New Preconditioned Iteration Method Solving the Special Linear System from the PDE-Constrained Optimal Control Problem

Yan-Ran Li, Xin-Hui Shao * and Shi-Yu Li

Department of Mathematics, College of Sciences, Northeastern University, Shenyang 110819, China; 1800116@stu.neu.edu.cn (Y.-R.L.); 1800115@stu.neu.edu.cn (S.-Y.L.)
* Correspondence: shaoxinhui@mail.neu.edu.cn; Tel.: +86-024-83683382

Abstract: In many fields of science and engineering, partial differential equation (PDE) constrained optimal control problems are widely used. We mainly solve the optimization problem constrained by the time-periodic eddy current equation in this paper. We propose the three-block splitting (TBS) iterative method and proved that it is unconditionally convergent. At the same time, the corresponding TBS preconditioner is derived from the TBS iteration method, and we studied the spectral properties of the preconditioned matrix. Finally, numerical examples in two-dimensions is applied to demonstrate the advantages of the TBS iterative method and TBS preconditioner with the Krylov subspace method.

Keywords: time-periodic eddy current problem; iteration methods; preconditioner; generalized saddle point problem; PDE-constrained optimization

1. Introduction

We consider the time-periodic eddy current optimization problem: Find the control $u(x, t)$ and the state $y(x, t)$ to make the cost function minimum, which is subject to the time-periodic problem

$$J(y, u) = \frac{\beta}{2} \int_0^T \int_{\Omega} |u(x, t)|^2 dx dt + \frac{1}{2} \int_0^T \int_{\Omega_0} |y(x, t) - y_d(x, t)|^2 dx dt$$

$$\begin{align*}
\sigma \frac{\partial y(x, t)}{\partial t} + \text{curl} (\text{curl} (y(x, t))) + \sigma y(x, t) &= u(x, t) \text{ in } Q_T, \\
y(x, t) \times n &= 0 \text{ on } \Gamma_T, \\
u(x, 0) &= u(x, T) \text{ in } Q_T, \\
y(x, 0) &= y(x, T) \text{ in } Q_T.
\end{align*}$$

where $\Omega$ is a bounded domain in $\mathbb{R}^d$ for $d \in \{2, 3\}$ and its boundary $\Gamma$ is Lipschitz-continuous. Unit outer normal vector of $\Omega$ is marked as $n$. $Q_T = \Omega \times (0, T)$ is space-time cylinder, which outer surface is $\Sigma_T = \Gamma \times (0, T)$ with $T > 0$. Besides, we denote the given target state function as $y_d(x, t)$, magnetic reluctivity $\nu$ uniformly positive, conductivity $\sigma \in L^\infty(\Omega)$ is positive constant, $\beta > 0$ is a regularization or cost parameter, and $\varepsilon > 0$ is an additional regularization parameter. In some actual calculation we may take $\varepsilon = 0$.

The partial differential equations constrained optimization problem was first proposed by J.L. Lions [1]. In the past 50 years, the research on optimal control problems with PDE constrained has developed to a certain extent, and has important applications in modern industrial, medical, economic and other application areas, such as semiconductor material design, water pressure control in oil exploitation, image registration and option price in parameter determination, etc. [1,2]. The eddy current equation originates from the problem of the time-period electromagnetic field, and usually the time-period current is the driving
control force. It is used widely in areas such as non-destructive testing of conductive materials [3].

In actual calculations, for such problems, \( y_d(x,t) \) is time-periodic in [4,5], that is,
\[
y_d(x,t) = \hat{y}_d(x)e^{i\omega t},
\]
with the angular frequency \( \omega = \frac{2\pi}{T} \). Then the solutions of the original control problems (1) and (2) are in the form of
\[
u(x,t) = \hat{u}(x)e^{i\omega t}, y(x,t) = \hat{y}(x)e^{i\omega t},
\]
\( \hat{u}(x) \) and \( \hat{y}(x) \) are the solutions of the equations
\[
\min_{\hat{u},\hat{y}} \int_{\Omega} |\hat{u}(x)|^2 \, dx + \frac{1}{2} \int_{\partial \Omega} |\hat{y}(x) - \hat{y}_d(x)|^2 \, dx,
\]
\[
s.t. \begin{cases} 
\text{i} \omega \hat{y}(x) + \text{curl}(\text{v} \text{curl}(\hat{y}(x))) + \epsilon \hat{y}(x) = \hat{u}(x) \text{ in } \Omega, \\
\hat{y}(x) \times n = 0 \text{ on } \Gamma.
\end{cases}
\]
For simplicity, we make the restrictive assumption that \( \Omega_0 = \Omega \). Since the problem operator of (5) is Hermitian [6], the method of discretize-then-optimize [7–9] is adopted to obtain the following finite dimensional optimization problems:
\[
\min_{y,a} \frac{1}{2}((y - y_d)^* M (y - y_d) + \frac{\beta}{2} u^* M u),
\]
\[
s.t. \begin{cases} 
\text{i} \omega a M y + Ky - Mu = 0,
\end{cases}
\]
where \( K = [K_{ij}]_{n \times n} \) is the stiffness matrix and \( M = [M_{ij}]_{n \times n} \) is the mass matrix. \( y, y_d, u \)
denote the discrete form of \( \hat{y}, \hat{y}_d, \hat{u} \), respectively. \((\cdot)^*\) means conjugate transpose.
The Lagrangian function of (6) is
\[
\mathcal{L}(y,u,p) = \frac{1}{2}((y - y_d)^* M (y - y_d) + \frac{\beta}{2} u^* M u + p^* (\text{i} \epsilon \omega a M y + Ky - Mu),
\]
with \( p \) is the Lagrangian multiplier. Then the following complex linear system can be derived:
\[
\begin{bmatrix} M & 0 & -K - \text{i} \omega a M \\
0 & \beta M & -M \\
K + \text{i} \omega a M & -M & 0 \end{bmatrix} \begin{bmatrix} y \\
u \\
p \end{bmatrix} = \begin{bmatrix} M y_d \\
0 \\
0 \end{bmatrix},
\]
where \( \bar{\omega} = \sigma \omega \). We use the relationship of \( u = \frac{1}{p} p \) obtained in the second row of (8) and a scaling \( \hat{u} = -\sqrt{p} u \) [6] to reduce (8) as follows:
\[
\mathcal{A} = \begin{bmatrix} M & -\sqrt{p} (K + \text{i} \omega a M) \\
\sqrt{p} (K + \text{i} \omega a M) & M \end{bmatrix} \begin{bmatrix} y \\
\hat{u} \end{bmatrix} = \begin{bmatrix} M y_d \\
0 \end{bmatrix} = b
\]
where \( K \in \mathbb{R}^{n \times n} \) is symmetric and \( M \in \mathbb{R}^{n \times n} \) is symmetric positive definite.
Obviously, (9) is a kind of generalized saddle point problem. The iterative method is an effective way to solve this kind of problems. The Hermitian and skew-Hermitian splitting (HSS) iterative method was proposed by Bai et al. in 2003 [10], which greatly accelerated the solution speed of the equation. In 2013 [11], for system
\[
\begin{bmatrix} W & -T \\
T & W \end{bmatrix} \begin{bmatrix} y \\
z \end{bmatrix} = \begin{bmatrix} p \\
g \end{bmatrix},
\]
in order to avoid the dense matrix \( \begin{bmatrix} aV & -T \\
T & aV \end{bmatrix} \) generated by the skew-Hermitian part of the HS splitting during the iteration, the coefficient matrix was rotated with matrix \( \begin{bmatrix} 0 & I \\
-I & 0 \end{bmatrix} \) and then HS splitting was carried out again. The Hermitian part of the new coefficient matrix can be left matrix of the second iteration equation. It greatly improves calculation efficiency. Meanwhile, it induced a new preconditioner. However, rotation matrix is
involved in the preconditioner and the iteration matrix, which adds to the computational complexity. In 2015, in [12], based on the idea of matrix rotation and [11], Zheng, Zhang et al. use two matrices to rotate the coefficient of the equation, then the idea of [11] is used to construct the block alternating splitting (BAS) iteration method and induces BAS preconditioner, which generalize the preconditioner \( P_{BD} \) in [13].

When the original equation is discrete, the spectral distribution is usually scattered and the condition number is large. The convergence speed of the Krylov subspace iteration algorithm is usually slow so an effective preconditioner is needed to improve the spectral property of coefficient matrix, so as to improve the calculation efficiency and accuracy. In 2014 [14], Axelsson, Neytcheva et al. rewrite the complex linear system \((A + iB)(x + iy) = f + ig\) into \[
\begin{bmatrix}
A & -B \\
B & A
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix} =
\begin{bmatrix}
f \\
g
\end{bmatrix},
\]
and rewrite it as \((A + BA^{-1}B)x = f + BA^{-1}g\).

Reference [15] gives a new preconditioner \( C_a = (A + aB)A^{-1}(A + aB) \), in every step of calculation, the coefficient matrix involves \( A \), and [14] rewrites the coefficient matrix \( C_a^{-1}(A + BA^{-1}B) \) into the form of \( A + aB \), so as to avoid the situation that the condition number of \( A \) is not good enough. In 2016, Axelsson, Farouq et al. [16], for the coefficient matrix \[
\begin{bmatrix}
M & \sqrt{\beta}K \\
\sqrt{\beta}K & -M
\end{bmatrix}
\]
obtained by discretizing the Poisson equation constrained optimization problem, apply the method in [14] to obtain \( P = \begin{bmatrix}
M + \sqrt{\beta}K & 0 \\
0 & M + \sqrt{\beta}K
\end{bmatrix}, \)

avoiding the problem of solving \( M^{-1} \), and reducing the computational complexity. In 2013, in [13], Kreidl and Simoncini et al. proposed a block diagonal preconditioner for the coefficient matrix of the optimization problem constrained by the time-periodic parabolic problem. In [17], Zheng, Zhang et al. proposed a new block triangle preconditioner \( P_{tri} = \begin{bmatrix}
M & \sqrt{\beta}(K + i\omega M) \\
\sqrt{\beta}(K + i\omega M) & S
\end{bmatrix}, \)
in which \( S = \begin{bmatrix}
(1 + \beta\omega^2)M + \sqrt{\beta}K \\
(1 + \beta\omega^2)M + \sqrt{\beta}K
\end{bmatrix} \)
approximately replaces Schur complement \((1 + \beta\omega^2)M + \beta KM^{-1}K\) of coefficient matrix.

After preconditioning, the spectral radius of the system is clustered to \([\frac{1}{2}, 1] \), which effectively accelerates the calculation of Krylov subspace iteration method. In 2018 [18], for coefficient matrix derived from optimization problems constrained by time-periodic parabolic problems, based on the idea of [16] and the coefficient matrix belongs to complex matrix, a new preconditioner was obtained, which performs better when preconditioning the Krylov subspace method. In [6], Axelsson added adjustable parameters \( a \) on the basis of [18], and obtained \[
\begin{bmatrix}
M & \sqrt{\beta}(K - i\omega M) \\
\sqrt{\beta}(K - i\omega M) & aM + 2\sqrt{\beta}(a + \beta\omega^2)K
\end{bmatrix}
\]
that, under the condition of optimal parameters, even if the stiffness matrix \( K \) is positive semidefinite, the spectral radius of the preconditioned system can be within the range \([\frac{1}{2}, 1] \).

The purpose of this article is to efficiently solve Equation (9). In this paper, a three-block splitting (TBS) scheme of coefficient matrix in (9) is proposed. Then, according to the new split format, the TBS iteration method is established to solve (9), and theoretically prove that TBS iterative method is convergent unconditionally. Meanwhile, the TBS iterative method can naturally derive a preconditioner. It can be proved that the preconditioned matrix is concentrated in a certain interval (to be proven). In addition, in the actual calculation, the iteration steps and running time of the iteration method have been greatly improved. The application of the preconditioner greatly reduces the number of the Krylov subspace method and improves the calculation efficiency.

Here is the structure of this paper. We propose the TB splitting and give the corresponding TBS iteration method in Section 2, then prove that it is convergent. In Section 3, we give the TBS preconditioner matrix derived from the TBS iteration method, and its spectral property is analyzed. Some actual experiments in Section 4 illustrate the efficiency and feasibility of the TBS iteration method and TBS preconditioner. Finally, the conclusions are given in Section 5.
2. TBS Iteration Method

First, according to HS splitting [10], we divide the coefficient matrix $A$ of (9) into

$$ A = A_1 + A_2 = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} + \begin{bmatrix} 0 & -\sqrt{\beta(K - i\omega M)} \\ \sqrt{\beta}(K + i\omega M) & 0 \end{bmatrix}, $$

then split $A_2$ into

$$ A_2 = B_2 + A_3 = \begin{bmatrix} 0 & -\sqrt{\beta K} \\ \sqrt{\beta K} & 0 \end{bmatrix} + \begin{bmatrix} 0 & i\sqrt{\beta\omega M} \\ i\sqrt{\beta\omega M} & 0 \end{bmatrix}, $$

let $B_1 = A_1 + A_3 = \begin{bmatrix} M & i\sqrt{\beta\omega M} \\ i\sqrt{\beta\omega M} & M \end{bmatrix}$, finally we get three-block (TB) splitting of the coefficient matrix

$$ A = B_1 + B_2 = \begin{bmatrix} M & i\sqrt{\beta\omega M} \\ i\sqrt{\beta\omega M} & M \end{bmatrix} + \begin{bmatrix} 0 & -\sqrt{\beta K} \\ \sqrt{\beta K} & 0 \end{bmatrix}, $$

further we have

$$ A = (\alpha I + B_1) - (\alpha I - B_2) = (\alpha I + B_2) - (\alpha I - B_1), $$

where $I \in \mathbb{R}^{n \times n}$ is an identity matrix, $\alpha > 0$.

According to the splitting in (13), we can get the following TBS iterative method.

**TBS Iteration Method.** Define $x^{(0)} = \begin{bmatrix} y^{(0)} \\ \hat{u}^{(0)} \end{bmatrix} \in \mathbb{C}^{2n}$ as an arbitrary initial guess. For $k = 0, 1, 2, \ldots$, calculate the next $x^{(k+1)}$ as follows until $\{x^{(k)}\}_{k=0}^{\infty} \subset \mathbb{C}^{2n}$ is convergent:

$$ \begin{bmatrix} y^{(k+\frac{1}{2})} \\ \hat{u}^{(k+\frac{1}{2})} \end{bmatrix} = \begin{bmatrix} \alpha I + M & i\sqrt{\beta\omega M} \\ i\sqrt{\beta\omega M} & \alpha I + M \end{bmatrix} \begin{bmatrix} y^{(k)} \\ \hat{u}^{(k)} \end{bmatrix} + \begin{bmatrix} \alpha I \sqrt{\beta K} \\ -\sqrt{\beta K} \alpha I \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}, $$

$$ = \begin{bmatrix} \alpha I - M & -i\sqrt{\beta\omega M} \\ -i\sqrt{\beta\omega M} & \alpha I - M \end{bmatrix} \begin{bmatrix} y^{(k)} \\ \hat{u}^{(k)} \end{bmatrix} + \begin{bmatrix} \alpha I \sqrt{\beta K} \\ -\sqrt{\beta K} \alpha I \end{bmatrix} \begin{bmatrix} M y_d \\ 0 \end{bmatrix}, $$

with $\alpha > 0$.

By using the Gauss elimination method, the steps of the TBS iteration method are described in Algorithm 1.

---

**Algorithm 1** For the given vector $\begin{bmatrix} y^{(k)} \\ \hat{u}^{(k)} \end{bmatrix} \in \mathbb{C}^{2n}$, follow the procedure below to calculate $\begin{bmatrix} y^{(k+1)} \\ \hat{u}^{(k+1)} \end{bmatrix} \in \mathbb{C}^{2n}$

1. Solve $\hat{u}^{(k+\frac{1}{2})}$ from
   $$ \left( (\alpha I + M)^2 + \beta\omega^2 M^2 \right) \hat{u}^{(k+\frac{1}{2})} = -\sqrt{\beta}(\alpha I + M)K y^{(k)} + a(\alpha I + M) u^{(k)} - i\alpha \sqrt{\beta\omega M} y^{(k)} - i\beta\omega My^{(k)} - i\sqrt{\beta\omega M} y_d. $$

2. Solve $y^{(k+\frac{1}{2})}$ from $$(\alpha I + M) y^{(k+\frac{1}{2})} = -i\sqrt{\beta\omega M} \hat{u}^{(k+\frac{1}{2})} + a y^{(k)} + \sqrt{\beta K} u^{(k)} + M y_d.$$

3. Solve $y^{(k+1)}$ from
   $$ a(\alpha I - M) y^{(k+1)} = a(\alpha I - M) y^{(k+\frac{1}{2})} - i\alpha \sqrt{\beta\omega M} u^{(k+\frac{1}{2})} + a My_d - i\beta\omega KM y^{(k+\frac{1}{2})} + \sqrt{\beta K}(\alpha I - M) \hat{u}^{(k+\frac{1}{2})}. $$

4. Compute $\hat{u}^{(k+1)}$ from
   $$ -i\sqrt{\beta\omega M} y^{(k+\frac{1}{2})} + (\alpha I - M) \hat{u}^{(k+\frac{1}{2})} - \sqrt{\beta K} y^{(k+1)}.$$

---

As can be seen from Algorithm 1, we can use the conjugate gradient (CG) method or the Cholesky decomposition method for the first three equations, of which the coefficient matrices are real symmetric positive definite matrices. In addition, the fourth step only
needs direct calculation without any additional solution steps. At the same time, the second step uses $\tilde{u}^{(k+\frac{1}{2})}$ obtained in the first step, and the fourth step uses $\tilde{y}^{(k+1)}$ from the third step, which greatly improves the calculation efficiency and shortens the calculation time of the TBS iteration method. This will also be demonstrated in numerical experiments.

By directly substituting, we can rewrite (14) as follows

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

where

$$L(\alpha) = \begin{bmatrix} \alpha l & \sqrt{\beta} k & \cdots & 0 \\ 0 & \alpha l & \sqrt{\beta} k & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha l \end{bmatrix},$$

is the iteration matrix of TBS iterative method, and

$$R(\alpha) = \begin{bmatrix} \alpha l & 0 & \cdots & 0 \\ 0 & \alpha l & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha l \end{bmatrix}.$$

Next, we will prove the unconditional convergence of the TBS iteration method. Moreover, the upper bound of the spectral radius of $L(\alpha)$ only depends on the parameters $\alpha$, which provides great convenience for us to further find the optimal parameters.

**Theorem 1.** (Convergence Theorem). Define $A \in \mathbb{C}^{2n \times 2n}$ as in (9), where $M \in \mathbb{R}^{n \times n}$ is a symmetric positive definite, $K \in \mathbb{R}^{n \times n}$ is symmetric, and $\alpha$ is a positive constant. Then for any $\alpha$, $\rho(L(\alpha)) \leq \sigma(\alpha)$, holds where

$$\sigma(\alpha) = \max_{\lambda \in \text{sp}(M)} \sqrt{(\alpha - \lambda_j)^2 + \beta \omega^2 \lambda_j^2} < 1 \forall \alpha > 0.$$

$\rho(L(\alpha))$ represents the spectral radius of $L(\alpha)$ and $\text{sp}(M)$ represents the eigenvalue of matrix $M$.

That is, for any initial vector, the TBS iterative method defined in method 2.1 is unconditionally convergent. Note that the convergence factor $\sigma(\alpha) \to 1$ as $\lambda_j \to 0$.

**Proof.** From the matrix similarity, we get

$$\mathcal{T}(\alpha) = \begin{bmatrix} \alpha l - M & -i \sqrt{\beta} K & \cdots & 0 \\ 0 & \alpha l - M & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha l \end{bmatrix},$$

and obviously $\rho(\mathcal{T}(\alpha)) = \rho(L(\alpha))$. Since the matrices $M$ and $K$ are real symmetric, there are orthogonal matrices $Q_M, Q_K \in \mathbb{R}^{n \times n}$, such that $M = Q_M \Lambda M Q_M^T$, $K = Q_K \Lambda K Q_K^T$, where the matrix $\Lambda_M = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{R}^{n \times n}$ and the matrix $\Lambda_K = \text{diag}(\mu_1, \mu_2, \ldots, \mu_n) \in \mathbb{R}^{n \times n}$. Then

$$\rho(\mathcal{T}(\alpha)) \leq \left\| \begin{bmatrix} \alpha l - M & -i \sqrt{\beta} K & \cdots & 0 \\ 0 & \alpha l - M & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha l \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} \alpha l & \sqrt{\beta} K \\ \sqrt{\beta} K & \alpha l \end{bmatrix} \right\|_2.$$

because

$$\begin{bmatrix} \alpha l & \sqrt{\beta} K \\ -\sqrt{\beta} K & \alpha l \end{bmatrix} = \begin{bmatrix} \alpha l & -\beta \Lambda K \\ \sqrt{\beta} \Lambda K & \alpha l \end{bmatrix}.$$
we have
\[
\begin{bmatrix}
\alpha l & \sqrt{\beta} \Lambda_K \\
-\sqrt{\beta} \Lambda_K & \alpha l
\end{bmatrix}
\begin{bmatrix}
\alpha l & \sqrt{\beta} \Lambda_K \\
-\sqrt{\beta} \Lambda_K & \alpha l
\end{bmatrix}^{-1} = 1,
\]

Then
\[
\rho(\mathcal{L}(\alpha)) \leq \left\| \begin{bmatrix}
\alpha l - \Lambda_M & -i \sqrt{\beta} \omega \Lambda_M \\
-i \sqrt{\beta} \omega \Lambda_M & \alpha l - \Lambda_M
\end{bmatrix}
\begin{bmatrix}
\alpha l + \Lambda_M & i \sqrt{\beta} \omega \Lambda_M \\
\sqrt{\beta} \omega \Lambda_M & \alpha l + \Lambda_M
\end{bmatrix}^{-1}
\right\|_2
= \max_{\lambda_j \in \sigma(M)} \left( \frac{(\alpha - \lambda_j)^2 + \beta \omega^2 \lambda_j^2}{(\alpha + \lambda_j)^2 + \beta \omega^2 \lambda_j^2} \right)
= \max_{\lambda_j \in \sigma(M)} \left( \frac{\alpha - \lambda_j}{\alpha + \lambda_j} \right)
= \sigma(\alpha).
\]

Note that \( \alpha > 0, \lambda_j > 0, \beta > 0, \omega > 0 \), so \( \sigma(\alpha) < 1 \) is always true for any \( \alpha > 0 \).

In the following we consider solving the optimal parameters that minimize \( \sigma(\alpha) \).

**Theorem 2.** *(The optimal parameter theorem).* Assume that Theorem 1 holds, then
\[
\alpha^* = \underset{\alpha}{\text{argmin}} \sigma(\alpha) = \sqrt{(1 + \beta \omega^2) \lambda_{\min} \lambda_{\max}},
\]
where \( \lambda_{\max} \) and \( \lambda_{\min} \) are the maximum and minimum eigenvalues of the matrix \( M \), respectively.

**Proof.** Define \( \sigma'(\lambda) \) as the derivation of \( \sigma(\lambda) \), then according to the relation between the \( \sigma'(\lambda) \) and monotonicity of \( \sigma(\lambda) \), we have
\[
\sigma(\alpha) = \max \left\{ \frac{(\alpha - \lambda_{\min})^2 + \beta \omega^2 \lambda_{\min}^2}{(\alpha + \lambda_{\min})^2 + \beta \omega^2 \lambda_{\min}^2}, \frac{(\alpha - \lambda_{\max})^2 + \beta \omega^2 \lambda_{\max}^2}{(\alpha + \lambda_{\max})^2 + \beta \omega^2 \lambda_{\max}^2} \right\}
= \max \{ \sigma_1(\alpha), \sigma_2(\alpha) \}.
\]

Derivatives for \( \sigma_1(\alpha) \) and \( \sigma_2(\alpha) \), respectively. By comparing the monotonicity and maximum of the function, we find that if \( \alpha^* \) makes \( \sigma(\alpha) \), then it satisfies
\[
\sqrt{\frac{(\alpha - \lambda_{\min})^2 + \beta \omega^2 \lambda_{\min}^2}{(\alpha + \lambda_{\min})^2 + \beta \omega^2 \lambda_{\min}^2}} = \sqrt{\frac{(\alpha - \lambda_{\max})^2 + \beta \omega^2 \lambda_{\max}^2}{(\alpha + \lambda_{\max})^2 + \beta \omega^2 \lambda_{\max}^2}},
\]
solve it, we can obtain
\[
\alpha^* = \sqrt{(1 + \beta \omega^2) \lambda_{\min} \lambda_{\max}}.
\]

Theorem 2 is proved. \( \square \)

### 3. TBS Preconditioner

As shown in [12], there is a unique split
\[
A = P(\alpha) - G(\alpha),
\] (15)

for Equation (9), where \( P(\alpha) \) is nonsingular, so that the iterative matrix \( L(\alpha) \) can be expressed (i.e., \( L(\alpha) = P^{-1}(\alpha)G(\alpha) \)), then we have
\[
P(\alpha) = \frac{1}{2\alpha} \begin{bmatrix}
\alpha l + M & i \sqrt{\beta} \omega M \\
-i \sqrt{\beta} \omega M & \alpha l + M
\end{bmatrix}
\begin{bmatrix}
\alpha l & -\sqrt{\beta} K \\
\sqrt{\beta} K & \alpha l
\end{bmatrix}
\] (16)
\[
G(\alpha) = \frac{1}{2\alpha} \begin{bmatrix}
\alpha l - M & -i \sqrt{\beta} \omega M \\
-i \sqrt{\beta} \omega M & \alpha l - M
\end{bmatrix}
\begin{bmatrix}
\alpha l & \sqrt{\beta} K \\
-\sqrt{\beta} K & \alpha l
\end{bmatrix},
\] (17)
Obviously, we can regard the splitting matrix \( P(\alpha) \) given in (16) as the TBS preconditioner of \( A \) in (9), which is considered as the preconditioner derived from the TBS iteration method.

When preconditioning the Krylov subspace iteration method with the TBS preconditioner \( P(\alpha) \), it usually involves solving

\[
P(\alpha)z = r,
\]

where \( r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \in \mathbb{R}^{2n} \) represents the current residual vector and \( z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{C}^{2n} \) represents the generalized residual vector. We can get the following algorithm to calculate the vector \( z \) by using the structure of matrix \( P(\alpha) \).

In the implementation of Algorithm 2, by using the conjugate gradient method (CG) or the Cholesky decomposition method, we can solve the first three equations, of which the coefficient matrices are real symmetric positive definite matrices. Additionally, the fourth step only needs direct calculation without any additional solution steps. In the meantime, the second step uses \( c_2 \) obtained in the first step, and the fourth step uses \( z_1 \) from the third step, which greatly improves the calculation efficiency and shortens the calculation time of the preconditioned Krylov subspace method. This will also be demonstrated in numerical experiments.

**Algorithm 2**

Given \( r = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \in \mathbb{C}^{2n} \) you can use the following steps to solve \( z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \in \mathbb{C}^{2n} \)

1. Solve \( c_2 \) from

\[
(\alpha I + M)^2 + \beta \delta^2 M^2)c_2 = 2\alpha(\alpha I + M)r_2 - i2\alpha \sqrt{\beta} \delta M r_1.
\]

2. Solve \( c_1 \) from

\[
(\alpha I + M)c_1 = -i\sqrt{\beta} \delta M c_2 + 2\alpha r_1.
\]

3. Solve \( z_1 \) from

\[
(\alpha^2 I + \beta K^2)z_1 = \alpha c_1 + \sqrt{\beta} Kc_2.
\]

4. Compute \( z_2 = \frac{\psi - \sqrt{\beta} \delta c_1}{\beta} \).

Next, we use Theorem 3 to show the eigenvalues distribution of \( P^{-1}(\alpha)A \).

**Theorem 3.** Suppose that Theorem 1 is satisfied. Then the eigenvalue \( \lambda \) of \( P^{-1}(\alpha)A \) is distributed in \((0, 2)\), and when the eigenvalue of \( L(\alpha) \) is smaller, the eigenvalue of \( P^{-1}(\alpha)A \) is more clustered near 1.

**Proof.** As shown in (15), \( A = P(\alpha) - G(\alpha) \). We also know that \( L(\alpha) = P^{-1}(\alpha)G(\alpha) \). Then we can obtain

\[
P^{-1}(\alpha)A = P^{-1}(\alpha)(P(\alpha) - G(\alpha)) = I - P^{-1}(\alpha)G(\alpha).
\]

According to Theorem 1, \( \rho(L(\alpha)) < 1 \). Therefore \( 0 < \lambda < 2 \). \( \Box \)

4. Numerical Experiments

Some numerical experiments were used to illustrate the performance of TBS iterative method and TBS preconditioner to solve Equation (9). We used the iteration time in seconds and the number of iterative steps (i.e., CPUs and IT) to show the advantages of the TBS iteration method and preconditioner over other iteration methods and preconditioner.

The triangle and tetrahedron defined in two-dimensions and three-dimensions obtained by lowest Nedelec edge element \([19,20]\) were used to discretize variables. We used the MATLAB toolbox in [21] to establish the coefficient matrix. All the calculations were done in the computer with MATLAB R2018a. The computer was configured with Intel (R) Core (TM) i5-8250 U CPU (1.60 GHz, 4 G RAM).
Example 1 [21]. We consider the problem (1) and (2), where $\Omega = [0, 1] \times [0, 1] \in \mathbb{R}^n \times \mathbb{R}^n$ and $\sigma = 1, \nu = 1, \beta = 10^{-8}$. We divide the area $\Omega$ into two parts, $\Omega_1 = \{x \in \Omega | x_1 > x_2 \}$ and $\Omega_2 = \Omega \setminus \Omega_1$. $\tilde{y}_d(x)$ in (3) is defined as

$$
\tilde{y}_d(x)|_{\Omega_1} = \begin{bmatrix}
\sin(2\pi x_1) + 2\pi \cos(2\pi x_1)(x_2 - x_1) \\
\sin((x_2 - x_1)^2(x_1 - 1)^2x_2) - \sin(2\pi x_1)
\end{bmatrix}
$$

and $\tilde{y}_d(x)|_{\Omega_2} = 0$.

We first used Table 1 to show the relationship between the level of mesh refinement and the order of the matrices $M$ and $K$ in the 2D case.

| Table 1. Relationship between matrix order and level of mesh refinement (2D). |
|-----------------|-----------------|
| Level | Order of $M$ and $K$ |
| 1 | 56 |
| 2 | 208 |
| 3 | 800 |
| 4 | 3136 |
| 5 | 12,416 |

Tables 2 and 3 showed the calculation results of TBS method in this paper and BAS iteration methods in [11] when parameter $\alpha$ was the optimal parameter, and $\tilde{\omega}$, $\epsilon$ was different values under different mesh refinement levels. In the experiment, the inner iteration used the conjugate gradient method (CG), and all the initial vectors involved in the iteration method were $x^{(0)} = 0$, the outer iteration termination condition was $\|b - A x^{(k)}\|_2 / \|b\|_2 \leq 10^{-5}$, while the inner iteration termination condition was $\|b - A x^{(k)}\|_2 / \|M\|_2 \leq 10^{-6}$, where $x^{(k)}$ denoted the value of the $k$-th step iteration.

| Table 2. Different mesh refinement levels and $\tilde{\omega}$, the results of iteration method in the 2D case, with $\epsilon = 10^{-2}$. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Method | Level | $10^{-2}$ | $10^{-1}$ | $10^0$ | $10^1$ | $10^2$ |
| | IT | CPUs | IT | CPUs | IT | CPUs | IT | CPUs | IT | CPUs |
| TBS | 1 | 13 | 0.395 | 13 | 0.411 | 13 | 0.422 | 13 | 0.407 | 13 | 0.403 |
| | 2 | 13 | 0.406 | 13 | 0.420 | 13 | 0.409 | 13 | 0.424 | 13 | 0.409 |
| | 3 | 14 | 0.455 | 14 | 0.474 | 14 | 0.461 | 14 | 0.465 | 14 | 0.455 |
| | 4 | 14 | 0.705 | 14 | 0.714 | 14 | 0.718 | 14 | 0.722 | 14 | 0.719 |
| | 5 | 14 | 2.530 | 14 | 2.544 | 14 | 2.573 | 14 | 2.586 | 14 | 2.549 |
| BAS | 1 | 34 | 0.501 | 34 | 0.481 | 34 | 0.489 | 34 | 0.492 | 34 | 0.448 |
| | 2 | 34 | 0.488 | 34 | 0.503 | 34 | 0.478 | 34 | 0.497 | 34 | 0.499 |
| | 3 | 34 | 0.598 | 34 | 0.605 | 34 | 0.595 | 34 | 0.592 | 34 | 0.631 |
| | 4 | 34 | 1.127 | 34 | 1.110 | 34 | 1.097 | 34 | 1.110 | 34 | 1.136 |
| | 5 | 34 | 5.671 | 34 | 4.376 | 34 | 4.691 | 34 | 4.882 | 34 | 4.069 |

Tables 2 and 3 illustrated that, for the iteration steps, no matter the parameter changed or the matrix order changed, the steps of the TBS iterative method performed better, and with the increase of the order, the steps of the TBS method were stable and the increase was small. The operation time was also less than that of previous method. Even though the matrix order was more than 10,000, the iteration time of TBS was shorter, which demonstrated the advantages of the TBS iteration method in actual operation.

Tables 4 and 5 showed the operation results of preconditioned Generalized Minimal Residual(5) method (GMRES(5)) with $P_{TBS}$, $P_{BAS}$ in [12] and $P_{str-II}$ in [6] when the parameters $\alpha$ were the optimal parameters, different grid refinement level and $\tilde{\omega}$. In the actual calculation, the CG method was used as the internal iteration method. The initial vectors of
all the iteration methods involved were \( x^{(0)} = 0 \). The iteration termination condition was

\[
\frac{\| b - Ax^{(k)} \|_2}{\| b \|_2} \leq 10^{-6},
\]

where \( x^{(k)} \) was the value obtained by the \( k \)-th iteration. At the same time, we used the new marks OUT and IN to denote the number of external and internal iterative steps, respectively. In addition, CPUs represented the calculation times as before.

Table 3. Different mesh refinement levels and \( \bar{\omega} \), the results of iteration method in the 2D case, with \( \varepsilon = 10^{-4} \).

| Method | Level | \( 10^{-2} \) | \( 10^{-1} \) | \( 10^{0} \) | \( 10^{1} \) | \( 10^{2} \) |
|--------|-------|-------------|-------------|-------------|-------------|-------------|
|        |       | IT CPUs | IT CPUs | IT CPUs | IT CPUs | IT CPUs | IT CPUs |
| TBS    | 1     | 13      | 0.393     | 13         | 0.402      | 13         | 0.421      | 13         | 0.405      | 13         | 0.406      |
|        | 2     | 13      | 0.419     | 13         | 0.434      | 13         | 0.430      | 13         | 0.436      | 13         | 0.424      |
|        | 3     | 14      | 0.485     | 14         | 0.479      | 14         | 0.477      | 14         | 0.481      | 14         | 0.471      |
|        | 4     | 14      | 0.743     | 14         | 0.728      | 14         | 0.750      | 14         | 0.709      | 14         | 0.743      |
|        | 5     | 14      | 2.597     | 14         | 2.567      | 14         | 2.556      | 14         | 2.575      | 14         | 2.658      |
| BAS    | 1     | 34      | 0.502     | 34         | 0.473      | 34         | 0.492      | 34         | 0.486      | 34         | 0.498      |
|        | 2     | 34      | 0.520     | 34         | 0.521      | 34         | 0.528      | 34         | 0.513      | 34         | 0.495      |
|        | 3     | 34      | 0.622     | 34         | 0.622      | 34         | 0.624      | 34         | 0.627      | 34         | 0.613      |
|        | 4     | 34      | 1.147     | 34         | 1.153      | 34         | 1.143      | 34         | 1.197      | 34         | 1.134      |
|        | 5     | 34      | 4.759     | 34         | 4.113      | 34         | 4.180      | 34         | 4.740      | 34         | 4.741      |

Table 4. Different mesh refinement levels and \( \bar{\omega} \), the result of the preconditioned GMRES (5) in the 2D case, with \( \varepsilon = 10^{-2} \).

| Method | Level | \( 10^{-2} \) | \( 10^{-1} \) | \( 10^{0} \) | \( 10^{1} \) | \( 10^{2} \) |
|--------|-------|-------------|-------------|-------------|-------------|-------------|
|        |       | OUT IN     | CPUs OUT IN | CPUs OUT IN | CPUs OUT IN | CPUs OUT IN |
| \( P_{TBS} \) | 1     | 1 2      | 4 0.418    | 2 4         | 0.425      | 2 4         | 0.425      | 2 5         | 0.426      | 3 2         | 0.432      |
|        | 2     | 2 5      | 0.451 2    | 5 0.450 2   | 5 0.440 3   | 1 0.451 3   | 1 0.451 3   | 2 0.471     | 2 0.471     |     |
|        | 3     | 2 5      | 0.485 2    | 5 0.486 3   | 1 0.505 3   | 1 0.502 3   | 3 0.502 3   | 3 0.508     | 3 0.508     |     |
|        | 4     | 2 4      | 0.663 2    | 4 0.656 2   | 4 0.670 2   | 5 0.676 3   | 3 2         | 0.738       | 2 0.738     |     |
|        | 5     | 2 4      | 1.205 2    | 4 1.254 2   | 5 1.263 2   | 5 1.294 3   | 3 2         | 1.283       | 2 1.283     |     |
| \( P_{BAS} \) | 1     | 2 4      | 0.458 2    | 4 0.429 3   | 2 0.446 4   | 2 0.451 4   | 2 0.451 4   | 2 0.476     | 2 0.476     |     |
|        | 2     | 3 4      | 0.451 3    | 4 0.452 3   | 4 0.453 4   | 5 0.468 5   | 5 0.468 5   | 2 0.533     | 2 0.533     |     |
|        | 3     | 4 4      | 0.517 4    | 4 0.505 5   | 4 0.507 5   | 4 0.521 6   | 4 2         | 0.726       | 2 0.726     |     |
|        | 4     | 4 2      | 0.698 4    | 4 0.799 4   | 4 0.718 5   | 5 0.731 5   | 2 0.731 5   | 2 0.726     | 2 0.726     |     |
|        | 5     | 4 2      | 1.945 4    | 3 1.897 4   | 4 2.283 5   | 3 2.352 5   | 3 2         | 2.213       | 2 2.213     |     |
| \( P_{str-II} \) | 1     | 1 5      | 0.424 1    | 5 0.425 2   | 1 0.432 2   | 1 0.426 2   | 1 0.426 2   | 1 0.430     | 1 0.430     |     |
|        | 2     | 2 5      | 0.454 2    | 4 0.453 2   | 3 0.441 3   | 2 0.457 3   | 2 0.457 3   | 2 0.447     | 2 0.447     |     |
|        | 3     | 2 4      | 0.474 2    | 5 0.489 2   | 5 0.473 3   | 3 1         | 0.475 3     | 3 1         | 0.475 3     |     |
|        | 4     | 2 5      | 0.638 2    | 5 0.636 3   | 1 0.620 3   | 2 0.626 3   | 2 0.626 3   | 2 0.644     | 2 0.644     |     |
|        | 5     | 2 5      | 1.462 2    | 5 1.449 3   | 1 1.603 3   | 3 2         | 1.656 3     | 3 2         | 2.176       |     |

Tables 4 and 5 showed that, for the steps number, regardless of parameter changed or order changeed, \( P_{TBS} \) performed better than \( P_{BAS} \) and \( P_{str-II} \). In terms of iteration time (i.e., CPU), \( P_{TBS} \) had less preconditioning time than \( P_{BAS} \), and the CPU of \( P_{TBS} \) was similar to \( P_{str-II} \). Therefore, we believed that \( P_{TBS} \) had a certain improvement in the steps number and time and performs well.

Example 2 [21]. We consider the problem (1) and (2), where \( \Omega = [0, 1] \times [0, 1] \in \mathbb{R}^{n \times n} \) and \( \sigma = 1, \nu = 1, \beta = 10^{-8} \). We divide the area \( \Omega \) into two parts, \( \Omega_1 = \{ x \in \Omega | x_2 < \frac{1}{2} \} \) and \( \Omega_2 = \Omega \setminus \Omega_1 \). \( \hat{y}_d(x) \) in (3) is defined as

\[
\hat{y}_d(x) |_{\Omega_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \hat{y}_d(x) |_{\Omega_2} = 0.
\]
First, we used Table 6 to show the relationship between the level of mesh refinement and the order of matrix \( M \) and \( K \) in the 2D case.

### Table 5. Different mesh refinement levels and \( \omega \), the result of the preconditioned GMRES (5) in the 2D case, with \( \epsilon = 10^{-4} \).

| Method | Level | \( \omega \) |
|--------|-------|-------------|
|        | 10^{-2} | 10^{-1} | 10^0 | 10 | 10^2 |
|        | OUT | IN | CPUs | OUT | IN | CPUs | OUT | IN | CPUs | OUT | IN | CPUs |
| \( P_{TBS} \) | 1 | 2 | 4 | 0.421 | 2 | 4 | 0.432 | 2 | 4 | 0.430 | 2 | 5 | 0.426 | 3 | 2 | 0.435 |
| | 2 | 2 | 5 | 0.446 | 2 | 5 | 0.451 | 2 | 5 | 0.443 | 3 | 1 | 0.461 | 3 | 2 | 0.463 |
| | 3 | 2 | 5 | 0.475 | 2 | 5 | 0.470 | 3 | 1 | 0.484 | 3 | 1 | 0.494 | 3 | 3 | 0.505 |
| | 4 | 2 | 4 | 0.657 | 2 | 4 | 0.660 | 2 | 4 | 0.665 | 2 | 5 | 0.680 | 3 | 2 | 0.729 |
| | 5 | 2 | 4 | 1.263 | 2 | 4 | 1.204 | 2 | 5 | 1.237 | 2 | 3 | 1.282 | 3 | 2 | 1.276 |
| \( P_{BAS} \) | 1 | 2 | 4 | 0.439 | 2 | 4 | 0.424 | 3 | 2 | 0.427 | 4 | 2 | 0.444 | 4 | 2 | 0.452 |
| | 2 | 3 | 4 | 0.448 | 3 | 4 | 0.450 | 3 | 4 | 0.455 | 4 | 5 | 0.456 | 5 | 2 | 0.457 |
| | 3 | 4 | 4 | 0.501 | 4 | 4 | 0.598 | 5 | 2 | 0.514 | 5 | 4 | 0.515 | 6 | 2 | 0.539 |
| | 4 | 4 | 2 | 0.693 | 4 | 4 | 0.702 | 4 | 4 | 0.692 | 5 | 2 | 0.715 | 5 | 2 | 0.731 |
| | 5 | 4 | 2 | 1.810 | 4 | 3 | 1.924 | 4 | 4 | 1.977 | 5 | 3 | 2.227 | 5 | 2 | 2.176 |
| \( P_{str-II} \) | 1 | 1 | 5 | 0.422 | 1 | 5 | 0.423 | 2 | 1 | 0.413 | 2 | 1 | 0.429 | 2 | 1 | 0.427 |
| | 2 | 2 | 2 | 0.453 | 2 | 2 | 0.467 | 2 | 3 | 0.448 | 2 | 3 | 0.459 | 2 | 4 | 0.448 |
| | 3 | 2 | 4 | 0.452 | 2 | 5 | 0.489 | 2 | 5 | 0.499 | 3 | 1 | 0.460 | 3 | 1 | 0.477 |
| | 4 | 2 | 5 | 0.658 | 2 | 5 | 0.631 | 3 | 1 | 0.659 | 3 | 2 | 0.627 | 3 | 2 | 0.650 |
| | 5 | 2 | 5 | 1.448 | 2 | 5 | 1.511 | 3 | 1 | 1.593 | 3 | 2 | 1.677 | 3 | 2 | 1.650 |

### Table 6. The relationship between matrix order and mesh refinement level (2D).

| Level | Order of \( M \) and \( K \) |
|-------|----------------------------|
| 1     | 56 |
| 2     | 208 |
| 3     | 800 |
| 4     | 3136 |
| 5     | 12,416 |

Tables 7 and 8 showed the calculation results of the TBS and BAS iteration methods when the parameters \( \alpha \) were the optimal parameters of iteration method, under different mesh refinement levels and \( \omega \). The inner iteration method and iteration termination conditions were the same as Example 1.

### Table 7. Different mesh refinement levels and \( \omega \), the results of the iteration method in the 2D case, with \( \epsilon = 10^{-2} \).

| Method | Level | \( \omega \) |
|--------|-------|-------------|
|        | 10^{-2} | 10^{-1} | 10^0 | 10 | 10^2 |
|        | IT | CPUs | IT | CPUs | IT | CPUs | IT | CPUs | IT | CPUs |
| TBS    | 1 | 13 | 0.349 | 13 | 0.366 | 13 | 0.354 | 13 | 0.370 | 13 | 0.343 |
| | 2 | 13 | 0.362 | 13 | 0.392 | 13 | 0.388 | 13 | 0.399 | 13 | 0.395 |
| | 3 | 14 | 0.417 | 14 | 0.429 | 14 | 0.436 | 14 | 0.437 | 14 | 0.418 |
| | 4 | 14 | 0.590 | 14 | 0.569 | 14 | 0.597 | 14 | 0.612 | 14 | 0.606 |
| | 5 | 14 | 1.613 | 14 | 1.378 | 14 | 1.406 | 14 | 1.458 | 14 | 1.340 |
| BAS    | 1 | 34 | 0.415 | 34 | 0.433 | 34 | 0.420 | 34 | 0.411 | 34 | 0.435 |
| | 2 | 34 | 0.468 | 34 | 0.475 | 34 | 0.466 | 34 | 0.496 | 34 | 0.482 |
| | 3 | 34 | 0.569 | 34 | 0.563 | 34 | 0.584 | 34 | 0.559 | 34 | 0.565 |
| | 4 | 34 | 1.039 | 34 | 1.022 | 34 | 1.095 | 34 | 1.011 | 34 | 1.076 |
| | 5 | 34 | 2.901 | 34 | 2.929 | 34 | 4.345 | 34 | 2.864 | 34 | 4.014 |
Table 8. Different mesh refinement levels and \( \tilde{\omega} \), the results of the iteration method in the 2D case, with \( \epsilon = 10^{-4} \).

| Method | Level | \( 10^{-2} \) IT CPUs | \( 10^{-1} \) IT CPUs | \( 10^{0} \) IT CPUs | \( 10 \) IT CPUs | \( 10^{2} \) IT CPUs |
|--------|-------|------------------------|------------------------|------------------------|------------------------|------------------------|
| TBS    | 1     | 13                     | 0.368                  | 0.361                  | 0.363                  | 0.358                  | 0.358                  |
|        | 2     | 13                     | 0.387                  | 0.362                  | 0.369                  | 0.379                  | 0.373                  |
|        | 3     | 14                     | 0.418                  | 0.422                  | 0.423                  | 0.421                  | 0.434                  |
|        | 4     | 14                     | 0.592                  | 0.591                  | 0.597                  | 0.607                  | 0.583                  |
|        | 5     | 14                     | 1.396                  | 1.377                  | 1.376                  | 1.400                  | 1.451                  |
| BAS    | 1     | 34                     | 0.425                  | 0.418                  | 0.420                  | 0.415                  | 0.413                  |
|        | 2     | 34                     | 0.445                  | 0.454                  | 0.450                  | 0.437                  | 0.451                  |
|        | 3     | 34                     | 0.544                  | 0.558                  | 0.550                  | 0.556                  | 0.563                  |
|        | 4     | 34                     | 1.158                  | 1.029                  | 1.148                  | 1.360                  | 1.031                  |
|        | 5     | 34                     | 4.065                  | 4.183                  | 4.125                  | 3.963                  | 3.968                  |

Tables 7 and 8 showed that the selection of parameters and the change of order had little influence on the iteration steps of TBS. With the increase of order, the iteration steps of the iteration method did not increase significantly. For CPUs, whether the order was increased or not, the iteration time of the algorithm was less than that of BAS iteration method.

Tables 9 and 10 showed the operation results of preconditioned GMRES(5) with \( P_{TBS} \) in this paper, \( P_{BAS} \) in [12] and \( P_{str-II} \) in [6] when the parameters \( \alpha \) were the optimal parameters, different grid refinement level and \( \tilde{\omega} \), \( \epsilon \). The marks, inner iteration algorithm and iteration termination conditions in the following table were the same as those in Tables 4 and 5.

Table 9. Different mesh refinement levels and \( \tilde{\omega} \), the result of the preconditioned GMRES (5) in the 2D case, with \( \epsilon = 10^{-2} \).

| Method | Level | \( 10^{-2} \) OUT IN CPUs | \( 10^{-1} \) OUT IN CPUs | \( 10^{0} \) OUT IN CPUs | \( 10 \) OUT IN CPUs | \( 10^{2} \) OUT IN CPUs |
|--------|-------|---------------------------|---------------------------|---------------------------|------------------------|------------------------|
| \( P_{TBS} \) | 1     | 2                         | 0.374                    | 0.381                    | 0.378                  | 2                     | 0.371                  | 3                     | 2                     | 0.377                  |
|        | 2     | 2                         | 0.382                    | 0.383                    | 0.389                  | 2                     | 0.393                  | 3                     | 2                     | 0.391                  |
|        | 3     | 2                         | 0.430                    | 0.439                    | 0.431                  | 2                     | 0.430                  | 3                     | 2                     | 0.454                  |
|        | 4     | 2                         | 0.552                    | 0.565                    | 0.556                  | 2                     | 0.571                  | 3                     | 1                     | 0.554                  |
|        | 5     | 2                         | 1.120                    | 1.128                    | 1.120                  | 2                     | 1.425                  | 3                     | 1                     | 1.327                  |
| \( P_{BAS} \) | 1     | 2                         | 0.383                    | 0.380                    | 0.382                  | 2                     | 0.386                  | 2                     | 5                     | 0.387                  |
|        | 2     | 2                         | 0.400                    | 0.416                    | 0.449                  | 3                     | 0.403                  | 3                     | 2                     | 0.411                  |
|        | 3     | 2                         | 0.593                    | 0.574                    | 0.594                  | 3                     | 0.571                  | 4                     | 2                     | 0.545                  |
|        | 4     | 3                         | 0.673                    | 0.683                    | 0.697                  | 4                     | 0.691                  | 5                     | 2                     | 0.714                  |
|        | 5     | 4                         | 1.250                    | 1.430                    | 1.207                  | 4                     | 1.399                  | 5                     | 4                     | 1.422                  |
| \( P_{str-II} \) | 1     | 1                         | 0.496                    | 0.492                    | 0.494                  | 1                     | 0.497                  | 1                     | 2                     | 0.479                  |
|        | 2     | 1                         | 0.509                    | 0.505                    | 0.514                  | 1                     | 0.516                  | 1                     | 3                     | 0.535                  |
|        | 3     | 1                         | 0.544                    | 0.526                    | 0.549                  | 1                     | 0.553                  | 1                     | 4                     | 0.554                  |
|        | 4     | 1                         | 0.880                    | 0.918                    | 0.870                  | 1                     | 0.874                  | 2                     | 1                     | 0.951                  |
|        | 5     | 2                         | 1.318                    | 1.447                    | 1.443                  | 2                     | 1.432                  | 2                     | 2                     | 1.504                  |
Table 10. Different mesh refinement levels and $\tilde{\omega}$, the result of the preconditioned GMRES (5) in the 2D case, with $\varepsilon = 10^{-4}$.

| Method | Level | $10^{-2}$ | $10^{-1}$ | $10^{0}$ | $10$ | $10^{2}$ |
|--------|-------|-----------|-----------|---------|------|--------|
|        |       | OUT       | IN        | CPUs    | OUT  | IN     | CPUs    | OUT   | IN   | CPUs   |
| $P_{TBS}$ | 1     | 2         | 4         | 0.380   | 2     | 4      | 0.385   | 2     | 4    | 0.378  | 3     | 2     | 0.382  |
|         | 2     | 2         | 4         | 0.396   | 2     | 4      | 0.390   | 2     | 4    | 0.388  | 3     | 2     | 0.411  |
|         | 3     | 2         | 4         | 0.427   | 2     | 4      | 0.432   | 2     | 4    | 0.438  | 3     | 2     | 0.445  |
|         | 4     | 2         | 4         | 0.554   | 2     | 4      | 0.544   | 2     | 4    | 0.538  | 5     | 1     | 0.587  |
|         | 5     | 2         | 4         | 1.144   | 2     | 4      | 1.156   | 2     | 4    | 1.173  | 5     | 1.193 | 3      | 1.331 |
| $P_{BAS}$ | 1     | 2         | 2         | 0.407   | 2     | 2      | 0.412   | 2     | 4    | 0.407  | 4     | 0.405 | 2      | 0.416 |
|         | 2     | 2         | 2         | 0.423   | 2     | 2      | 0.429   | 3     | 2    | 0.443  | 3     | 2     | 0.424  | 2      | 0.425 |
|         | 3     | 2         | 4         | 0.494   | 2     | 5      | 0.460   | 3     | 2    | 0.468  | 3     | 2     | 0.463  | 4      | 0.483 |
|         | 4     | 3         | 4         | 0.655   | 3     | 4      | 0.603   | 4     | 2    | 0.641  | 4     | 0.644 | 5      | 0.673 |
|         | 5     | 4         | 2         | 1.639   | 4     | 2      | 1.646   | 4     | 4    | 1.673  | 5     | 4     | 1.473  | 5      | 1.502 |
| $P_{str-II}$ | 1     | 1         | 2         | 0.396   | 1     | 2      | 0.403   | 1     | 2    | 0.398  | 1     | 2     | 0.404  | 1      | 2      | 0.393 |
|          | 2     | 1         | 2         | 0.409   | 1     | 2      | 0.408   | 1     | 2    | 0.411  | 1     | 2     | 0.417  | 1      | 3      | 0.411 |
|          | 3     | 1         | 3         | 0.424   | 1     | 3      | 0.422   | 1     | 3    | 0.416  | 3     | 3     | 0.427  | 1      | 4      | 0.410 |
|          | 4     | 1         | 5         | 0.580   | 1     | 5      | 0.581   | 1     | 5    | 0.591  | 5     | 5     | 0.576  | 2      | 1      | 0.580 |
|          | 5     | 2         | 1         | 1.530   | 2     | 1      | 1.519   | 2     | 1    | 1.543  | 2     | 1     | 0.546  | 2      | 2      | 1.581 |

From Tables 9 and 10 we can see that, compared with $P_{BAS}$ and $P_{str-II}$, $P_{TBS}$ had obvious improvement in the iteration number. For the running time (i.e., CPU), with the increase of the order of the matrices $M$ and $K$, CPU of $P_{TBS}$ was larger than that of $P_{str-II}$ and $P_{BAS}$.

5. Conclusions

We mainly solve the time-period eddy current optimization problems (1) and (2) in this paper. The method previously put forward for the generalized saddle point problem is not fully applicable to this method, since the coefficient matrix of the equation is a complex matrix, $t$. Based on this, this paper proposes a new split iteration method (i.e., the TBS iteration method) and a new preconditioner (i.e., the TBS preconditioner) to solve Equation (9). The iteration method is unconditionally convergent, and the optimal parameter which makes the upper bound of the spectral radius of the iteration matrix minimum is obtained. Numerical experiments demonstrate that the TBS method is more feasible and efficient than the previous one. At the same time, the iteration time and steps of the preconditioned GMRES (5) iteration method with the TBS preconditioner also have a significant improvement.

Author Contributions: Formal analysis, Y.-R.L.; funding acquisition, X.-H.S.; investigation, S.-Y.L.; project administration, X.-H.S.; software, Y.-R.I.; supervision, X.-H.S.; validation, S.-Y.L.; writing—original draft, Y.-R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Liaoning Province, grant number 20170540323 and the Central University Basic Scientific Research Business Expenses Special Funds, grant number N2005013.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.
References

1. Lions, J.L. Optimal Control of Systems Governed by Partial Differential Equations; Springer: Berlin/Heidelberg, Germany, 1971; ISBN 3340051155.
2. Hinze, M.; Pinnau, R.; Ulbrich, M.; Ulbrich, S. Optimization with PDE Constraints; Springer: Dordrecht, The Netherlands, 2009; ISBN 9781402088384.
3. Ida, N. Numerical Modeling for Electromagnetic Non-Destructive Evaluation. Nondestruct. Test. Eval. 1996, 12, 283. [CrossRef]
4. Gunzburger, M.; Trenchea, C. Optimal control of the time-periodic MHD equations. Nonlinear Anal. Theory Methods Appl. 2005, 63. [CrossRef]
5. Kolmbauer, M.; Langer, U. A robust preconditioned minres solver for distributed time-periodic eddy current optimal control problems. SIAM J. Sci. Comput. 2012, 34. [CrossRef]
6. Axelsson, O.; Liang, Z.Z. A note on preconditioning methods for time-periodic eddy current optimal control problems. J. Comput. Appl. Math. 2019, 352, 262–277. [CrossRef]
7. Rees, T.; Dollar, H.S.; Wathen, A.J. Optimal solvers for PDE-constrained optimization. SIAM J. Sci. Comput. 2010, 32, 271–298. [CrossRef]
8. Bai, Z.-Z. Block preconditioners for elliptic PDE-constrained optimization problems. Computing 2011, 91, 379–395. [CrossRef]
9. Zhang, G.; Zheng, Z. Block-symmetric and block-lower-triangular preconditioners for PDE-constrained optimization problems. J. Comput. Math. 2013, 31, 370–381. [CrossRef]
10. Bai, Z.Z.; Golub, G.H.; Ng, M.K. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. 2003, 24, 603–626. [CrossRef]
11. Bai, Z.Z.; Benzi, M.; Chen, F.; Wang, Z.Q. Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems. IMA J. Numer. Anal. 2013, 33, 343–369. [CrossRef]
12. Zheng, Z.; Zhang, G.F.; Zhu, M.Z. A block alternating splitting iteration method for a class of block two-by-two complex linear systems. J. Comput. Appl. Math. 2015, 288, 203–214. [CrossRef]
13. Krendl, W.; Simoncini, V.; Zulehner, W. Stability estimates and structural spectral properties of saddle point problems. Numer. Math. 2013, 124, 183–213. [CrossRef]
14. Axelsson, O.; Neytcheva, M.; Ahmad, B. A comparison of iterative methods to solve complex valued linear algebraic systems. Numer. Algorithms 2014, 66, 811–841. [CrossRef]
15. Axelsson, O.; Kucherov, A. Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra with Appl. 2000, 7, 197–218. [CrossRef]
16. Axelsson, O.; Farouq, S.; Neytcheva, M. Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems: Poisson and convection-diffusion control. Numer. Algorithms 2016, 73, 631–663. [CrossRef]
17. Zheng, Z.; Zhang, G.F.; Zhu, M.Z. A note on preconditioners for complex linear systems arising from PDE-constrained optimization problems. Appl. Math. Lett. 2016, 61, 114–121. [CrossRef]
18. Liang, Z.Z.; Axelsson, O.; Neytcheva, M. A robust structured preconditioner for time-harmonic parabolic optimal control problems. Numer. Algorithms 2018, 79, 575–596. [CrossRef]
19. Nédélec, J.C. A new family of mixed finite elements in R3. Numer. Math. 1986, 50, 57–81. [CrossRef]
20. Nédélec, J.C. Mixed finite elements in R3. Numer. Math. 1980, 35, 315–341. [CrossRef]
21. Anjam, I.; Valdman, J. Fast MATLAB assembly of FEM matrices in 2D and 3D: Edge elements. Appl. Math. Comput. 2015, 267, 252–263. [CrossRef]