Journal of Computational and Applied Mathematics 288 (2015) 203–214

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# A block alternating splitting iteration method for a class of block two-by-two complex linear systems[☆]

Zhong Zheng[a], Guo-Feng Zhang[a,*], Mu-Zheng Zhu[a,b]

[a] School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, PR China
[b] School of Mathematics and Statistics, Hexi University, Zhangye 734000, PR China

ARTICLE INFO

*Article history:*
Received 10 February 2014
Received in revised form 6 February 2015

*MSC:*
65F08
65F10
65F20
65F50
65W05

*Keywords:*
Saddle-point problems
Iteration methods
Preconditioning
Time-periodic parabolic equations
Discretize-then-optimization

ABSTRACT

Optimization problems with partial differential equations as constraints arise widely in many areas of science and engineering. In this paper, we focus on solving a class of block two-by-two complex linear systems arising from the distributed optimal control with time-periodic parabolic equations. A new block alternating splitting (BAS) iteration method is presented for solving the class of complex linear systems. The convergence theory and the spectral properties of the BAS iteration method are discussed. Numerical experiments are presented to illustrate the efficiency of the BAS iteration as a solver as well as a preconditioner for Krylov subspace methods.

## 1. Introduction

We consider the distributed control problem [1,2] of the form: find the state $y(x,t)$ and the control $u(x,t)$ such that it minimizes the cost functional

$$J(y,u)=\frac{1}{2}\int_0^T\int_{\Omega}|y(x,t)-y_d(x,t)|^2dxdt+\frac{\upsilon}{2}\int_0^T\int_{\Omega}|u(x,t)|^2dxdt \tag{1.1}$$

subject to the time-periodic parabolic equation

$$\frac{\partial}{\partial t}y(x,t)-\Delta y(x,t)=u(x,t)\quad \text{in } Q_T, \tag{1.2}$$

$$y(x,t)=0\quad \text{on } \Sigma_T, \tag{1.3}$$

$$y(x,0)=y(x,T)\quad \text{on } \partial\Omega, \tag{1.4}$$

$$u(x,0)=u(x,T)\quad \text{in } \Omega, \tag{1.5}$$

[☆] This work was supported by the National Natural Science Foundation of China (Nos. 11271174, 1141101097).

[*] Corresponding author.
*E-mail address:* gf_zhang@lzu.edu.cn (G.-F. Zhang).

http://dx.doi.org/10.1016/j.cam.2015.04.005
0377-0427/© 2015 Elsevier B.V. All rights reserved.

where $\Omega$ is an open and bounded domain in $\mathbb{R}^d$ for $d \in \{1, 2, 3\}$ with Lipschitz-continuous boundary $\partial\Omega$. Here we denote by $\partial\Omega$ the boundary of the domain $\Omega$. We introduce the space–time cylinder $Q_T = \Omega \times (0, T)$ and its lateral surface $\Sigma_T = \partial Q \times (0, T)$. $T > 0$ and $y_d(x, t)$ is a given target (or desired) state and $\upsilon > 0$ is a cost or regularization parameter.

Such optimization problems with partial differential equations as constraints arise widely in many areas of science and engineering [3,4,1,2]. Moreover, in many practical applications, for example, in electromagnetics, we can assume that $y_d(x, t)$ is time-harmonic [3,5], i.e.,

$$y_d(x,t) = y_d(x)e^{iwt} \quad \text{with } w = \frac{2\pi m}{T} \text{ for some } m \in \mathbb{Z}.$$

Then there is a time-periodic solution to the original control problem of the form

$$y(x,t) = y(x)e^{iwt}, \qquad u(x,t) = u(x)e^{iwt},$$

where $y(x)$ and $u(x)$ are the solutions of the following time-independent optimal problem: Minimize

$$\frac{1}{2}\int_{\Omega} |y(x) - y_d(x)|^2 dx + \frac{\upsilon}{2}\int_{\Omega} |u(x)|^2 dx \tag{1.6}$$

subject to

$$iwy(x) - \Delta y(x) = u(x) \quad \text{in } \Omega, \tag{1.7}$$

$$y(x) = 0 \quad \text{on } \partial\Omega. \tag{1.8}$$

Of course, we cannot solve the PDE-constraint in (1.1)–(1.5) analytically (in general). So we must discretize the constraint and seek a finite-dimensional solution. Consequently, by discretize-then-optimization [6–8], the optimization problem (1.6)–(1.8) is reformulated as follows:

Minimize

$$\frac{1}{2}(\tilde{y} - \tilde{y}_d)^* M (\tilde{y} - \tilde{y}_d) + \frac{\upsilon}{2}\tilde{u}^* M \tilde{u} \tag{1.9}$$

subject to

$$iwM\tilde{y} + K\tilde{y} = M\tilde{u}. \tag{1.10}$$

Here the real matrices $M$ and $K$ are the mass matrix, representing the $L^2$-inner product in $V_h$, and the discretized negative Laplacian, respectively. $\tilde{y}$, $\tilde{y}_d$, and $\tilde{u}$ denote the coefficient vectors of the corresponding finite element functions in $V_h$. Here and in the sequel, $u^*$ is the conjugate transpose of the vector $u$.

The Lagrangian functional for this constrained optimization problem (1.9)–(1.10) is given by

$$\mathscr{L}(\tilde{y}, \tilde{u}, \tilde{p}) = \frac{1}{2}(\tilde{y} - \tilde{y}_d)^* M (\tilde{y} - \tilde{y}_d) + \frac{\upsilon}{2}\tilde{u}^* M \tilde{u} + \tilde{p}^*(iwM\tilde{y} + K\tilde{y} - M\tilde{u}),$$

where $\tilde{p}$ denotes the Lagrangian multiplier associated with the constraint. Then it leads to a system matrix which takes the form of the following complex linear system

$$\begin{pmatrix} M & 0 & K - iwM \\ 0 & \upsilon M & -M \\ K + iwM & -M & 0 \end{pmatrix} \begin{pmatrix} \tilde{y} \\ \tilde{u} \\ \tilde{p} \end{pmatrix} = \begin{pmatrix} M\tilde{y}_d \\ 0 \\ 0 \end{pmatrix}. \tag{1.11}$$

Obviously, the linear system (1.11) is of the form of the saddle point problems. For solving this linear system efficiently, Bai in [9] proposed a block alternating splitting implicit iteration method. Bai and Golub in [10] introduced the accelerated Hermitian and skew-Hermitian splitting iteration method. Jiang and Cao in [11] presented the local HSS (LHSS) and modified LHSS iteration methods. In [12], Krukier et al. gave the generalized skew-Hermitian triangular splitting (GSTS) iteration method and Pan et al. in [13] proposed a preconditioned-GMRES method for this class of problems. For more details, we refer to [14–16].

However, (1.11) can also be equivalently rewritten into a block element form as

$$\begin{cases} M\tilde{y} + (K - iwM)\tilde{p} = M\tilde{y}_d, \\ \upsilon M\tilde{u} - M\tilde{p} = 0, \\ (K + iwM)\tilde{y} - M\tilde{u} = 0. \end{cases} \tag{1.12}$$

From the second equation in (1.12), we easily get $\tilde{u} = \frac{1}{\upsilon}\tilde{p}$. By substituting this equation with respect to $\tilde{u}$ into the other two equations in (1.12), we obtain the following complex linear system:

$$\begin{cases} M\tilde{y} + (K - iwM)\tilde{p} = M\tilde{y}_d, \\ (K + iwM)\tilde{y} - \dfrac{1}{\upsilon}M\tilde{p} = 0, \end{cases}$$

or equivalently,

$$\mathscr{A}x \equiv \begin{pmatrix} M & \sqrt{v}(K - iwM) \\ \sqrt{v}(K + iwM) & -M \end{pmatrix} \begin{pmatrix} \tilde{y} \\ \tilde{q} \end{pmatrix} = \begin{pmatrix} M\tilde{y}_d \\ 0 \end{pmatrix} \equiv b, \tag{1.13}$$

where $\tilde{q} = \frac{1}{\sqrt{v}}\tilde{p}$, $M, K \in \mathbb{R}^{n\times n}$ are both symmetric and positive definite matrices, $v > 0$ is a cost or regularization parameter, $w = 2\pi m/T$ for some $m \in \mathbb{Z}$, $\tilde{y}$ and $\tilde{q}$ denote the coefficient vectors of the corresponding finite element functions relative to the chosen set of basis functions in $V_h$ and a scaling of Lagrangian multiplier, respectively. For more details about this problem, we refer to [5,17–19].

In this paper, we focus on the solution of the complex linear equations (1.13). This type of complex linear equations arises in numerous applications, ranging from optimization to the solution of PDEs to the other areas. For solving this complex linear system, it is known that there are two options: either tackling the linear system directly or working with one of the several equivalent real formulations to avoid solving the complex linear system. For the later, the new coefficient matrix is often doubled in size. Working with the original linear system requires using complex arithmetic.

To solve the linear system (1.13), we can adopt some Krylov subspace methods, such as GMRES($\sharp$) [20]. The linear system of equations (1.13) can also be regarded as a special case of the generalized saddle-point problem, and it has recently attracted a lot of attention. As (1.13) is a large, sparse complex linear system, frequently, iterative methods are more attractive for solving the linear system. Many efficient iterative algorithms have been proposed and studied in many papers [21–24,9,20]. For example, HSS methods [23,25,26], MHSS method [27], PMHSS method [28,29] and so on, see [18] for a comprehensive survey.

Since the coefficient matrix in (1.13) possesses special block and sparse structure, to compute an approximate solution of the system of the complex linear equations fast and accurately, Krendl [2] proposed the real block diagonal preconditioner $P_{BD}$ and the alternative indefinite preconditioner $P_{AI}$ as follows:

$$P_{BD} = \begin{pmatrix} M + \sqrt{v}(K + wM) & 0 \\ 0 & M + \sqrt{v}(K + wM) \end{pmatrix}, \tag{1.14}$$

$$P_{AI} = \begin{pmatrix} 0 & M + \sqrt{v}(K - iwM) \\ M + \sqrt{v}(K + iwM) & -M \end{pmatrix}. \tag{1.15}$$

It is well known that using complex arithmetic throughout the code may be wasteful in practice. For the complex linear system of the form

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

with $A = W + \mathbf{i}T$ being complex symmetric matrix and $W, T \in \mathbb{R}^{n\times n}$ real symmetric and positive definite matrices [30,28], Bai et al. in [28] established a PMHSS iteration method:

$$\begin{cases} (\alpha V + W)u^{\left(k+\frac{1}{2}\right)} = (\alpha V - iT)u^{(k)} + b, \\ (\alpha V + T)u^{(k+1)} = (\alpha V + iW)u^{\left(k+\frac{1}{2}\right)} - ib, \end{cases}$$

where $\alpha > 0$ is a given positive constant and $V \in \mathbb{R}^{n\times n}$ is a prescribed symmetric positive definite matrix. Particularly, if $V = W$, they proved that the PMHSS iteration method is unconditionally convergent and have shown that both the stationary PMHSS iteration and PMHSS-preconditioned GMRES method have mesh size-independent convergence behavior for their tested numerical examples. However, noting that the real part of this complex linear system (1.13) is not positive definite, the PMHSS iteration method [28] cannot be used directly to solve the problem.

In this paper, by utilizing the block structure of the coefficient matrix of the complex linear system (1.13), we split the matrix $\mathscr{A}$ by a new technique and introduce a new block alternating splitting (BAS) iteration method for solving the system of complex linear equations (1.13).

This paper is organized as follows. In Section 2, we present the BAS iteration method and the corresponding preconditioner for solving the block two-by-two complex linear system (1.13). Then, in Section 3, we discuss the convergence of the BAS iteration method and the eigenproperties of the preconditioned matrix. In Section 4, numerical experiments are given to confirm and illustrate the theoretical results and the effectiveness of the BAS iteration as a solver as well as a preconditioner. Concluding remarks are given in Section 5.

## 2. The block alternating splitting (BAS) iteration method

Firstly, we split the coefficient matrix $\mathscr{A}$ of (1.13) into the sum of two matrices as follows:

$$\mathscr{A} = \begin{pmatrix} M & -i\sqrt{v}wM \\ i\sqrt{v}wM & -M \end{pmatrix} + \begin{pmatrix} 0 & \sqrt{v}K \\ \sqrt{v}K & 0 \end{pmatrix}. \tag{2.1}$$

Define

$$P_1 := \frac{1}{1+w^2v}\begin{pmatrix} I & -iw\sqrt{v}I \\ iw\sqrt{v}I & -I \end{pmatrix}, \qquad P_2 := \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}, \tag{2.2}$$

$I$ is an identity matrix with proper sizes here and in the subsequent discussions.

By left-multiplying $P_1$ for the complex linear system (1.13), i.e., $P_1\mathscr{A}x = P_1b$, we obtain

$$\left(\begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} + \begin{pmatrix} -\frac{iwv}{1+w^2v}K & \frac{\sqrt{v}}{1+w^2v}K \\ -\frac{\sqrt{v}}{1+w^2v}K & \frac{iwv}{1+w^2v}K \end{pmatrix}\right) x = \begin{pmatrix} \frac{1}{1+w^2v}M\tilde{y}_d \\ \frac{iw\sqrt{v}}{1+w^2v}M\tilde{y}_d \end{pmatrix}, \tag{2.3}$$

or

$$(H_1 + S_1)x = P_1b,$$

where

$$H_1 := \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix}, \qquad S_1 := \begin{pmatrix} -\frac{iwv}{1+w^2v}K & \frac{\sqrt{v}}{1+w^2v}K \\ -\frac{\sqrt{v}}{1+w^2v}K & \frac{iwv}{1+w^2v}K \end{pmatrix}. \tag{2.4}$$

Based on the matrix splitting (2.3), we can construct a first half-step iterative formation:

$$(\alpha V + H_1)x^{\left(k+\frac{1}{2}\right)} = (\alpha V - S_1)x^{(k)} + P_1b. \tag{2.5}$$

Similarly, by left multiplying $P_2$ for the linear system (1.13), we have $P_2\mathscr{A}x = P_2b$, which is equivalent to

$$\left(\begin{pmatrix} \sqrt{v}K & 0 \\ 0 & \sqrt{v}K \end{pmatrix} + \begin{pmatrix} i\sqrt{v}wM & -M \\ M & -i\sqrt{v}wM \end{pmatrix}\right) x = \begin{pmatrix} 0 \\ M\tilde{y}_d \end{pmatrix}. \tag{2.6}$$

We define the matrices $H_2$ and $S_2$ as follows:

$$H_2 := \begin{pmatrix} \sqrt{v}K & 0 \\ 0 & \sqrt{v}K \end{pmatrix}, \qquad S_2 := \begin{pmatrix} i\sqrt{v}wM & -M \\ M & -i\sqrt{v}wM \end{pmatrix}. \tag{2.7}$$

Then we can introduce our second half-step iterative formation:

$$(\alpha V + H_2)x^{(k+1)} = (\alpha V - S_2)x^{\left(k+\frac{1}{2}\right)} + P_2b. \tag{2.8}$$

Alternating between above two iterations (2.5) and (2.8) leads to the block alternating splitting (**BAS**) iteration method for solving the block two-by-two complex linear system (1.13), which is described as follows.

**Method 2.1** (*The BAS Iteration Method*). *Let $x^{(0)} \in \mathbb{R}^{2n}$ be an arbitrary initial guess. For $k = 0, 1, 2, \ldots$ until the sequence of iterations $\{x^{(k)}\}_{k=0}^{\infty} \subset \mathbb{R}^{2n}$ converges, compute the next iterate $x^{(k+1)}$ according to the following procedure:*

$$\begin{cases} (\alpha V + H_1)x^{\left(k+\frac{1}{2}\right)} = (\alpha V - S_1)x^{(k)} + P_1b, \\ (\alpha V + H_2)x^{(k+1)} = (\alpha V - S_2)x^{\left(k+\frac{1}{2}\right)} + P_2b, \end{cases} \tag{2.9}$$

*where $\alpha > 0$ is an iteration parameter and $V \in \mathbb{R}^{2n\times 2n}$ is a prescribed symmetric and positive definite matrix. $H_1$, $S_1$, $H_2$, $S_2$, $P_1$, $P_2$ are defined as in (2.4), (2.7) and (2.2), respectively.*

For the BAS iteration method, the coefficient matrices $(\alpha V + H_1)$ and $(\alpha V + H_2)$ of the two subsystems of the iteration (2.9) possess good structure. In particular, if the matrix blocks $V$ has some simple structure or special property, for example, $V = \mathrm{blkdiag}(M, M)$, then the two sub-systems of the iteration may be solved more easily and effectively than the original linear system (1.13).

Eliminating $x^{(k+\frac{1}{2})}$ from (2.9), we can rewrite the BAS iteration method as a stationary scheme:

$$x^{(k+1)} = L(V;\alpha)x^{(k)} + R(V;\alpha)b, \quad k = 0, 1, 2, \ldots,$$

where $L(V;\alpha)$ is the iteration matrix of the BAS method with

$$L(V;\alpha) = (\alpha V + H_2)^{-1}(\alpha V - S_2)(\alpha V + H_1)^{-1}(\alpha V - S_1) \tag{2.10}$$

and

$$R(V;\alpha) = (\alpha V + H_2)^{-1}[(\alpha V - S_2) + P_2 P_1^{-1}(\alpha V + H_1)](\alpha V + H_1)^{-1}P_1.$$

As in [27,18], one can show that there is a unique splitting $\mathscr{A} = P_{\mathrm{BAS}}(\alpha) - G_{\mathrm{BAS}}(\alpha)$ with $P_{\mathrm{BAS}}(\alpha)$ nonsingular such that the iteration matrix $L(V;\alpha)$ is the matrix induced by that splitting, i.e., $L(V;\alpha) = P_{\mathrm{BAS}}(\alpha)^{-1}G_{\mathrm{BAS}}(\alpha) = I - P_{\mathrm{BAS}}(\alpha)^{-1}\mathscr{A}$.

If we choose $V = \mathrm{blkdiag}(M, M)$, then the matrices $P_{\mathrm{BAS}}(\alpha)$ and $G_{\mathrm{BAS}}(\alpha)$ are given by

$$P_{\mathrm{BAS}}(\alpha) = (\alpha+1)P(\alpha)\begin{pmatrix} \alpha M + \sqrt{v}K & 0 \\ 0 & \alpha M + \sqrt{v}K \end{pmatrix} \tag{2.11}$$

and

$$G_{\mathrm{BAS}}(\alpha) = P(\alpha)\begin{pmatrix} \alpha I - iw\sqrt{v}I & I \\ -I & \alpha I + iw\sqrt{v}I \end{pmatrix}\begin{pmatrix} \alpha M + \dfrac{iwv}{1+w^2v}K & -\dfrac{\sqrt{v}}{1+w^2v}K \\ \dfrac{\sqrt{v}}{1+w^2v}K & \alpha M - \dfrac{iwv}{1+w^2v}K \end{pmatrix},$$

with

$$P(\alpha) = \frac{1}{\alpha(2+w^2v)}\begin{pmatrix} I & (1+w^2v - iw\sqrt{v})I \\ (1+w^2v+iw\sqrt{v})I & -I \end{pmatrix}. \tag{2.12}$$

We refer to the iteration (2.9) as the BAS iteration method, and $P_{\mathrm{BAS}}(\alpha)$ as the BAS-preconditioner. Actually, when we choose $\alpha = 1 + w\sqrt{v}$, a part of $P_{\mathrm{BAS}}(\alpha)$, i.e., $\mathrm{blkdiag}(\alpha M + \sqrt{v}K, \alpha M + \sqrt{v}K)$ reduces to $P_{\mathrm{BD}}$ defined as in (1.14). Hence, the preconditioner $P_{\mathrm{BAS}}(\alpha)$ can be seen as a generalization of the preconditioner $P_{\mathrm{BD}}(\alpha)$.

## 3. Convergence of the BAS iteration method and the eigenproperties of preconditioning

In this section, we will prove that, under suitable conditions, the BAS iteration method (2.9) converges to the solution of (1.13) for any initial guesses. Moreover, an upper bound on the spectral radius is given. These facts are precisely stated in the following theorems.

**Theorem 3.1.** *Let $\mathscr{A} \in \mathbb{C}^{2n\times 2n}$ be the block two-by-two matrix defined as in (1.13), $V = \mathrm{blkdiag}(\tilde{V}, \tilde{V}) \in \mathbb{R}^{2n\times 2n}$ with $\tilde{V} \in \mathbb{R}^{n\times n}$ being a symmetric positive definite matrix. Denote the largest and smallest eigenvalues of $\tilde{V}^{-1}M$ as $\tilde{\lambda}_{\max}$, $\tilde{\lambda}_{\min}$, respectively, and the smallest and the largest eigenvalues of $\tilde{V}^{-1}K$ as $\tilde{\mu}_{\min}$ and $\tilde{\mu}_{\max}$, respectively. Assume that $\alpha$ satisfies*

$$\alpha \geq \max\left\{\frac{(k-1)\tilde{\lambda}_{\max}^2 - 2\tilde{\lambda}_{\min}\sqrt{v}\tilde{\mu}_{\min}}{2(\tilde{\lambda}_{\min} + \sqrt{v}\tilde{\mu}_{\min})}, 0\right\}.$$

*Then it holds that*

$$\rho(L(V,\alpha)) \leq \tilde{\sigma}(\alpha) < 1,$$

*with*

$$\tilde{\sigma}(\alpha) := \max_{\tilde{\lambda}_j \in \sigma(\tilde{V}^{-1}M)} \frac{\sqrt{\alpha^2 + k\tilde{\lambda}_j^2}}{\alpha + \tilde{\lambda}_j} \cdot \max_{\tilde{\mu}_j \in \sigma(\sqrt{v}\tilde{V}^{-1}K)} \frac{\sqrt{\alpha^2 + \frac{1}{k}\tilde{\mu}_j^2}}{\alpha + \tilde{\mu}_j}, \tag{3.1}$$

*where $\sigma(\cdot)$ and $\rho(\cdot)$ denote the spectral set and the spectral radius of the corresponding matrix, $k := 1 + w^2v$.*

**Proof.** By introducing the notation $\tilde{M} := \tilde{V}^{-\frac{1}{2}}M\tilde{V}^{-\frac{1}{2}}$ and $\tilde{K} := \sqrt{v}\tilde{V}^{-\frac{1}{2}}K\tilde{V}^{-\frac{1}{2}}$. We know that there exist two orthogonal matrices $\tilde{Q}_M, \tilde{Q}_K \in \mathbb{R}^{n\times n}$, two positive diagonal matrices $\Sigma = \mathrm{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_n) \in \mathbb{R}^{n\times n}$ and $\Lambda = \mathrm{diag}(\tilde{\mu}_1, \tilde{\mu}_2, \dots, \tilde{\mu}_n) \in \mathbb{R}^{n\times n}$ such that

$$\tilde{M} = \tilde{Q}_M \Sigma \tilde{Q}_M^T \quad \text{and} \quad \tilde{K} = \tilde{Q}_K \Lambda \tilde{Q}_K^T.$$

By block scaling and matrix similarity, we know that the iteration matrix $L(V,\alpha)$ of the BAS method is similar to the matrix $\tilde{L}(V;\alpha)$, i.e.,

$$\begin{aligned} L(V;\alpha) \sim \tilde{L}(V;\alpha) &= (\alpha V - S_2)(\alpha V + H_1)^{-1}(\alpha V - S_1)(\alpha V + H_2)^{-1} \\ &= \begin{pmatrix} V^{-\frac{1}{2}} & \\ & V^{-\frac{1}{2}} \end{pmatrix} \tilde{L}(\alpha) \begin{pmatrix} V^{\frac{1}{2}} & \\ & V^{\frac{1}{2}} \end{pmatrix}, \end{aligned}$$

with

$$\begin{aligned}\tilde{L}(\alpha) = \frac{1}{k}&\begin{pmatrix}\alpha I - iw\sqrt{v}\tilde{M} & \tilde{M}\\ -\tilde{M} & \alpha I + iw\sqrt{v}\tilde{M}\end{pmatrix}\begin{pmatrix}\alpha I + \tilde{M} & 0\\ 0 & \alpha I + \tilde{M}\end{pmatrix}^{-1}\\ &* \begin{pmatrix}k\alpha I + iwv\tilde{K} & -\sqrt{v}\tilde{K}\\ \sqrt{v}\tilde{K} & k\alpha I - iwv\tilde{K}\end{pmatrix}\begin{pmatrix}\alpha I + \sqrt{v}\tilde{K} & 0\\ 0 & \alpha I + \sqrt{v}\tilde{K}\end{pmatrix}^{-1}.\end{aligned}$$

Note that $\tilde{L}(V;\alpha)$ is similar to the matrix $\tilde{L}(\alpha)$. Therefore, it holds that

$$\rho(L(V;\alpha)) = \rho(\tilde{L}(\alpha)).$$

By straightforward computations it follows that

$$\begin{aligned}\rho(\tilde{L}(\alpha)) &\le \frac{1}{k}\left\|\begin{pmatrix}\alpha I - iw\sqrt{v}\tilde{M} & \tilde{M}\\ -\tilde{M} & \alpha I + iw\sqrt{v}\tilde{M}\end{pmatrix}\begin{pmatrix}\alpha I + \tilde{M} & 0\\ 0 & \alpha I + \tilde{M}\end{pmatrix}^{-1}\right\|_2\\ &\quad\cdot\left\|\begin{pmatrix}k\alpha I + iwv\tilde{K} & -\sqrt{v}\tilde{K}\\ \sqrt{v}\tilde{K} & k\alpha I - iwv\tilde{K}\end{pmatrix}\begin{pmatrix}\alpha I + \sqrt{v}\tilde{K} & 0\\ 0 & \alpha I + \sqrt{v}\tilde{K}\end{pmatrix}^{-1}\right\|_2\\ &= \frac{1}{k}\left\|\begin{pmatrix}\alpha I - iw\sqrt{v}\Sigma & \Sigma\\ -\Sigma & \alpha I + iw\sqrt{v}\Sigma\end{pmatrix}\begin{pmatrix}\alpha I + \Sigma & 0\\ 0 & \alpha I + \Sigma\end{pmatrix}^{-1}\right\|_2\\ &\quad\cdot\left\|\begin{pmatrix}k\alpha I + iwv\Lambda & -\sqrt{v}\Lambda\\ \sqrt{v}\Lambda & k\alpha I - iwv\Lambda\end{pmatrix}\begin{pmatrix}\alpha I + \sqrt{v}\Lambda & 0\\ 0 & \alpha I + \sqrt{v}\Lambda\end{pmatrix}^{-1}\right\|_2\\ &= \max_{\tilde{\lambda}_j\in\sigma(\tilde{M})}\left|\frac{\alpha \pm i\sqrt{k}\tilde{\lambda}_j}{\alpha + \tilde{\lambda}_j}\right| \cdot \max_{\tilde{\mu}_j\in\sigma(\sqrt{v}\tilde{K})}\left|\frac{\alpha \pm i\frac{1}{\sqrt{k}}\tilde{\mu}_j}{\alpha + \tilde{\mu}_j}\right|\\ &= \max_{\tilde{\lambda}_j\in\sigma(\tilde{M})}\frac{\sqrt{\alpha^2 + k\tilde{\lambda}_j^2}}{\alpha + \tilde{\lambda}_j} \cdot \max_{\tilde{\mu}_j\in\sigma(\sqrt{v}\tilde{K})}\frac{\sqrt{\alpha^2 + \frac{1}{k}\tilde{\mu}_j^2}}{\alpha + \tilde{\mu}_j} \equiv \tilde{\sigma}(\alpha).\end{aligned}$$

Let $\tilde{\sigma}(\alpha) < 1$. We have the following inequality

$$\frac{\alpha^2 + k\tilde{\lambda}^2}{(\alpha + \tilde{\lambda})^2} \cdot \frac{\alpha^2 + \frac{1}{k}\tilde{\mu}^2}{(\alpha + \tilde{\mu})^2} < 1,$$

or equivalently,

$$(\alpha + \tilde{\lambda})^2(\alpha + \tilde{\mu})^2 - (\alpha^2 + k\tilde{\lambda}^2)\left(\alpha^2 + \frac{1}{k}\tilde{\mu}^2\right) > 0,$$

which directly leads to

$$2\alpha^2(\tilde{\lambda} + \tilde{\mu}) - \alpha\left(\frac{1}{k}(k\tilde{\lambda} - \tilde{\mu})^2 - (\tilde{\lambda} + \tilde{\mu})^2\right) + 2\tilde{\lambda}\tilde{\mu}(\tilde{\lambda} + \tilde{\mu}) > 0. \tag{3.2}$$

(3.2) is a quadratic inequality about iteration parameter $\alpha$. Note that the discriminant is

$$\begin{aligned}\Delta &= (\tilde{\lambda} + \tilde{\mu})^4 + \frac{1}{k^2}(k\tilde{\lambda} - \tilde{\mu})^4 - \frac{2}{k^2}(\tilde{\lambda} + \tilde{\mu})^2(k\tilde{\lambda}^2 - \tilde{\mu})^2 - 16\tilde{\lambda}\tilde{\mu}(\tilde{\lambda} + \tilde{\mu})^2\\ &= \left((k-1)\tilde{\lambda}^2 + \frac{1}{k}(k-1)\tilde{\mu}^2\right)^2 - 8\left(k + \frac{3}{2}\right)\tilde{\lambda}^3\tilde{\mu} - 4\left(k + \frac{1}{k} + 4\right)\tilde{\lambda}^2\tilde{\mu}^2 - 8\left(\frac{1}{k} + \frac{3}{2}\right)\tilde{\lambda}\tilde{\mu}^3\\ &< \left((k-1)\tilde{\lambda}^2 + \frac{1}{k}(k-1)\tilde{\mu}^2\right)^2, \quad \text{with } \tilde{\lambda}, \tilde{\mu} > 0,\ k > 1.\end{aligned}$$

By the quadratic formula of a quadratic equation, a sufficient condition for the convergence of the BAS iteration method (2.9) is given by the following inequality

$$\alpha > \frac{\frac{1}{k}(k\tilde{\lambda} - \tilde{\mu})^2 - (\tilde{\lambda} + \tilde{\mu})^2 + \sqrt{\Delta}}{4(\tilde{\lambda} + \tilde{\mu})},$$

or equivalently,

$$\alpha \geq \frac{\frac{1}{k}(k\tilde{\lambda}-\tilde{\mu})^2-(\tilde{\lambda}+\tilde{\mu})^2+(k-1)\tilde{\lambda}^2+\frac{1}{k}(k-1)\tilde{\mu}^2}{4(\tilde{\lambda}+\tilde{\mu})}=\frac{(k-1)\tilde{\lambda}^2-2\tilde{\lambda}\tilde{\mu}}{2(\tilde{\lambda}+\tilde{\mu})}.$$

Thus it then follows that if $\alpha$ satisfies the condition:

$$\alpha \geq \max\left\{\frac{(k-1)\tilde{\lambda}^2_{\max}-2\tilde{\lambda}_{\min}\tilde{\mu}_{\min}}{2(\tilde{\lambda}_{\min}+\tilde{\mu}_{\min})},\ 0\right\}. \tag{3.3}$$

Then the BAS iteration method converges to the unique solution of the block two-by-two complex linear system (1.13). □

In Theorem 3.1, by minimizing an upper bound $\sigma(\alpha)$ of the spectral radius of the iteration matrix we obtain a sufficient condition for guaranteeing the convergence for the BAS method. Next we consider the convergence condition of the BAS method for a special case with $V = \text{blkdiag}(M, M) \in \mathbb{R}^{2n\times 2n}$.

**Theorem 3.2.** *Assume that the conditions of Theorem 3.1 are satisfied and $V = blkdiag(M, M) \in \mathbb{R}^{2n\times 2n}$. Then the spectral radius of the iteration matrix $L(V; \alpha)$ of the BAS method satisfies*

$$\rho(L(V,\ \alpha)) \leq \overline{\sigma}(\alpha)$$

*with*

$$\overline{\sigma}(\alpha) \equiv \frac{\sqrt{\alpha^2+1+w^2v}}{\alpha+1}. \tag{3.4}$$

*Moreover, the BAS iteration method converges to the unique solution of the block two-by-two complex linear systems (1.13) for any initial guess, provided*

$$\alpha \geq \frac{w^2v}{2}.$$

**Proof.** We can easily know that

$$\max_{\tilde{\mu}_j\in\sigma(\sqrt{v}M^{-1}K)} \frac{\sqrt{\alpha^2+\frac{1}{k}\tilde{\mu}_j^2}}{\alpha+\tilde{\mu}_j} < 1. \tag{3.5}$$

Coupled with the assumption $V = \text{blkdiag}(M, M)$, it follows that

$$\overline{\sigma}(\alpha) = \max_{\tilde{\lambda}_j\in\sigma(M^{-1}M)} \frac{\sqrt{\alpha^2+(1+w^2v)\tilde{\lambda}_j^2}}{\alpha+\tilde{\lambda}_j} = \frac{\sqrt{\alpha^2+(1+w^2v)}}{\alpha+1}. \tag{3.6}$$

It is well known that the BAS method converges if and only if $\rho(\tilde{L}(\alpha)) < 1$. Then from

$$\frac{\sqrt{\alpha^2+1+w^2v}}{\alpha+1} \leq 1,$$

we get $\alpha \geq \frac{1}{2}w^2v$.

**Remark 1.** If we choose $V = \text{blkdiag}(M, M)$, which means $\Sigma$ is an identity matrix, i.e., $\tilde{\lambda}_j = 1, j = 1, 2, \ldots, n$. In this case, a sufficient condition for the convergence of the BAS iteration method (2.9) is given by the following inequality

$$\alpha \geq \max\left\{\frac{w^2v-2\tilde{\mu}_{\min}}{2(1+\tilde{\mu}_{\min})},\ 0\right\}.$$

**Remark 2.** Assume that $V = \text{blkdiag}(M, M)$. Then, when $v \to 0$, $\tilde{\mu}_{\min} \to 0$ is the minimum eigenvalue of the matrix $\sqrt{v}\tilde{K}$. Thus we can know that when $\alpha \geq \frac{1}{2}w^2v$, the BAS method is convergent. In this case, the convergent condition is independent on the eigenvalues of $M^{-1}K$. $\alpha = \frac{1}{2}w^2v$(as $v \to 0$) is an inexact lower bound of $\alpha$.

**Remark 3.** From Theorem 3.2 we see that when $\alpha = 1 + w^2v$, $\overline{\sigma}(\alpha)$ reaches its minimum value $\sqrt{\frac{1+w^2v}{2+w^2v}}$. Then we can take $\alpha = 1 + w^2v$ as a choice of the parameter $\alpha$, which we list in Table 2 in Section 4.

The following theorem describes the spectral properties of the preconditioned matrix $P_{\text{BAS}}(\alpha)^{-1}\mathscr{A}$.

**Theorem 3.3.** *Let $\mathscr{A} \in \mathbb{R}^{2n\times 2n}$ be the block two-by-two matrix defined as in (1.13), with $M, K \in \mathbb{R}^{n\times n}$ being symmetric positive definite matrices, $\alpha > 0$ be a positive constant, $v > 0$, and $w = 2\pi m/T$ for some $m \in \mathbb{Z}$. Denote by $\Lambda = diag(\mu_1, \mu_2, \ldots, \mu_n)$ the eigenvalue matrix of the symmetric matrix $M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$. Then the eigenvalues of the BAS preconditioned matrix $P_{BAS}(\alpha)^{-1}\mathscr{A}$ are*

$$\lambda_{\pm}^{(j)} = \frac{\alpha\sqrt{1+w^2v}(2+w^2v)}{\alpha+1}\left(\sqrt{1+w^2v}(1-\sqrt{v}\mu_j) \pm i(1+w^2v-\sqrt{v}\mu_j)\right)(\alpha+\sqrt{v}\mu_j)^{-1}, \quad j = 1, \ldots, n$$

*and the Euclidean norm of the matrix of the corresponding eigenvectors is* 1.

**Proof.** Let $U \in \mathbb{R}^{n\times n}$ be an orthogonal matrix such that $M^{-\frac{1}{2}}KM^{-\frac{1}{2}} = U^T\Lambda U$ with $\Lambda = \operatorname{diag}(\mu_1, \mu_2, \ldots, \mu_n)$. Because $K$ and $M$ are symmetric positive definite matrices, then $\mu_j > 0$ $(j = 1, 2, \ldots, n)$. Hence

$$P_{BAS}^{-1}(\alpha)\mathscr{A} = \frac{1}{\alpha+1}\begin{pmatrix} \alpha M + \sqrt{v}K & 0 \\ 0 & \alpha M + \sqrt{v}K \end{pmatrix}^{-1} P^{-1}(\alpha)\mathscr{A}$$

with $P(\alpha)$ being defined as in (2.12) and

$$P^{-1}(\alpha)\mathscr{A} = \alpha(1+k)\begin{pmatrix} k(1+i\sqrt{w^2v})M + \sqrt{v}(k-i\sqrt{w^2v})K & -kM+\sqrt{v}K \\ kM-\sqrt{v}K & k(1-i\sqrt{w^2v})M + \sqrt{v}(k+i\sqrt{w^2v})K \end{pmatrix},$$

where $k := 1 + w^2v$. Then we have

$$P_{BAS}^{-1}(\alpha)\mathscr{A} = \frac{\alpha(1+k)}{\alpha+1}\begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix}^T S(\alpha)\begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix},$$

where

$$\begin{aligned} S(\alpha) &= \begin{pmatrix} \alpha I + \sqrt{v}\Lambda & 0 \\ 0 & \alpha I + \sqrt{v}\Lambda \end{pmatrix}^{-1}\begin{pmatrix} r_1 + i\sqrt{w^2v}(kI-\sqrt{v}\Lambda) & -r_2 \\ kI - \sqrt{v}\Lambda & k(I+\sqrt{v}\Lambda) - i\sqrt{w^2v}(kI-\sqrt{v}\Lambda) \end{pmatrix} \\ &= \begin{pmatrix} \alpha I + \sqrt{v}\Lambda & 0 \\ 0 & \alpha I + \sqrt{v}\Lambda \end{pmatrix}^{-1}\begin{pmatrix} r_1 + i(\sqrt{w^2v})r_2 & -r_2 \\ r_2 & r_1 - i(\sqrt{w^2v})r_2 \end{pmatrix} \\ &= Q^*\begin{pmatrix} (\alpha I + \sqrt{v}\Lambda)^{-1}\Sigma_+ & 0 \\ 0 & (\alpha I + \sqrt{v}\Lambda)^{-1}\Sigma_- \end{pmatrix} Q \end{aligned}$$

with

$$\begin{aligned} &r_1 := k(I+\sqrt{v}\Lambda), \qquad r_2 := kI - \sqrt{v}\Lambda, \qquad \Sigma_{\pm} := r_1 \pm i\sqrt{k}r_2, \qquad Q := [q_+, \quad q_-], \\ &q_{\pm} := \frac{1}{\chi_{\pm}}(i(\sqrt{k-1} \pm \sqrt{k})\,\mathrm{I},\ \mathrm{I})^T, \qquad \chi_{\pm}^2 := 2\sqrt{k}(\sqrt{k} \pm \sqrt{k-1}). \end{aligned}$$

Obviously, $Q \in \mathbb{C}^{2n\times 2n}$ and $Q^*Q = QQ^* = I$. Thus,

$$\begin{aligned} P_{BAS}^{-1}(\alpha)\mathscr{A} &= \begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix}^T Q^*\begin{pmatrix} \frac{\alpha(1+k)}{\alpha+1}(\alpha I + \sqrt{v}\Lambda)^{-1}\Sigma_+ & 0 \\ 0 & \frac{\alpha(1+k)}{\alpha+1}(\alpha I + \sqrt{v}\Lambda)^{-1}\Sigma_- \end{pmatrix} Q\begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix} \\ &= \tilde{Q}^*\begin{pmatrix} \Xi_+ & 0 \\ 0 & \Xi_- \end{pmatrix}\tilde{Q}, \end{aligned}$$

where $\Xi_{\pm} = \operatorname{diag}(\lambda_{\pm}^{(1)}, \lambda_{\pm}^{(2)}, \ldots, \lambda_{\pm}^{(n)})$ with

$$\lambda_{\pm}^{(j)} = \frac{\alpha\sqrt{k}(1+k)}{\alpha+1}[\sqrt{k}(1-\sqrt{\upsilon}\mu_j) \pm i(k-\sqrt{\upsilon}\mu_j)](\alpha+\sqrt{\upsilon}\mu_j)^{-1}, \quad j = 1, \ldots, n.$$

Moreover, $U \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $Q \in \mathbb{C}^{2n\times 2n}$ is a unitary matrix. So it holds that $\tilde{Q}^*\tilde{Q} = \tilde{Q}\tilde{Q}^* = I$. It then follows that $\kappa_2(\tilde{Q}^*) = \kappa_2(\tilde{Q}) = 1$, where $\kappa_2(\cdot)$ represents the condition number in the Euclidean norm. □

## 4. Numerical experiments

In this section, we use some numerical experiments to examine the feasibility and effectiveness of the BAS iteration method and the BAS preconditioning. We show the advantages when BAS is used as a solver as well as a preconditioner, in terms of the number of iteration steps (denoted by "IT") and the elapsed CPU time in seconds (denoted by "CPU"). All experiments are performed by using MATLAB on an Intel Core(TM) i5-3470 CPU @ 3.2 GHz, 8G RAM.

The numerical experiments are performed for the distributed control problem (1.1)–(1.5) with the two dimensional case ($d = 2$). The computational domain is a unit square, that is, $\Omega = (0, 1) \times (0, 1) \in \mathbb{R}^2$. The target state is chosen as

$$y_d(x, y) = \begin{cases} (2x-1)^2(2y-1)^2, & \text{if } (x, y) \in \left(0, \dfrac{1}{2}\right)^2, \\ 0, & \text{otherwise}. \end{cases}$$

The tolerance for all methods is set to be $10^{-6}$, i.e., all iteration processes are terminated when the current-residuals satisfy

$$\frac{\|b - \mathscr{A}x^{(k)}\|_2}{\|b\|_2} \leq 10^{-6}$$

or if the prescribed iteration number $\mathrm{k}_{\max} = 1000$ is exceeded. $x^{(0)} = 0$ is the initial guess and $x^{(k)}$ is the $k$-th iteration of corresponding iteration processes. In addition, the mesh step-size $h$ is determined by $h = 1/(\sqrt{n}+1)$, where the degree of freedom $n$ represents the order of the matrix $M$ tested in our performance.

We perform several experiments for some cases for varying parameters $\alpha, \upsilon, w$, and $h$. Numerical results (IT and CPU) for this example are listed in Tables 1–4.

Firstly, we test that the BAS iteration method as a solver. We compare the BAS method (denoted by BAS in Tables 1 and 2) with GMRES and GMRES(5) methods. The corresponding number of the iteration steps (IT), and the computing time (CPU) with varying $h = 2^{-6}, 2^{-7}, 2^{-8}$ are listed in Table 1.

From Table 1 we see that the number of iteration steps of the BAS method is mildly dependent on the parameter $\upsilon$ and $w$, expect for a special case when $\upsilon = 10^{-2}$ and $w = 10^2$. As the mesh-size $h$ becomes small, the CPUs of the BAS method are increasing. In total, the CPU time of the BAS method is always little. We see that GMRES and GMRES(5) almost fail to solve this problem. Comparing the results in Table 1 we observe the BAS method is effectual both in terms of iteration counts and in terms of CPU time in most situations. So the BAS method is very competitive and effectual when solving this class of complex linear systems.

In Table 2, we simply take the iteration parameter $\alpha = 1 + w^2\upsilon$. We see that the iteration steps and the computing time with respect to both random $\alpha$ and $\alpha = 1 + w^2\upsilon$ are almost the same. This implies that in the actual implementation of the BAS method, we can simply take $\alpha = 1 + w^2\upsilon$, resulting in a parameter-free method.

Next, the BAS iteration is used as a preconditioner with GMRES(5), comparing with the preconditioners $P_{\mathrm{BD}}$ and $P_{\mathrm{AI}}$ defined in (1.14) and (1.15). The numerical results are reported in Table 3. In these tables, "OUT" represents the outer iteration steps, "INN" is the inner iteration steps and "CPU" is the total elapsed CPU times.

From Table 3 we see that when $P_{\mathrm{BAS}}(\alpha)$ is used as a preconditioner, it performs better than $P_{IB}$ both in terms of IT and CPU. In Table 4, we simply take $\alpha = \frac{1+w^2\upsilon}{1+w\sqrt{\upsilon}}$. We see that the iteration steps and the CPU times are less. This implies that we also can simply take the iteration parameter $\alpha = \frac{1+w^2\upsilon}{1+w\sqrt{\upsilon}}$ in actual implementations.

## 5. Concluding remarks

In this paper, we focus on solving a class of block two-by-two complex linear systems arising from the distributed optimal control with time-periodic parabolic equations. As the real part of this complex linear system is not positive definite, the PMHSS iteration method [28] cannot solve the problem directly. Based on the special structure of the complex linear system, we introduce a new block alternating splitting (BAS) iteration method and its preconditioner for solving the class of complex linear systems. The convergence of the BAS method and some spectral properties of the BAS preconditioned matrix are discussed. Numerical experiments show that the BAS iteration is competitive and feasible as both solver and preconditioner for Krylov subspace methods such as GMRES.

**Table 1**
Numerical results with different $h$, $v$ and $w$.

| $h$ | | $v$ | $w = 10^{-1}$ | | | $w = 1$ | | | $w = 10$ | | | $w = 10^2$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\alpha$ | IT | CPU | $\alpha$ | IT | CPU | $\alpha$ | IT | CPU | $\alpha$ | IT | CPU |
| $2^{-6}$ | BAS | $10^{-2}$ | 0.98 | 40 | 0.3916 | 1.01 | 40 | 0.3910 | 2.10 | 26 | 0.2887 | 32.0 | 187 | 1.4568 |
| | | $10^{-4}$ | 0.95 | 39 | 0.3831 | 0.98 | 39 | 0.3831 | 1.08 | 39 | 0.3819 | 1.18 | 27 | 0.2959 |
| | | $10^{-6}$ | 1.02 | 35 | 0.3539 | 1.09 | 35 | 0.3527 | 1.05 | 35 | 0.3541 | 1.09 | 35 | 0.3529 |
| | | $10^{-8}$ | 1.08 | 31 | 0.3249 | 1.02 | 31 | 0.3235 | 0.92 | 31 | 0.3246 | 0.98 | 31 | 0.3245 |
| | GM-RES | $10^{-2}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-4}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-6}$ | | 212 | 5.9267 | | 212 | 5.8877 | | 212 | 5.8940 | | 212 | 5.8913 |
| | | $10^{-8}$ | | 24 | 0.5689 | | 24 | 0.5690 | | 24 | 0.5522 | | 24 | 0.5491 |
| | GMR-RE(5) | $10^{-2}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-4}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-6}$ | | 1637 | 2.2946 | | 1637 | 2.2383 | | 1637 | 2.2758 | | 1637 | 2.2603 |
| | | $10^{-8}$ | | 32 | 0.4844 | | 32 | 0.4833 | | 32 | 0.4838 | | 32 | 0.4844 |
| $2^{-7}$ | BAS | $10^{-2}$ | 1.15 | 40 | 3.0897 | 1.08 | 40 | 3.3761 | 2.12 | 26 | 2.1721 | 34.0 | 212 | 16.737 |
| | | $10^{-4}$ | 1.15 | 39 | 2.8220 | 1.06 | 39 | 3.1216 | 1.10 | 39 | 3.0825 | 1.08 | 28 | 2.4631 |
| | | $10^{-6}$ | 0.98 | 38 | 2.9326 | 1.01 | 38 | 3.1622 | 1.11 | 38 | 3.0627 | 1.09 | 38 | 2.8105 |
| | | $10^{-8}$ | 1.02 | 30 | 2.6562 | 1.08 | 30 | 2.6074 | 1.01 | 30 | 2.5728 | 1.06 | 30 | 2.6032 |
| | GM-RES | $10^{-2}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-4}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-6}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-8}$ | | 134 | 5.5788 | | 134 | 5.5435 | | 134 | 5.5463 | | 134 | 5.5809 |
| | GMR-RE(5) | $10^{-2}$ | | – | – | | – | – | | – | – | | 528 | 0.1254 |
| | | $10^{-4}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-6}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-8}$ | | 1033 | 5.4305 | | 1034 | 5.4388 | | 1033 | 5.4350 | | 1033 | 5.4407 |
| $2^{-8}$ | BAS | $10^{-2}$ | 1.24 | 42 | 14.713 | 1.23 | 42 | 15.705 | 2.24 | 26 | 9.3129 | 35.0 | 204 | 68.837 |
| | | $10^{-4}$ | 1.15 | 40 | 13.515 | 1.11 | 40 | 14.039 | 1.15 | 40 | 13.303 | 1.05 | 28 | 10.002 |
| | | $10^{-6}$ | 1.04 | 38 | 13.185 | 1.11 | 38 | 12.889 | 1.24 | 38 | 12.411 | 0.91 | 38 | 12.621 |
| | | $10^{-8}$ | 1.05 | 30 | 10.183 | 1.12 | 30 | 10.735 | 1.14 | 30 | 10.843 | 1.10 | 30 | 10.487 |
| | GM-RES | $10^{-2}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-4}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-6}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-8}$ | | 495 | 352.68 | | 495 | 339.57 | | 495 | 341.25 | | 495 | 347.17 |
| | GMR-RE(5) | $10^{-2}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-4}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-6}$ | | – | – | | – | – | | – | – | | – | – |
| | | $10^{-8}$ | | – | – | | – | – | | – | – | | – | – |

**Table 2**
Numerical results for BAS method with $\alpha = 1 + w^2 v$.

| $h$ | $w$ | $w = 10^{-1}$ | | $w = 1$ | | $w = 10$ | | $w = 10^2$ | |
|---|---|---|---|---|---|---|---|---|---|
| | $v$ | IT | CPU | IT | CPU | IT | CPU | IT | CPU |
| $2^{-6}$ | $10^{-2}$ | 40 | 0.3904 | 40 | 0.3899 | 26 | 0.2884 | 335 | 3.2825 |
| | $10^{-4}$ | 39 | 0.3824 | 39 | 0.3825 | 39 | 0.3824 | 28 | 0.3031 |
| | $10^{-6}$ | 35 | 0.3531 | 35 | 0.3529 | 35 | 0.3531 | 35 | 0.3531 |
| | $10^{-8}$ | 31 | 0.3239 | 31 | 0.3237 | 31 | 0.3249 | 31 | 0.3242 |
| $2^{-7}$ | $10^{-2}$ | 40 | 3.0654 | 40 | 3.3185 | 26 | 2.3870 | 310 | 16.387 |
| | $10^{-4}$ | 39 | 2.8739 | 39 | 3.0710 | 39 | 2.9738 | 28 | 2.4632 |
| | $10^{-6}$ | 38 | 2.9127 | 38 | 2.8970 | 38 | 2.8570 | 38 | 2.9029 |
| | $10^{-8}$ | 30 | 2.7394 | 30 | 2.6900 | 30 | 2.8001 | 30 | 2.6134 |
| $2^{-8}$ | $10^{-2}$ | 42 | 14.331 | 42 | 15.059 | 27 | 9.9350 | 308 | 84.732 |
| | $10^{-4}$ | 40 | 14.086 | 40 | 13.420 | 40 | 13.705 | 28 | 9.8805 |
| | $10^{-6}$ | 38 | 12.228 | 38 | 11.893 | 38 | 12.067 | 38 | 12.452 |
| | $10^{-8}$ | 30 | 11.098 | 30 | 10.643 | 30 | 10.971 | 30 | 11.086 |

However, the parameters $\alpha$ in numerical experiments are chosen randomly, so they may not be the optimal parameters. The choice of the acceleration parameter $\alpha$ needs to be further studied for the BAS iteration.

**Table 3**
Numerical results using preconditioners $P_{\text{BAS}}$, $P_{\text{BD}}$ or $P_{\text{AI}}$ with GMRES(5).

| $h$ | | $\alpha$ | $v$ | $w=10^{-1}$ | | | $w=1$ | | | $w=10$ | | | $w=10^2$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Out | Inn | CPU | Out | Inn | CPU | Out | Inn | CPU | Out | Inn | CPU |
| $2^{-6}$ | $P_{BAS}$ | 0.25 | $10^{-2}$ | 2 | 2 | 0.1471 | 2 | 2 | 0.1435 | 3 | 2 | 0.2174 | 4 | 4 | 0.2545 |
| | | 0.83 | $10^{-4}$ | 4 | 1 | 0.2939 | 4 | 1 | 0.2916 | 4 | 1 | 0.2928 | 5 | 2 | 0.3826 |
| | | 1.00 | $10^{-6}$ | 4 | 3 | 0.3318 | 4 | 3 | 0.3185 | 4 | 3 | 0.3210 | 4 | 3 | 0.3317 |
| | | 1.32 | $10^{-8}$ | 4 | 2 | 0.3124 | 4 | 2 | 0.3053 | 4 | 2 | 0.3046 | 4 | 2 | 0.3039 |
| | $P_{BD}$ | | $10^{-2}$ | 4 | 4 | 0.3762 | 5 | 2 | 0.3896 | 6 | 4 | 0.5083 | 5 | 4 | 0.4337 |
| | | | $10^{-4}$ | 5 | 2 | 0.4033 | 5 | 3 | 0.4200 | 5 | 4 | 0.4340 | 7 | 2 | 0.5774 |
| | | | $10^{-6}$ | 5 | 2 | 0.3904 | 5 | 2 | 0.3913 | 5 | 2 | 0.3907 | 5 | 4 | 0.4186 |
| | | | $10^{-8}$ | 4 | 4 | 0.3206 | 5 | 2 | 0.3186 | 5 | 2 | 0.3190 | 4 | 3 | 0.3192 |
| | $P_{AI}$ | | $10^{-2}$ | 2 | 3 | 2.3943 | 2 | 3 | 2.4111 | 2 | 2 | 2.4334 | 1 | 5 | 1.9341 |
| | | | $10^{-4}$ | 4 | 5 | 4.1096 | 4 | 5 | 4.1769 | 4 | 5 | 4.2414 | 3 | 4 | 3.4993 |
| | | | $10^{-6}$ | 5 | 4 | 5.2584 | 5 | 4 | 4.9433 | 5 | 4 | 4.9820 | 5 | 4 | 4.9587 |
| | | | $10^{-8}$ | 5 | 5 | 5.3269 | 5 | 5 | 5.3617 | 5 | 5 | 5.3379 | 5 | 5 | 5.3325 |
| $2^{-7}$ | $P_{BAS}$ | 0.25 | $10^{-2}$ | 3 | 3 | 1.1415 | 3 | 3 | 1.2427 | 4 | 3 | 1.4480 | 5 | 2 | 1.6685 |
| | | 0.88 | $10^{-4}$ | 4 | 2 | 1.4199 | 4 | 2 | 1.6415 | 4 | 2 | 1.5344 | 5 | 2 | 1.6881 |
| | | 1.18 | $10^{-6}$ | 4 | 3 | 1.4474 | 4 | 3 | 1.4870 | 4 | 3 | 1.5472 | 5 | 2 | 1.7216 |
| | | 1.10 | $10^{-8}$ | 4 | 3 | 1.4534 | 4 | 3 | 1.4241 | 4 | 3 | 1.5390 | 4 | 3 | 1.4509 |
| | $P_{BD}$ | | $10^{-2}$ | 4 | 4 | 1.5933 | 5 | 2 | 1.8892 | 6 | 4 | 2.3398 | 5 | 4 | 2.0107 |
| | | | $10^{-4}$ | 5 | 2 | 1.8847 | 5 | 4 | 1.9592 | 5 | 4 | 1.9430 | 7 | 2 | 2.6683 |
| | | | $10^{-6}$ | 5 | 2 | 1.9063 | 5 | 2 | 1.8683 | 5 | 2 | 1.8671 | 5 | 4 | 1.9568 |
| | | | $10^{-8}$ | 5 | 1 | 1.8141 | 5 | 1 | 1.7989 | 5 | 1 | 1.8012 | 5 | 1 | 1.7971 |
| | $P_{AI}$ | | $10^{-2}$ | 3 | 4 | 12.438 | 3 | 4 | 12.419 | 4 | 2 | 15.946 | 1 | 5 | 7.1265 |
| | | | $10^{-4}$ | 5 | 1 | 21.334 | 5 | 1 | 20.276 | 5 | 1 | 20.823 | 3 | 5 | 13.702 |
| | | | $10^{-6}$ | 5 | 5 | 24.267 | 5 | 5 | 25.457 | 5 | 5 | 25.166 | 5 | 5 | 24.894 |
| | | | $10^{-8}$ | 6 | 1 | 26.549 | 6 | 1 | 27.344 | 6 | 1 | 26.167 | 6 | 1 | 26.631 |
| $2^{-8}$ | $P_{BAS}$ | 0.34 | $10^{-2}$ | 3 | 4 | 5.4447 | 3 | 4 | 5.4335 | 4 | 1 | 6.2326 | 4 | 4 | 7.6846 |
| | | 0.67 | $10^{-4}$ | 4 | 2 | 6.9724 | 4 | 2 | 6.8277 | 4 | 2 | 6.8446 | 5 | 2 | 8.0905 |
| | | 1.04 | $10^{-6}$ | 4 | 3 | 7.4595 | 4 | 3 | 7.4108 | 4 | 3 | 7.3938 | 4 | 4 | 7.6948 |
| | | 1.13 | $10^{-8}$ | 4 | 4 | 7.6617 | 4 | 4 | 7.6453 | 4 | 4 | 7.6448 | 4 | 4 | 7.6527 |
| | $P_{BD}$ | | $10^{-2}$ | 4 | 4 | 7.7651 | 5 | 2 | 8.9601 | 6 | 4 | 11.210 | 5 | 4 | 9.3958 |
| | | | $10^{-4}$ | 5 | 2 | 8.8516 | 5 | 4 | 9.2654 | 5 | 4 | 9.3301 | 7 | 2 | 12.803 |
| | | | $10^{-6}$ | 5 | 2 | 8.9165 | 5 | 2 | 9.0478 | 5 | 3 | 9.2944 | 5 | 4 | 9.2860 |
| | | | $10^{-8}$ | 5 | 2 | 8.9551 | 5 | 2 | 9.1141 | 5 | 2 | 8.8573 | 5 | 2 | 8.7365 |
| | $P_{AI}$ | | $10^{-2}$ | 3 | 4 | 83.237 | 3 | 4 | 84.136 | 4 | 2 | 98.238 | 1 | 5 | 41.277 |
| | | | $10^{-4}$ | 5 | 1 | 122.85 | 5 | 1 | 127.14 | 5 | 1 | 124.15 | 3 | 5 | 89.183 |
| | | | $10^{-6}$ | 5 | 5 | 148.63 | 5 | 5 | 149.39 | 5 | 5 | 146.18 | 5 | 5 | 146.77 |
| | | | $10^{-8}$ | 6 | 1 | 154.58 | 6 | 1 | 152.20 | 6 | 1 | 152.63 | 6 | 1 | 154.25 |

**Table 4**
Numerical results using preconditioners $P_{\text{BAS}}$ with $\alpha = \frac{1+w^2 v}{1+w\sqrt{v}}$.

| $h$ | $v$ | $w=10^{-1}$ | | | $w=1$ | | | $w=10$ | | | $w=10^2$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Out | Inn | CPU | Out | Inn | CPU | Out | Inn | CPU | Out | Inn | CPU |
| $2^{-6}$ | $10^{-2}$ | 4 | 2 | 0.3131 | 4 | 1 | 0.2910 | 4 | 2 | 0.3052 | 5 | 2 | 0.3930 |
| | $10^{-4}$ | 4 | 4 | 0.3378 | 4 | 4 | 0.3357 | 4 | 3 | 0.3197 | 5 | 2 | 0.3934 |
| | $10^{-6}$ | 4 | 3 | 0.3240 | 4 | 3 | 0.3206 | 5 | 2 | 0.3337 | 5 | 1 | 0.3786 |
| | $10^{-8}$ | 5 | 2 | 0.3827 | 5 | 2 | 0.3931 | 5 | 2 | 0.3942 | 5 | 2 | 0.3939 |
| $2^{-7}$ | $10^{-2}$ | 4 | 2 | 1.4141 | 4 | 1 | 1.3407 | 4 | 2 | 1.4636 | 5 | 2 | 1.6630 |
| | $10^{-4}$ | 4 | 4 | 1.6020 | 4 | 4 | 1.5965 | 4 | 2 | 1.4678 | 5 | 2 | 1.6967 |
| | $10^{-6}$ | 4 | 3 | 1.4723 | 4 | 3 | 1.4671 | 5 | 3 | 1.7288 | 4 | 4 | 1.5978 |
| | $10^{-8}$ | 5 | 2 | 1.6682 | 5 | 2 | 1.6654 | 5 | 2 | 1.6671 | 5 | 2 | 1.6676 |
| $2^{-8}$ | $10^{-2}$ | 4 | 2 | 6.8157 | 4 | 1 | 6.4407 | 4 | 2 | 7.0407 | 5 | 2 | 8.0422 |
| | $10^{-4}$ | 4 | 4 | 7.7323 | 4 | 4 | 7.7500 | 4 | 3 | 7.3870 | 5 | 2 | 7.9798 |
| | $10^{-6}$ | 4 | 3 | 7.4295 | 4 | 3 | 7.3738 | 5 | 3 | 8.3733 | 4 | 4 | 7.7233 |
| | $10^{-8}$ | 5 | 2 | 8.1181 | 5 | 2 | 8.1287 | 5 | 2 | 8.1732 | 5 | 2 | 8.1167 |

## Acknowledgments

The authors are very much indebted to the anonymous referees for their constructive suggestions and helpful comments, which greatly improved the original manuscript of this paper.

## References

[1] M. Kollmann, M. Kolmbauer, A preconditioned minres solver for time-periodic parabolic optimal control problems, Numer. Linear Algebra Appl. 20 (2012) 761–784.

[2] W. Krendl, V. Simoncini, W. Zulehner, Stability estimates and structural spectral properties of saddle point problems, Numer. Math. 124 (2013) 183–213.

[3] M. Gunzburger, C. Trenchea, Optimal control of the time-periodic MHD equations, Nonlinear Anal. 63 (2005) 1687–1699.

[4] D.W. Peaceman Jr., H.H. Rachford, The numerical solution of parabolic and elliptic differential equations, J. Soc. Ind. Appl. Math. 3 (1995) 28–41.

[5] M. Kolmbauer, U. Langer, A robust preconditioned minres solver for distributed time-periodic eddy current optimal control problems, SIAM J. Sci. Comput. 34 (2012) 785–809.

[6] T. Rees, H.S. Dollar, A.J. Wathen, Optimal solvers for PDE-constrained optimization, SIAM J. Sci. Comput. 32 (2010) 271–298.

[7] Z.-Z. Bai, Block preconditioners for elliptic PDE-constrained optimization problems, Computing 91 (2011) 379–395.

[8] G.-F. Zhang, Z. Zheng, Block-symmertic and block-lower-triangular preconditioners for PDE-constrained optimization problems, J. Comput. Math. 31 (2013) 370–381.

[9] Z.-Z. Bai, Block alternating splitting implicit iteration methods for saddle-point problems from time-harmonic eddy current models, Numer. Linear Algebra Appl. 19 (2012) 914–936.

[10] Z.-Z. Bai, G.H. Golub, Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems, IMA J. Numer. Anal. 27 (2007) 1–23.

[11] M.-Q. Jiang, Y. Cao, On local Hermitian and skew-Hermitian splitting iteration methods for generalized saddle point problems, J. Comput. Appl. Math. 231 (2009) 973–982.

[12] L.A. Krukier, B.L. Krukier, Z.-R. Ren, Generalized skew-Hermitian triangular splitting iteration methods for saddle-point linear systems, Numer. Linear Algebra Appl. 21 (2014) 152–170.

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

[14] S.-L. Wu, C.-X. Li, A splitting iterative method for the discrete dynamic linear systems, J. Comput. Anal. Math. 267 (2014) 49–60.

[15] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian methods for non-Hermitian positive semidefinite linear systems, Numer. Math. 98 (2004) 1–32.

[16] M. Zhu, G.-F. Zhang, Z. Zheng, Z.-Z. Liang, On HSS-based sequential two-stage method for non-Hermitian saddle point problems, Appl. Math. Comput. 242 (2014) 907–916.

[17] M. Kolmbauer, U. Langer, Efficient solvers for some classes of time-periodic eddy current optimal control problems, in: Numeri. Solution Partial Differential Eq., Vol. 45, 2013, pp. 203–216.

[18] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.

[19] W. Zulehner, Nonstandard norms and robust estimates for saddle point problems, SIAM J. Matrix Anal. Appl. 32 (2011) 536–560.

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

[21] A. Axelsson, O. Kucherov, Real valued iterative methods for solving complex symmetric linear systems, Numer. Linear Algebra Appl. 7 (2000) 197–218.

[22] A. Bunse-Gerstner, R. Stöver, On a conjugate gradient–type method for solving complex symmetric linear systems, Linear Algebra Appl. 287 (1999) 105–123.

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

[24] Z.-Z. Bai, M.K. Ng, J.-Y. Pan, Alternating splitting waveform relaxation method and its successive overrelaxation acceleration, Comput. Math. Appl. 49 (2005) 157–170.

[25] Z.-Z. Bai, G.H. Golub, M.K. Ng, On successive overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations, Numer. Linear Algebra Appl. 14 (2007) 319–335.

[26] Z.-Z. Bai, G.H. Golub, M.K. Ng, On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, Linear Algebra Appl. 428 (2008) 413–440.

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

[28] Z.-Z. Bai, M. Benzi, F. Chen, On preconditioned MHSS iteration methods for complex symmetric linear systems, Numer. Algor. 56 (2011) 297–317.

[29] Z.-Z. Bai, M. Benzi, F. Chen, Z.-Q. Wang, 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 (2013) 343–369.

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