\varOmega_2} = 0$.

Firstly, the relationship of the matrix order and the degree of grid refinement is given in the following Table 1 in the 2D case.

Table 1. Relationship between grid refinement and matrix order in the 2D case.

level	Order
1	56
2	208
3	800
4	3136

From the perspective of numerical experiments, the parameter α in complex rotating block triangular preconditioners is optimized. We specifically analyze how this parameter α affects the computation time of the three related preconditioned iteration methods for solving linear systems (9). Let α change from 0.1–20 with the interval of 0.1. Figure 1 show the dependencies for $\mathcal{P}_L$-GMRES, $\mathcal{P}_U$-GMRES and $\mathcal{P}_{TP}$-GMRES methods while the grid refinemen level =2 and $\beta = 10^{-12}$, respectively.

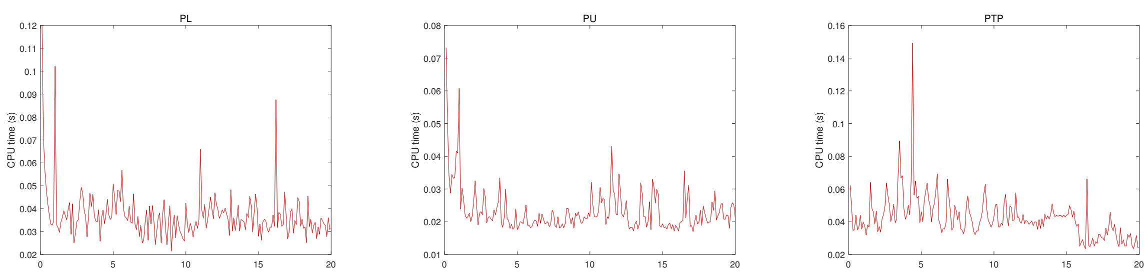

Figure 1. CPU verse α when $\beta = 10^{-12}$, $\omega = 1$, level=2.

Apart from a few special points, it is clear that, within a given range, the performance of these approaches does not strongly depend on α. And based on the Figure 1 we can find an optimal α for different parameters β, ω and grid refinement level, which makes these method cost least computational time. When α valuing 10, all three complex rotated block triangular preconditioners spend very little computational time. Therefore, we set $\alpha = 10$ during the entire computing process.

Then Numerical results for the iteration steps (IT) and the computing time (CPU) with varying $\omega \in \left\{10^{-2}, 10^{-1}, 1, 10^1, 10^2\right\}$ and $\beta \in \left\{10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}\right\}$ have been presented in Tables 2-5.

Tables 2-5 show the iteration counts and computational time of complex rotated block triangular preconditioner preconditioned GMRES methods versus BAS preconditioned GMRES methods for solving system (9). The convergence behavior of the GMRES algorithm is always improved by all preconditioners. We can observe that the GMRES method with the $\mathcal{P}_{TP}$ consumes iteration steps than the corresponding PGMRES method with the $\mathcal{P}_L$ and $\mathcal{P}_U$ when the mesh refinement level is very small. In general, the three complex rotating block preconditioners are less affected by the parameters β and ω. From either the point of view of iteration steps or computing time, it demonstrates that the preconditioners $\mathcal{P}_L$, $\mathcal{P}_U$ and $\mathcal{P}_{TP}$ are preferable to the BAS preconditioner.

5. Conclusions

In order to the complex linear systems (9) obtained by a class of finite element discretizations, a class of complex rotated block triangular preconditioners were proposed. Eigenvalue distributions of the preconditioners are characterized. Numerical tests confirm the reliable performance of the three preconditioners within Krylov subspace acceleration.

Table 2. Numerical results of the PGMRES with mesh refinement level=1 and different ω and β in the 2D case.

Method	β	$\omega=10^{-2}$		$\omega=10^{-1}$		$\omega=10^{0}$		$\omega=10^{1}$		$\omega=10^{2}$	
		IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU
P-BAS	10^{-14}	4	0.0611	4	0.0247	4	0.0075	6	0.0088	6	0.0118
	10^{-12}	4	0.0069	4	0.0112	6	0.0125	6	0.0088	6	0.0122
	10^{-10}	6	0.0186	7	0.0165	7	0.0152	7	0.0205	7	0.0378
	10^{-8}	10	0.0089	10	0.0108	12	0.0153	12	0.0093	12	0.0095
$\mathcal{P}_L$	10^{-14}	2	0.0476	2	0.0247	2	0.0271	2	0.0062	3	0.0126
	0^{-12}	3	0.0055	3	0.0073	3	0.0066	4	0.0046	5	0.0095
	10^{-10}	4	0.0157	4	0.0058	5	0.0142	6	0.0060	6	0.0071
	10^{-8}	4	0.0046	5	0.0067	6	0.0121	6	0.0151	6	0.0056
$\mathcal{P}_U$	10^{-14}	2	0.0316	2	0.0170	2	0.0186	2	0.0074	3	0.0126
	10^{-12}	3	0.0053	3	0.0105	3	0.0102	4	0.0049	4	0.0065
	10^{-10}	3	0.0043	3	0.0046	4	0.0047	4	0.0088	4	0.0058
	10^{-8}	4	0.0047	5	0.0067	5	0.0052	5	0.0052	6	0.0057
$\mathcal{P}_{TP}$	10^{-14}	2	0.0495	2	0.0236	2	0.0291	2	0.0118	3	0.0203
	10^{-12}	3	0.0095	3	0.0171	3	0.0111	3	0.0140	4	0.0084
	10^{-10}	4	0.0154	4	0.0062	4	0.0066	5	0.0083	5	0.0068
	10^{-8}	4	0.0105	5	0.0167	6	0.0145	6	0.0090	6	0.0178

Acknowledgements

The authors would like to express their great thankfulness to the referees for the comments and constructive suggestions, which are valuable in improving the quality of the original paper.

Table 3. Numerical results of the PGMRES with mesh refinement level=2 and different ω and β in the 2D case.

Method	β	$\omega = 10^{-2}$		$\omega = 10^{-1}$		$\omega = 10^{0}$		$\omega = 10^{1}$		$\omega = 10^{2}$	
		IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU
P-BAS	10^{-14}	4	0.0968	4	0.0627	4	0.0391	6	0.0516	6	0.0486
	10^{-12}	4	0.0397	4	0.0389	6	0.0614	6	0.0424	6	0.0642
	10^{-10}	6	0.0556	8	0.0859	8	0.0568	10	0.0399	10	0.0518
	10^{-8}	12	0.0561	14	0.0544	15	0.0695	15	0.0531	16	0.0667
$\mathcal{P}_L$	10^{-14}	3	0.0645	3	0.0415	3	0.0307	3	0.0254	3	0.0252
	0^{-12}	4	0.0257	4	0.0244	4	0.0226	4	0.0203	5	0.0303
	10^{-10}	4	0.0218	4	0.0218	5	0.0290	6	0.0242	6	0.0349
	10^{-8}	5	0.0552	6	0.0384	6	0.0402	7	0.0361	7	0.0415
$\mathcal{P}_U$	10^{-14}	3	0.0752	3	0.0450	3	0.0372	3	0.0273	3	0.0256
	10^{-12}	3	0.0291	3	0.0381	3	0.0230	4	0.0313	4	0.0472
	10^{-10}	3	0.0254	3	0.0296	4	0.0402	4	0.0252	4	0.0394
	10^{-8}	5	0.0491	6	0.0400	6	0.0410	7	0.0486	7	0.0522
$\mathcal{P}_{TP}$	10^{-14}	2	0.1032	2	0.0490	2	0.0389	2	0.0333	3	0.0444
	10^{-12}	3	0.0358	3	0.0321	3	0.0485	4	0.0471	4	0.0579
	10^{-10}	4	0.0468	4	0.0460	5	0.0431	5	0.0491	5	0.0524
	10^{-8}	4	0.0615	5	0.0564	6	0.0376	6	0.0531	6	0.0497

Table 4. Numerical results of the PGMRES with mesh refinement level=3 and different ω and β in the 2D case.

Method	β	$\omega = 10^{-2}$		$\omega = 10^{-1}$		$\omega = 10^{0}$		$\omega = 10^{1}$		$\omega = 10^{2}$	
		IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU
P-BAS	10^{-14}	4	1.1138	4	1.1139	4	1.0574	6	1.5378	6	1.4954
	10^{-12}	6	1.5018	6	1.4562	7	1.6896	7	1.7130	7	1.7321
	10^{-10}	10	2.7237	10	2.5833	10	2.3753	12	2.8346	12	2.8867
	10^{-8}	15	3.1393	17	3.3115	17	3.6235	17	3.5243	18	3.5540
$\mathcal{P}_L$	10^{-14}	3	0.5954	3	0.5784	3	0.5703	3	0.5548	4	0.7295
	0^{-12}	4	0.7209	4	0.7181	4	0.8363	5	1.0941	5	0.8937
	10^{-10}	4	0.7400	4	0.7215	5	0.8661	6	1.0678	6	1.0752
	10^{-8}	7	1.1888	7	1.2445	7	1.3012	8	1.3896	9	1.6114
$\mathcal{P}_U$	10^{-14}	3	1.0870	3	1.1167	3	0.9598	3	0.9494	3	1.2400
	10^{-12}	3	0.9485	3	0.9473	3	0.9339	4	1.1879	4	1.3472
	10^{-10}	4	1.1918	4	0.9351	5	1.1220	5	1.1867	5	1.1543
	10^{-8}	7	1.4965	7	1.4974	8	1.5495	9	1.6884	9	1.7272
$\mathcal{P}_{TP}$	10^{-14}	2	0.9599	2	0.8882	2	0.8987	2	0.9127	3	1.2182
	10^{-12}	3	1.2221	3	1.2036	3	1.2077	4	1.5202	5	1.8906
	10^{-10}	4	1.5608	4	1.5534	5	1.8721	5	1.8134	6	1.7764
	10^{-8}	6	1.7331	6	1.7223	7	2.0123	8	2.2785	8	2.3062

Table 5. Numerical results of the PGMRES with mesh refinement level=4 and different ω and β in the 2D case.

Method	β	$\omega = 10^{-2}$		$\omega = 10^{-1}$		$\omega = 10^{0}$		$\omega = 10^{1}$		$\omega = 10^{2}$	
		IT	CPU	IT	CPU	IT	CPU	IT	CPU	IT	CPU
P-BAS	10^{-14}	4	14.3536	4	14.3125	4	14.3339	6	20.3618	6	20.2670
	10^{-12}	6	20.4119	6	19.0878	8	21.4811	8	21.2408	8	21.4369
	10^{-10}	12	29.1084	12	29.5337	14	35.7062	14	32.0740	16	40.6884
	10^{-8}	15	46.4966	17	61.3391	17	65.7112	17	64.7597	18	51.1480
$\mathcal{P}_L$	10^{-14}	3	9.6754	3	13.4133	3	13.8699	3	13.6600	4	13.7360
	0^{-12}	4	17.7005	4	17.4383	4	17.7635	5	22.3127	5	22.3232
	10^{-10}	5	20.2926	5	20.3405	6	23.4348	6	22.7477	7	22.9843
	10^{-8}	19	66.3471	19	64.1450	20	64.3524	20	60.8923	21	62.8647
$\mathcal{P}_U$	10^{-14}	3	15.8031	3	15.3658	3	16.3402	3	16.6293	4	18.8816
	10^{-12}	3	14.8248	3	14.8438	3	13.8766	4	17.4779	4	18.2458
	10^{-10}	5	21.3790	5	21.5077	6	25.5934	6	25.5821	6	25.4894
	10^{-8}	19	62.9452	19	62.7072	19	58.6153	20	50.7517	21	54.1127
$\mathcal{P}_{TP}$	10^{-14}	3	13.2457	3	13.2583	3	13.1888	3	13.14807	3	13.1913
	10^{-12}	4	17.3823	4	16.7204	4	16.6493	4	16.6737	5	20.2346
	10^{-10}	4	16.7149	4	16.6876	5	20.1335	6	23.9013	6	24.0311
	10^{-8}	22	76.6241	23	110.5328	23	84.9290	23	80.8778	24	84.3824

References

[1] O. Axelsson and Z.-Z. Liang, *A note on preconditioning methods for time-periodic eddy current optimal control problems*, Journal of Computational and Applied Mathematics, 2019, 352, 262–277.

[2] Z.-Z. Bai, *Block preconditioners for elliptic pde-constrained optimization problems*, Computing, 2011, 91(4), 379–395.

[3] Z.-Z. Bai, *Rotated block triangular preconditioning based on pmhss*, Science China Mathematics, 2013, 56, 2523–2538.

[4] Z.-Z. Bai, M. Benzi and F. Chen, *Modified hss iteration methods for a class of complex symmetric linear systems*, Computing, 2010, 87(3), 93–111.

[5] Z.-Z. Bai, M. Benzi, F. Chen and Z.-Q. Wang, *Preconditioned mhss iteration methods for a class of block two-by-two linear systems with applications to distributed control problems*, IMA Journal of Numerical Analysis, 2013, 33(1), 343–369.

[6] Z.-Z. Bai, G. H. Golub and M. K. Ng, *Hermitian and skew-hermitian splitting methods for non-hermitian positive definite linear systems*, SIAM Journal on Matrix Analysis and Applications, 2003, 24(3), 603–626.

[7] M. Gunzburger and C. Trenchea, *Optimal control of the time-periodic mhd equations*, Nonlinear Analysis: Theory, Methods & Applications, 2005, 63(5–7), e1687–e1699.

[8] M. Kollmann and M. Kolmbauer, *A preconditioned minres solver for time-periodic parabolic*

optimal control problems, Numerical Linear Algebra with Applications, 2013, 20(5), 761–784.

[9] W. Krendl, V. Simoncini and W. Zulehner, *Stability estimates and structural spectral properties of saddle point problems*, Numerische Mathematik, 2013, 124(1), 183–213.

[10] C. Lang and Z.-R. Ren, *Inexact rotated block triangular preconditioners for a class of block two-by-two matrices*, Journal of Engineering Mathematics, 2015, 93(1), 87–98.

[11] G. Leugering, S. Engell, A. Griewank, et al., *Constrained Optimization and Optimal Control for Partial Differential Equations*, 160, Springer Science & Business Media, 2012.

[12] Y.-R. Li, X.-H. Shao and S.-Y. Li, *New preconditioned iteration method solving the special linear system from the pde-constrained optimal control problem*, Mathematics, 2021, 9(5), 510.

[13] H. Mirchi and D. K. Salkuyeh, *A new iterative method for solving the systems arisen from finite element discretization of a time-harmonic parabolic optimal control problems*, Mathematics and Computers in Simulation, 2021, 185, 771–782.

[14] H. J. Pesch, *Optimal control of dynamical systems governed by partial differential equations: A perspective from real-life applications*, IFAC Proceedings Volumes, 2012, 45(2), 1–12.

[15] T. Rees, H. S. Dollar and A. J. Wathen, *Optimal solvers for pde-constrained optimization*, SIAM Journal on Scientific Computing, 2010, 32(1), 271–298.

[16] D. K. Salkuyeh, *A new iterative method for solving a class of two-by-two block complex linear systems*, Calcolo, 2021, 58(4), 42.

[17] D. K. Salkuyeh and M. Pourbagher, *On the Solution of the Distributed Optimal Control Problem with Time-Periodic Parabolic Equations*, Mathematical Methods in the Applied Sciences, 2022.

[18] M.-L. Zeng, *Respectively scaled splitting iteration method for a class of block 4-by-4 linear systems from eddy current electromagnetic problems*, Japan Journal of Industrial and Applied Mathematics, 2021, 38, 489–501.

[19] M.-L. Zeng, *The rss-like iteration method for block two-by-two linear systems from time-periodic parabolic optimal control problems*, Applied Mathematics and Computation, 2021, 410, 126477.

[20] M.-L. Zeng and G.-F. Zhang, *Parameterized rotated block preconditioning techniques for block two-by-two systems with application to complex linear systems*, Computers & Mathematics with Applications, 2015, 70(12), 2946–2957.

[21] G. Zhang and Z. Zheng, *Block-symmetric and block-lower-triangular preconditioners for pde-constrained optimization problems*, Journal of Computational Mathematics, 2013, 370–381.

[22] J. Zhang and H. Dai, *Inexact splitting-based block preconditioners for block two-by-two linear systems*, Applied Mathematics Letters, 2016, 60, 89–95.

[23] Z. Zheng, G.-F. Zhang and M.-Z. Zhu, *A block alternating splitting iteration method for a class of block two-by-two complex linear systems*, Journal of Computational and Applied Mathematics, 2015, 288, 203–214.

Received August 2023; Accepted October 2024; Available online January 2026.