Jun Li; Xiangtuan Xiong

A new block triangular preconditioner for three-by-three block saddle-point problem

Applications of Mathematics, Vol. 69 (2024), No. 1, 67–91

Persistent URL: http://dml.cz/dmlcz/152253

69 (2024) APPLICATIONS OF MATHEMATICS No. 1, 67–91

A NEW BLOCK TRIANGULAR PRECONDITIONER FOR THREE-BY-THREE BLOCK SADDLE-POINT PROBLEM

Jun Li, Xiangtuan Xiong, Lanzhou

Received December 9, 2022. Published online December 7, 2023.

Abstract. In this paper, to solve the three-by-three block saddle-point problem, a new block triangular (NBT) preconditioner is established, which can effectively avoid the solving difficulty that the coefficient matrices of linear subsystems are Schur complement matrices when the block preconditioner is applied to the Krylov subspace method. Theoretical analysis shows that the iteration method produced by the NBT preconditioner is unconditionally convergent. Besides, some spectral properties are also discussed. Finally, numerical experiments are provided to show the effectiveness of the NBT preconditioner.

Keywords: three-by-three block saddle-point problems; matrix splitting; convergence; preconditioning, GMRES method

MSC 2020: 65F08, 65F10, 65F50

1. Introduction

In recent years, lots of work have been devoted to the problems of solving large linear systems with the three-by-three block saddle-point structure, which arises in many plentiful backgrounds in scientific computing and practical applications, for example, solving the quadratic program [8],

$$\min\Bigl\{\frac{1}{2}x^\top Ax + r^\top x + q^\top y\Bigr\}, \quad \text{s.t. } Bx + C^\top y = b, \quad x \in \mathbb{R}^n,\ y \in \mathbb{R}^l,$$

where $r \in \mathbb{R}^n$ and $q \in \mathbb{R}^l$ are given vectors; solving the Maxwell equations in a dielectric medium [6], [16],

$$\varepsilon\partial_t\mathbf{E} - \operatorname{curl}\mathbf{H} = \mathbf{J} \quad \text{in } \Omega \times (0, T),$$

This work was supported by the Foundation for Distinguished Young Scholars of Gansu Province (No. 20JR5RA540) and the National Natural Science Foundation of China (No. 11661072).

DOI: 10.21136/AM.2023.0289-22

$$
\begin{aligned}
\mu\partial_t \mathbf{H} + \operatorname{curl}\mathbf{E} &= 0 \quad \text{in } \Omega \times (0,T),\\
\operatorname{div}(\varepsilon\mathbf{E}) &= \varrho \quad \text{in } \Omega \times (0,T),\\
\operatorname{div}(\mu\mathbf{H}) &= 0 \quad \text{in } \Omega \times (0,T).
\end{aligned}
$$

Here $\Omega \subset R^3$ is a simply-connected Lipschitz polyhedral domain with connected boundary which is occupied by dielectric material, $\mathbf{E}$ and $\mathbf{H}$ are the electric and magnetic fields, and $\mathbf{J}$ and ϱ are the current density and charge density. We assume that the permeability parameter μ and the permittivity parameter ε of medium are discontinuous across an interface $\Gamma \subset \Omega$, where Γ is the boundary of a simply-connected Lipschitz polyhedral domain Ω_1 with $\overline{\Omega}_1 \subset \Omega$ and $\Omega_2 = \Omega \setminus \Omega_1$. We use the Picard iteration method for a class of mixed finite element schemes for stationary magnetohydrodynamics models [9],

$$
\begin{aligned}
(\mathbf{u}\cdot\nabla)\mathbf{u} - R_e^{-1}\Delta\mathbf{u} - S\mathbf{j}\times\mathbf{B} + \nabla p &= \mathbf{f},\\
\mathbf{j} - R_m^{-1}\nabla\times\mathbf{B} &= \mathbf{0},\\
\nabla\times\mathbf{E} &= \mathbf{0},\\
\nabla\cdot\mathbf{B} &= 0,\\
\nabla\cdot\mathbf{u} &= 0,
\end{aligned}
$$

where Ohm's law

$$\mathbf{j} = E + \mathbf{u}\times\mathbf{B}$$

holds. Here $\mathbf{u}$ is the velocity of conducting fluids, p is the pressure, $\mathbf{B}$ is the magnetic field, $\mathbf{E}$ is the electric field and $\mathbf{j}$ is the volume current density. Dimensionless parameters R_e, R_m and S are the Reynolds number of fluids, magnetic field and the coupling number, respectively.

All the above tasks mainly involve the iteration solution of the following form of a large sparse three-by-three saddle-point problem

$$
\hat{\mathcal{A}}\mathrm{u} = \begin{pmatrix} A & B^\top & 0\\ B & 0 & C^\top\\ 0 & C & 0 \end{pmatrix}\begin{pmatrix} x\\ y\\ z\end{pmatrix} = \begin{pmatrix} f\\ g\\ h\end{pmatrix} \equiv \hat{b}, \tag{1.1}
$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite (SPD), $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{p\times m}$ are of full row rank. Here, $f \in \mathbb{R}^n$, $g \in \mathbb{R}^m$ and $h \in \mathbb{R}^p$ are given vectors, $(\cdot)^\top$ denotes the transpose of the corresponding matrix. Under these conditions, the solution of the linear system (1.1) is unique [11], [10].

It is obvious that the coefficient matrix of the linear system (1.1) is symmetric, then solving it by applying classical iteration solution methods may not be advantage.

One usually equivalently reformulates (1.1) into

$$\mathcal{A}\mathrm{u} = \begin{pmatrix} A & B^\top & 0 \\ -B & 0 & -C^\top \\ 0 & C & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f \\ -g \\ h \end{pmatrix} \equiv b, \tag{1.2}$$

which forms the possibility to apply classical iteration solution methods, such as stationary iteration methods and Krylov subspace methods, of which the latter is particularly popular. Because of the slow convergence or even nonconvergence of the nonpreconditioned Krylov subspace methods in solving the linear system (1.2), it is necessary to construct a suitable preconditioner to be used in combination with Krylov subspace methods, which can avoid nonconvergence and make Krylov subspace methods achieve satisfactory convergence results quickly. Huang et al. [12] proposed an exact block diagonal (BD) preconditioner for solving the nonsingular system (1.2)

$$P_{BD} = \begin{pmatrix} A & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & CS^{-1}C^\top \end{pmatrix}, \tag{1.3}$$

where $S = BA^{-1}B^\top$. The inexact versions of the preconditioner were also studied because of the complexity of solving the residual equation of the exact preconditioner. Then, Cao [4] established the shift splitting (SS) iteration method and the SS and relaxed SS (RSS) preconditioners were also derived, the form of the former is

$$P_{SS} = \frac{1}{2} \begin{pmatrix} \alpha I + A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & C & \alpha I \end{pmatrix}. \tag{1.4}$$

Based on the SS preconditioner, a generalized shift-splitting (GSS) preconditioner was considered in [19]. Zhang et al. [23] proposed a lopsided shift-splitting preconditioner for the saddle-point problem (1.2). Xie et al. [21] considered three efficient preconditioners in order to improve the preconditioning effect of the BD preconditioner,

$$\begin{aligned} P_1 &= \begin{pmatrix} A & 0 & 0 \\ B & -S & C^\top \\ 0 & 0 & -CS^{-1}C^\top \end{pmatrix}, \quad P_2 = \begin{pmatrix} A & 0 & 0 \\ B & -S & C^\top \\ 0 & 0 & CS^{-1}C^\top \end{pmatrix}, \\ P_3 &= \begin{pmatrix} A & B^\top & 0 \\ B & -S & 0 \\ 0 & 0 & -CS^{-1}C^\top \end{pmatrix}. \end{aligned} \tag{1.5}$$

Aslani et al. [3] presented a new matrix splitting and deduced a block preconditioner for solving the linear system (1.2)

$$P_4 = \begin{pmatrix} A & B^\top & 0 \\ 0 & D & -C^\top \\ 0 & C & 0 \end{pmatrix}, \tag{1.6}$$

where D is an SPD matrix. Salkuyeh et al. in [18] considered the alternating positive semi-definite splitting (APSS) iteration method and the APSS preconditioner was also produced for solving the problem (1.2),

$$P_{APSS} = \begin{pmatrix} \alpha I + A & B^\top & 0 \\ -B & \alpha I & 0 \\ 0 & 0 & \alpha I \end{pmatrix} \begin{pmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I & -C^\top \\ 0 & C & \alpha I \end{pmatrix}. \tag{1.7}$$

Furthermore, the APSS method was also considered for solving three-by-three large, sparse, and singular saddle-point problems [2]. Abdolmaleki et al. [1] from another considerations proposed a block three-by-three diagonal preconditioner for the three-by-three block saddle-point problem (1.2). Wang et al. [20] established an exact parameterized block SPD preconditioner, and its inexact version for the block three-by-three saddle-point problem (1.2) was also considered.

In [24], based on the existing block preconditioner, authors proposed a two-parameter block triangular preconditioner for double saddle-point problem. Inspired by the idea in [24], we establish a new block triangular (NBT) preconditioner for the linear system (1.2) on the basis of the matrix decomposition of the coefficient matrix $\mathcal{A}$. Our purpose is to improve the preconditioning effect of the existing block preconditioners by avoiding solving Schur complement matrices in each iteration. Theoretical analysis shows that the iteration method produced by the NBT preconditioner is unconditionally convergent. The choices of parameters for the NBT preconditioner are also discussed. Additionally, we examine the spectral properties of the NBT preconditioned matrix. Two numerical experiments further confirm the proposed preconditioner is efficient and feasible.

The framework of this paper can be arranged as follows. We establish a new block triangular (NBT) preconditioner, and the iteration scheme produced by the preconditioner with unconditional convergence is discussed in Section 2. In Section 3, we discuss the spectral properties of the NBT preconditioned matrix. Then, in Section 4, numerical experiments are provided to demonstrate the efficiency of the proposed preconditioner. Finally, conclusions are given in Section 5.

2. A new block triangular preconditioner

In this section, we introduce a new block triangular (NBT) preconditioner. For the linear system (1.2), we decompose the three-by-three block saddle-point coefficient matrix $\mathcal{A}$ into

$$\mathcal{A} = \begin{pmatrix} I & 0 & 0 \\ -BA^{-1} & I & 0 \\ 0 & CS^{-1} & CS^{-1}C^\top \end{pmatrix} \begin{pmatrix} A & B^\top & 0 \\ 0 & S & -C^\top \\ 0 & 0 & I \end{pmatrix}, \tag{2.1}$$

where $S = BA^{-1}B^\top$ and I denotes the identity matrix with suitable dimension. We find from the above matrix decomposition and existing block preconditioners that, when we use block preconditioners that are close to the coefficient matrix $\mathcal{A}$ to solve the linear system (1.2), we need to solve linear subsystems with the coefficient matrix A and Schur complementary matrices $BA^{-1}B^\top$ and $CS^{-1}C^\top$ in each iteration, such as the preconditioners P_1, P_2 and P_3 in (1.5), which is very time-consuming. To reduce CPU time, solvers can choose to use their inexact versions to calculate the linear system, but this will face a new problem, that is, the relative error between the iteration solution and the exact solution will increase. In order to improve the computational efficiency of the block preconditioners, the strategy we can adopt is to replace the Schur complementary matrix with an easily solved matrix. Combined with the thought of [24], we establish a new block triangular preconditioner based on (2.1),

$$\begin{aligned} P_{NBT} &= \begin{pmatrix} I & 0 & 0 \\ -BA^{-1} & I & 0 \\ 0 & 0 & \beta I + \frac{1}{\alpha}CC^\top \end{pmatrix} \begin{pmatrix} A & B^\top & 0 \\ 0 & \alpha I + S & -C^\top \\ 0 & 0 & I \end{pmatrix} \\ &= \begin{pmatrix} A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & 0 & \beta I + \frac{1}{\alpha}CC^\top \end{pmatrix}, \end{aligned} \tag{2.2}$$

where α and β are two positive parameters. This can avoid solving the dense Schur complement matrices in each iteration.

It is easy to observe that the coefficient matrix $\mathcal{A}$ admits the matrix splitting

$$\mathcal{A} = \begin{pmatrix} A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & 0 & \beta I + \frac{1}{\alpha}CC^\top \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & \alpha I & 0 \\ 0 & -C & \beta I + \frac{1}{\alpha}CC^\top \end{pmatrix} = P_{NBT} - Q_{NBT}. \tag{2.3}$$

By the splitting (2.3), the NBT iteration method can be naturally established.

NBT iteration method. Let α and β be positive parameters. Given an initial guess $u^{(0)} = (x^{(0)^\top}, y^{(0)^\top}, z^{(0)^\top})^\top \in \mathbb{R}^{m+n+p}$ for $k = 0, 1, 2, \ldots$ until the iteration sequence $u^{(k)} = (x^{(k)^\top}, y^{(k)^\top}, z^{(k)^\top})^\top \in \mathbb{R}^{m+n+p}$ converges, compute

$$
\begin{aligned}
&\begin{pmatrix} A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & 0 & \beta I + \dfrac{1}{\alpha}CC^\top \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ z^{(k+1)} \end{pmatrix} \\
&\qquad = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \alpha I & 0 \\ 0 & -C & \beta I + \dfrac{1}{\alpha}CC^\top \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ z^{(k)} \end{pmatrix} + \begin{pmatrix} f \\ -g \\ h \end{pmatrix}.
\end{aligned} \tag{2.4}
$$

The iteration matrix of the NBT iteration method is

$$
\mathcal{T}(\alpha, \beta) = \begin{pmatrix} A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & 0 & \beta I + \dfrac{1}{\alpha}CC^\top \end{pmatrix}^{-1} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \alpha I & 0 \\ 0 & -C & \beta I + \dfrac{1}{\alpha}CC^\top \end{pmatrix}. \tag{2.5}
$$

Let $\varrho(\mathcal{T}(\alpha, \beta))$ denote the spectral radius of the iteration matrix $\mathcal{T}(\alpha, \beta)$. Then the NBT iteration method is convergent if and only if $\varrho(\mathcal{T}(\alpha, \beta)) < 1$ [4], [18].

In fact, for a given matrix splitting $\mathcal{A} = M - N$ with M being nonsingular, we can construct a splitting iteration method. Besides, matrix M can naturally be regarded as the preconditioner for Krylov subspace methods. The splitting preconditioner corresponding to the NBT iteration method (2.4) is P_{NBT}.

When the preconditioner P_{NBT} is used to accelerate Krylov subspace methods (such as the GMRES method), the generalized residual equation

$$
P_{NBT} z = r \tag{2.6}
$$

needs to be solved, where $r = (r_1; r_2; r_3)$ and $z = (z_1; z_2; z_3)$ are the given residual vector and current vector, respectively, and $r_1, z_1 \in \mathbb{R}^n$, $r_2, z_2 \in \mathbb{R}^m$, $r_3, z_3 \in \mathbb{R}^p$. Since

$$
\begin{aligned}
P_{NBT} = &\begin{pmatrix} I & 0 & 0 \\ 0 & I & -C^\top \\ 0 & 0 & \beta I + \dfrac{1}{\alpha}CC^\top \end{pmatrix} \begin{pmatrix} I & \dfrac{1}{\alpha}B^\top & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \\
&\times \begin{pmatrix} A + \dfrac{1}{\alpha}B^\top B & 0 & 0 \\ 0 & \alpha I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ -\dfrac{1}{\alpha}B & I & 0 \\ 0 & 0 & I \end{pmatrix},
\end{aligned}
$$

we have the following algorithmic implementation to solve the linear system (2.6) in actual computation.

NBT algorithm. For a given residual vector $r = [r_1^\top, r_2^\top, r_3^\top]^\top$, the current vector $z = [z_1^\top, z_2^\top, z_3^\top]^\top$ of (2.6) can be computed according to the following procedures:

(1) Solve $(\beta I + \alpha^{-1}CC^\top)z_3 = r_3$.
(2) Solve $(A + \alpha^{-1}B^\top B)z_1 = r_1 - \alpha^{-1}B^\top r_2 - \alpha^{-1}B^\top C^\top z_3$.
(3) Solve $z_2 = \alpha^{-1}(r_2 + C^\top z_3 + Bz_1)$.

In the above algorithm, the linear systems with coefficient matrices $\beta I + \alpha^{-1}CC^\top$ and $A + \alpha^{-1}B^\top B$ have to be solved at each iteration. They are easier to solve than linear subsystems with the Schur complement matrices $BA^{-1}B^\top$ and $C(BA^{-1}B^\top)^{-1}C^\top$. Note that both $\beta I + \alpha^{-1}CC^\top$ and $A + \alpha^{-1}B^\top B$ are symmetric positive definite for all $\alpha > 0$, $\beta > 0$. We can employ the sparse Cholesky factorization to solve them.

Remark 2.1. Note that we need to select suitable parameters when the NBT preconditioner is used to solve the linear system (1.2). There are many practical methods to calculate optimal parameters in [5], [13], [14], [15]. Using the method in [5], [13], we see that the NBT preconditioned GMRES method will reach the best when the parameters α and β minimize the function:

$$\Phi(\alpha, \beta) = \|Q_{NBT}\|_F^2,$$

where Q_{NBT} is defined in (2.3) and $\|\cdot\|_F$ represents the Frobenius norm. By computation, we have

$$\begin{aligned}\Phi(\alpha, \beta) &= \|Q_{NBT}\|_F^2 = \operatorname{tr}(Q_{NBT}^\top Q_{NBT})\\ &= m\alpha^2 + \frac{2\beta}{\alpha}\|C\|_F^2 + \frac{1}{\alpha^2}\|CC^\top\|_F^2 + p\beta^2 + \|C\|_F^2,\end{aligned}$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. By taking the first-order derivative of $\Phi(\alpha, \beta)$ and making use of the necessary condition for the extreme value of a function, we obtain

$$\begin{aligned}\Phi_\alpha &= 2m\alpha - \frac{2\beta}{\alpha^2}\|C\|_F^2 - \frac{2}{\alpha^3}\|CC^\top\|_F^2 = 0,\\ \Phi_\beta &= 2p\beta + \frac{2}{\alpha}\|C\|_F^2 = 0.\end{aligned}$$

Thus it follows that $\beta = -(p\alpha)^{-1}\|C\|_F^2$ and α satisfies the equation

$$m\alpha^4 - \beta\|C\|_F^2\alpha - \|CC^\top\|_F^2 = 0.$$

In fact, the parameter β is NOT a negative constant, we can choose β as close to zero as possible. The corresponding parameter α can be calculated by the above quarto equation. (It needs to be explained that the negative and complex roots of the quarto equation will be omitted.)

Next, we analyze the convergence of the NBT iteration method (2.4). Firstly, we give a useful lemma.

Lemma 2.1 ([22]). *Both roots of the real quadratic equation $x^2 - px + q = 0$ are less than 1 in modulus if and only if $|q| < 1$ and $|p| < 1 + q$.*

Theorem 2.1. *Assume that $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{p\times m}$ be of full row rank. Let α and β be the positive constants. Then the NBT iteration method (2.4) is unconditionally convergent.*

Proof. Let λ be an eigenvalue of the iteration matrix $\mathcal{T}(\alpha, \beta)$ and $l = [u; v; w]$ be the corresponding eigenvector. If $\lambda = 0$, the NBT iteration method is convergent. Next, we only consider the case of $\lambda \neq 0$ corresponding to $l \neq 0$. It follows according to the relationships between eigenvalues and eigenvectors that

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \alpha I & 0 \\ 0 & -C & \beta I + \frac{1}{\alpha}CC^\top \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \lambda \begin{pmatrix} A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & 0 & \beta I + \frac{1}{\alpha}CC^\top \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}. \tag{2.7}$$

The equation (2.7) can equivalently be reformulated into

$$\begin{cases} \lambda(Au + B^\top v) = 0, \\ \lambda Bu + \alpha(1-\lambda)v + \lambda C^\top w = 0, \\ Cv + (\lambda - 1)\Big(\beta I + \frac{1}{\alpha}CC^\top\Big)w = 0. \end{cases} \tag{2.8}$$

If $\lambda = 1$, (2.8) reduces to

$$\begin{cases} Au + B^\top v = 0, \\ Bu + C^\top w = 0, \\ Cv = 0. \end{cases} \tag{2.9}$$

Note that both u and v are not zero vectors, otherwise $(u; v; w) = (0; 0; 0)$, which is a contradiction. Multiplying the second equation in (2.9) from left by v^*, and combining it with the first and third equations in (2.9) gives $u^*Au = 0$, i.e., $u = 0$, furthermore, it follows that $v = 0$ and $w = 0$, contradiction happens. Thus $\lambda \neq 1$.

Next, let's consider the case that λ is not equal to 1. Based on this assumption, it is easy to find that both u and v are nonzero vectors. We have from the third equation of (2.8) that

$$w = \frac{1}{1-\lambda}\Big(\beta I + \frac{1}{\alpha}CC^\top\Big)^{-1}Cv.$$

Taking it into the second equation of (2.8) gives

$$\lambda Bu + \alpha(1-\lambda)v + \frac{\lambda}{1-\lambda}C^\top\Big(\beta I + \frac{1}{\alpha}CC^\top\Big)^{-1}Cv = 0. \tag{2.10}$$

The first equation of (2.8) gives $u = -A^{-1}B^\top v$. Inserting it into (2.10) and multiplying from left by $v^*/(v^*v)$ it becomes

$$-\lambda\frac{v^*BA^{-1}B^\top v}{v^*v} + \alpha(1-\lambda) + \frac{\lambda}{1-\lambda}\frac{v^*C^\top(\beta I + \alpha^{-1}CC^\top)^{-1}Cv}{v^*v} = 0. \tag{2.11}$$

Through the assumed conditions, we put

$$\frac{v^*BA^{-1}B^\top v}{v^*v} = b > 0; \quad \frac{v^*C^\top(\beta I + \alpha^{-1}CC^\top)^{-1}Cv}{v^*v} = c \geqslant 0.$$

Therefore, (2.11) can be written as

$$\lambda^2 - \frac{2\alpha + b - c}{\alpha + b}\lambda + \frac{\alpha}{\alpha + b} = 0. \tag{2.12}$$

It is easy to verify that $|\alpha/(\alpha + b)| < 1$ and

$$\Big|\frac{2\alpha + b - c}{\alpha + b}\Big| = \Big|1 + \frac{\alpha - c}{\alpha + b}\Big| \leqslant 1 + \frac{\alpha}{\alpha + b}.$$

By Lemma 2, we know that $|\lambda| < 1$. This completes the proof. □

3. Spectral properties of the preconditioned matrix $P_{NBT}^{-1}\mathcal{A}$

In this section, we analyze the spectral properties of the preconditioned matrix under the preconditioner P_{NBT}.

Firstly, we have the following lemma.

Lemma 3.1. *Assume that $A \in \mathbb{R}^{n\times n}$ is SPD, $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{p\times m}$ be of full row rank. Let $\alpha > 0$, $\beta > 0$ be the constants, θ be the eigenvalue of $P_{NBT}^{-1}\mathcal{A}$. Then we have*

$$|1 - \theta| < 1.$$

In other words, the eigenvalues of the NBT preconditioned matrix $P_{NBT}^{-1}\mathcal{A}$ are located in the unit circle centered at $(1, 0)$.

P r o o f. By the fact $P_{NBT}^{-1}\mathcal{A} = I - P_{NBT}^{-1}Q_{NBT}$, combined with Theorem 2.1, the conclusion is valid. □

Next, we further analyze the eigenvalue and eigenvector distribution results of the NBT preconditioned matrix.

Theorem 3.1. *Under the conditions in Theorem* 2.1. *Let the NBT preconditioner* P_{NBT} *be defined in* (2.2). *Then the NBT preconditioned matrix* $P_{NBT}^{-1}\mathcal{A}$ *has an eigenvalue* θ. *Further, we have that*

(1) n *eigenvectors of the form* $[u_l^1;0;0]$ $(l=1,2,\ldots,n)$ *that correspond to the eigenvalue* $\theta=1$, *where* u_l^1 $(l=1,2,\ldots,n)$, *are arbitrary linearly independent vectors.*

(2) i $(0\leqslant i\leqslant m+p)$ *eigenvectors of the form* $[u_l^2;v_l^2;w_l^2]$ $(1\leqslant l\leqslant i)$ *that correspond to eigenvalues* $\theta\neq 1$, *where the nonzero vectors* u_l^2 *and* v_l^2 *satisfy the equation* (3.4), *have* $w_l^2=\theta^{-1}(\beta I+\alpha^{-1}CC^\top)^{-1}Cv_l^2$.

P r o o f. Let $[\theta,l]$ be an eigenpair of the preconditioned matrix $P_{NBT}^{-1}\mathcal{A}$, where $l=[u;v;w]$. We consider the eigenvalue problem

$$\begin{pmatrix} A & B^\top & 0 \\ -B & 0 & -C^\top \\ 0 & C & 0 \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix}=\theta\begin{pmatrix} A & B^\top & 0 \\ -B & \alpha I & -C^\top \\ 0 & 0 & \beta I+\dfrac{1}{\alpha}CC^\top \end{pmatrix}\begin{pmatrix} u \\ v \\ w \end{pmatrix}. \tag{3.1}$$

The equation (3.1) can equivalently be reformulated into

$$\begin{cases} (\theta-1)(Au+B^\top v)=0, \\ (1-\theta)(Bu+C^\top w)+\alpha\theta v=0, \\ Cv-\theta\Big(\beta I+\dfrac{1}{\alpha}CC^\top\Big)w=0. \end{cases} \tag{3.2}$$

If $\theta=1$, (3.2) becomes

$$\begin{cases} \alpha\theta v=0, \\ Cv-\Big(\beta I+\dfrac{1}{\alpha}CC^\top\Big)w=0. \end{cases} \tag{3.3}$$

It is obvious that $v=0$ and $w=0$. Therefore, there are n linearly independent eigenvectors $[u_l^1;0;0]$ $(l=1,2,\ldots,n)$ corresponding to the eigenvalue 1, where u_l^1 $(l=1,2,\ldots,n)$ are arbitrary linearly independent vectors.

If $\theta\neq 1$, by the equation (3.2), we have that both u and v are nonzero vectors, otherwise $[u;v;w]=0$, contradiction. The third equation of (3.2) means that $w=\theta^{-1}(\beta I+\alpha^{-1}CC^\top)^{-1}Cv$, inserting it into the second equation of (3.2) gives $(1-\theta)Bu+(\alpha\theta+(1-\theta)\theta^{-1}C^\top(\beta I+\alpha^{-1}CC^\top)^{-1}C)v=0$. Hence, the preconditioned matrix has i $(0\leqslant i\leqslant m+p)$ linearly independent eigenvectors $[u_l^2;v_l^2;w_l^2]$

$(1 \leqslant l \leqslant i)$ that correspond to the eigenvalue $\theta \neq 1$, where u_l^2 and v_l^2 satisfy the equation

$$
(3.4) \qquad \begin{cases} Au_l^2 + B^\top v_l^2 = 0, \\ (1-\theta)Bu_l^2 + \Big(\alpha\theta I + \dfrac{1-\theta}{\theta} C^\top \Big(\beta I + \dfrac{1}{\alpha} CC^\top\Big)^{-1} C\Big) v_l^2 = 0 \end{cases}
$$

and $w_l^2 = \theta^{-1}(\beta I + \alpha^{-1} CC^\top)^{-1} C v_l^2$.

Finally, we show that the $n+i$ eigenvectors are linearly independent. That is to say, it is required to prove that

$$
(3.5) \qquad \begin{pmatrix} u_1^1 \dots u_n^1 \\ 0 \dots 0 \\ 0 \dots 0 \end{pmatrix} \begin{pmatrix} k_1^1 \\ \vdots \\ k_n^1 \end{pmatrix} + \begin{pmatrix} u_1^2 \dots u_i^2 \\ v_1^2 \dots v_i^2 \\ w_1^2 \dots w_i^2 \end{pmatrix} \begin{pmatrix} k_1^2 \\ \vdots \\ k_i^2 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}
$$

holds only when the vectors k^1 and k^2 are zero. We know that the first matrix in the equation (3.5) consists of eigenvectors corresponding to the eigenvalue 1, the last matrix consists of those corresponding to the eigenvalue $\lambda_l \neq 1$ $(l = 1, 2, \dots, i)$. By multiplying both sides of (3.5) from left with $P_{NBT}^{-1}\mathcal{A}$, we obtain

$$
(3.6) \qquad \begin{pmatrix} u_1^1 \dots u_n^1 \\ 0 \dots 0 \\ 0 \dots 0 \end{pmatrix} \begin{pmatrix} k_1^1 \\ \vdots \\ k_n^1 \end{pmatrix} + \begin{pmatrix} u_1^2 \dots u_i^2 \\ v_1^2 \dots v_i^2 \\ w_1^2 \dots w_i^2 \end{pmatrix} \begin{pmatrix} \lambda_1 k_1^2 \\ \vdots \\ \lambda_i k_i^2 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.
$$

By subtracting (3.5) from (3.6), it holds that

$$
(3.7) \qquad \begin{pmatrix} u_1^2 \dots u_i^2 \\ v_1^2 \dots v_i^2 \\ w_1^2 \dots w_i^2 \end{pmatrix} \begin{pmatrix} (\lambda_1 - 1) k_1^2 \\ \vdots \\ (\lambda_i - 1) k_i^2 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.
$$

Since $[u_l^2; v_l^2; w_l^2]$ are i linearly independent vectors and the eigenvalues $\lambda_l \neq 1$ $(l = 1, 2, \dots, i)$, we have $k_l^2 = 0$ $(l = 1, 2, \dots, i)$. Substituting $k_l^2 = 0$ $(l = 1, 2, \dots, i)$ into (3.6), we see that both k^1 and k^2 are equal to zero. $\square$

As is known to all, Krylov subspace theory states that the iteration with any method of optimality property (such as GMRES method) in exact arithmetic will terminate as soon as the degree of the minimal polynomial is attained [17]. In the following, we study an upper bound of the degree of the minimal polynomial of the preconditioned matrix.

Theorem 3.2. *Assume that the conditions of Theorem* 2.1 *hold. Then the dimension of the Krylov subspace* $\mathcal{K}(P_{NBT}^{-1}\mathcal{A}, b)$ *is at most* $m + p + 1$.

P r o o f. The proof process is similar to [15], Theorem 3. Due to the expression (2.2), we have

$$P_{NBT}^{-1} = \begin{pmatrix} A^{-1} - A^{-1}B^\top U^{-1}BA^{-1} & -A^{-1}B^\top U^{-1} & -A^{-1}B^\top U^{-1}C^\top R^{-1} \\ U^{-1}BA^{-1} & U^{-1} & U^{-1}C^\top R^{-1} \\ 0 & 0 & R^{-1} \end{pmatrix},$$

where $U = \alpha I + BA^{-1}B^\top$ and $R = \beta I + \alpha^{-1}CC^\top$. Furthermore, it follows that

$$(3.8)\quad P_{NBT}^{-1}\mathcal{A} = \begin{pmatrix} I & A^{-1}B^\top - A^{-1}B^\top U^{-1}BA^{-1}B^\top - A^{-1}B^\top U^{-1}C^\top R^{-1}C & A^{-1}B^\top U^{-1}C^\top \\ 0 & U^{-1}BA^{-1}B^\top + U^{-1}C^\top R^{-1}C & -U^{-1}C^\top \\ 0 & R^{-1}C & 0 \end{pmatrix}.$$

It is easy to observe that the preconditioned matrix (3.8) can be equivalently written as

$$P_{NBT}^{-1}\mathcal{A} = \begin{pmatrix} I & \Pi \\ 0 & \Theta \end{pmatrix},$$

where $\Pi \in \mathbb{R}^{n\times(m+p)}$ and $\Theta \in \mathbb{R}^{(m+p)\times(m+p)}$.

Let λ_l $(l = 1, 2, \ldots, m+p)$ be the eigenvalues of the matrix Θ. Then the characteristic polynomial of the matrix $P_{NBT}^{-1}\mathcal{A}$ is

$$\Phi_{P_{NBT}^{-1}\mathcal{A}}(\lambda) = \det(P_{NBT}^{-1}\mathcal{A} - \lambda I) = (-1)^{n+m+p}(\lambda-1)^n \prod_{l=1}^{m+p}(\lambda - \lambda_l).$$

Let

$$\Psi(\lambda) = (\lambda - 1)\prod_{l=1}^{m+p}(\lambda - \lambda_l).$$

Then

$$\Psi(P_{NBT}^{-1}\mathcal{A}) = (P_{NBT}^{-1}\mathcal{A} - I)\prod_{l=1}^{m+p}(P_{NBT}^{-1}\mathcal{A} - \lambda_l I) = \begin{pmatrix} 0 & \Pi \prod_{l=1}^{m+p}(\Theta - \lambda_l I) \\ 0 & (\Theta - I)\prod_{l=1}^{m+p}(\Theta - \lambda_l I) \end{pmatrix}.$$

Because λ_l $(l = 1, 2, \ldots, m+p)$ are the eigenvalues of Θ, we have $\prod_{l=1}^{m+p}(\Theta - \lambda_l I) = 0$ by the Hamilton-Cayley theorem. Therefore, the degree of the minimal polynomial of the preconditioned matrix $P_{NBT}^{-1}\mathcal{A}$ is at most $m+p+1$. Consequently, the dimension of the corresponding Krylov subspace $\mathcal{K}(P_{NBT}^{-1}\mathcal{A}, b)$ is at most $m + p + 1$. $\square$

4. Numerical experiments

In this section, we present two numerical examples, which show the performance and robustness of the proposed preconditioner over existing preconditioners, i.e., the BD preconditioner, block preconditioners P_3 (authors in [21] showed that the P_1, P_2 and P_3 preconditioners in (1.5) can significantly improve the convergence speed of the GMRES method, from numerical results it can be observed that the preconditioner P_3 outperforms other two preconditioners, thus we test only the P_3 preconditioner), P_4 and APSS preconditioner. The above preconditioners are listed in Section 1. The numerical behavior of these preconditioned GMRES iteration methods is tested and evaluated in terms of the number of iteration steps (denoted 'IT') and the computing time (denoted 'CPU'). All of them are used as the left preconditioners of the GMRES iteration method to solve the three-by-three block saddle-point problem (1.2).

In all the tests, the initial vector is the zero vector and the right-hand side vector b is chosen such that the exact solution of (1.2) is the vector that has all its components equal to one. The iterations stop once the relative residual satisfies

$$\mathrm{RES} := \frac{\sqrt{\|f - Ax^{(k)} - B^\top y^{(k)}\|_2^2 + \|g - Bx^{(k)} - C^\top z^{(k)}\|_2^2 + \|h - Cy^{(k)}\|_2^2}}{\sqrt{\|f\|_2^2 + \|g\|_2^2 + \|h\|_2^2}} < 10^{-6}$$

with $((x^{(k)})^\top, (y^{(k)})^\top, (z^{(k)})^\top)^\top$ being the current approximate solution or if the prescribed maximum iteration count $k_{\max} = 1500$ is exceeded. The numerical tables indicate '-' if iteration steps exceed 1500. All experiments are performed in MATLAB 2017(a) on an Intel Core(4G RAM) Windows 10 system.

Example 4.1. Consider the three-by-three block saddle-point problem (1.2), in which

$$\begin{aligned} A &= \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times 2p^2}, \\ B &= (\, I \otimes F \quad F \otimes I \,) \in \mathbb{R}^{p^2 \times 2p^2} \quad \text{and} \quad C = E \otimes F \in \mathbb{R}^{p^2 \times p^2}, \end{aligned}$$

where

$$T = \frac{\nu}{h^2}\mathrm{tridiag}(-1, 2, -1) \in \mathbb{R}^{p \times p}, \quad F = \frac{1}{h}\mathrm{tridiag}(0, 1, -1) \in \mathbb{R}^{p \times p},$$

and $E = \mathrm{diag}(1, p+1, \ldots, p^2 - p + 1)$, $\otimes$ means the Kronecker product symbol and $h = 1/(p+1)$.

Example 4.2 ([11], [20]). We consider the three-by-three saddle-point problem (1.2), where the block matrices A and B arise from the two dimensional "leak" lid-driven cavity problem in a square domain $\Omega = (-1 \leqslant x \leqslant 1, -1 \leqslant y \leqslant 1)$, i.e., the Stokes equations

$$(4.1) \qquad \begin{cases} -\Delta \mathbf{u} + \nabla p = 0 & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega. \end{cases}$$

A Dirichlet no-flow condition is applied on the side and bottom boundaries, and the nonzero horizontal velocity on the lid is $\{y = 1; -1 \leqslant x \leqslant 1 \mid \mathbf{u}_x = 1\}$. Here, $\mathbf{u}$ and p represent the velocity vector field and the pressure scalar field, respectively, Δ is the vector Laplacian in $\mathbb{R}^2$, ∇ denotes the gradient, and $\nabla\cdot$ is the divergence.

To obtain the block matrices A and B, we adopted the IFISS software developed by Elman et al. [7] to discretize the Stokes equation (4.1). Here, the $Q_1 - P_0$ finite element method (FEM) on the uniform and stretched grids are taken. Note that the block B generated by the IFISS package is not of full row rank, so the first two rows of B are dropped to get a full row rank matrix. Besides, to make the linear system (1.2) ill-conditioned and not too sparse, we consider the matrix C in the form

$$C = [\operatorname{diag}(1, 3, 5, \ldots, 2l - 1), \operatorname{randn}(l, 2)],$$

where $l = m - 2$, $\operatorname{randn}(l, 2)$ denotes an l-by-2 matrix of normally distributed random numbers.

Experimental explanations—Example 4.1. The experiment problem is formed by setting different dimension sizes in Example 4.1. In existing preconditioners, to make the preconditioners have a better preconditioning effect, we choose their respective optimal parameter or matrix, i.e., $D = I$ for P_4 [3], $\alpha = 0.01$ for SS preconditioner [4], $\alpha = 1.5$ for APSS preconditioner (the APSS preconditioned GMRES method has better convergence under this value of parameter). Through Remark 2.1 in Section 2, we can choose a smaller $\beta = 10^{-5}$ and the parameter α could be calculated by the quarto equation. The values of the parameter α for the NBT preconditioner are listed in Table 1 for different dimensions. Since the value of parameter α is only related to β and the matrix C, the choices of parameter are still applicable under different values of ν. In Tables 2, 3 and 4, we list the number of iteration steps, the relative residuals and the corresponding CPU times of the preconditioned GMRES iteration methods with I (i.e., no preconditioning), P_{BD}, P_3, P_4, P_{SS}, P_{APSS} and P_{NBT} for Example 4.1 with different p. We choose $\nu = 1$, 0.1, 0.01 in Tables 2, 3 and 4, respectively. In addition, the eigenvalue distributions of the different preconditioned matrices with $p = 16$ ($\nu = 1$) are shown in Figure 1. In

Figure 4, iteration steps of the NBT preconditioned GMRES method with varying α ($p = 32$ and $\nu = 1$) are given to show the effectiveness of the calculated parameter α.

p	16	32	48	56	64	80
α	4338	$3.45e+4$	$1.16e+5$	$1.84e+5$	$2.75e+5$	$5.37e+5$

Table 1. The value of parameter α for the NBT preconditioner with $\beta = 10^{-5}$ in different dimensions for Example 4.1.

Pre.	p	16	32	48	56	64	80
I	IT	865	-	-	-	-	-
	RES	$8.3e-07$	-	-	-	-	-
	CPU	2.5452	-	-	-	-	-
P_{BD}	IT	4	4	4	4	5	6
	RES	$1.5e-10$	$1.2e-08$	$7.7e-07$	$9.9e-08$	$4.9e-07$	$2.5e-07$
	CPU	0.0567	0.4774	4.0985	12.5711	34.9930	96.0728
P_3	IT	3	3	3	3	3	5
	RES	$9.1e-11$	$2.3e-08$	$1.0e-07$	$1.4e-07$	$6.0e-07$	$2.2e-09$
	CPU	0.0638	0.4682	4.6270	13.0306	35.1039	96.6838
P_4	IT	2	2	2	2	2	2
	RES	$1.7e-12$	$2.1e-11$	$9.6e-11$	$1.7e-10$	$2.3e-10$	$5.2e-10$
	CPU	0.0441	0.1760	1.1059	2.3689	6.4137	19.2243
P_{SS}	IT	2	2	2	2	2	2
	RES	$7.6e-07$	$3.4e-07$	$2.1e-07$	$1.7e-07$	$1.5e-07$	$1.3e-07$
	CPU	0.0450	0.2301	1.1673	1.9706	3.6363	12.5058
P_{APSS}	IT	11	11	12	12	11	13
	RES	$2.2e-07$	$5.4e-07$	$4.8e-07$	$8.1e-07$	$4.4e-07$	$2.9e-07$
	CPU	0.0793	0.1976	0.8835	1.7202	3.2089	12.1361
P_{NBT}	IT	7	9	11	11	13	15
	RES	$1.6e-08$	$9.5e-07$	$1.5e-07$	$9.2e-07$	$1.9e-08$	$8.9e-09$
	CPU	0.0563	0.1777	0.8429	1.7192	3.1772	10.8361

Table 2. The numerical results of the preconditioned GMRES method with $\nu = 1$ for Example 4.1.

Experimental explanations—Example 4.2. In Example 4.2, we choose various gradient parameters h to obtain test problems of different dimensions. Especially, the parameters of the APSS preconditioner are selected as the testing optimal parameters. For other existing preconditioners, the parameters or parameter matrix of the preconditioners are similar to Example 4.1. Besides, we can select $\beta = 10^{-6}$, it is easy to find from Remark 2.1 that the parameter α of the NBT preconditioner is only related to the matrix C under uniform and stretched grids, therefore the values of α under the two kinds of different grids are the same. The numerical results of the different preconditioned GMRES methods are listed in Tables 5 and 6. In Figures 2 and 3, eigenvalue distributions of the preconditioned matrices under compared preconditioners are plotted on uniform and stretched 16×16 grids. We also draw the relationships between the parameter α and iteration steps of the NBT preconditioned GMRES method on uniform 32×32 grids.

Pre.	p	16	32	48	56	64	80
I	IT	618	-	-	-	-	-
	RES	9.8e-07	-	-	-	-	-
	CPU	1.7501	-	-	-	-	-
P_{BD}	IT	4	4	4	4	4	4
	RES	$1.9\mathrm{e}-11$	$1.2\mathrm{e}-09$	$7.3\mathrm{e}-09$	$5.9\mathrm{e}-08$	$5.9\mathrm{e}-08$	$2.7\mathrm{e}-07$
	CPU	0.0610	0.5810	5.1490	14.8590	35.1190	108.4560
P_3	IT	3	3	3	3	3	3
	RES	$4.5\mathrm{e}-12$	$5.1\mathrm{e}-10$	$9.8\mathrm{e}-09$	$2.9\mathrm{e}-08$	$2.2\mathrm{e}-08$	$1.4\mathrm{e}-07$
	CPU	0.0680	0.5790	5.0610	14.3270	35.5440	110.5970
P_4	IT	2	2	2	2	2	2
	RES	$1.8\mathrm{e}-12$	$2.2\mathrm{e}-11$	$1.0\mathrm{e}-10$	$1.7\mathrm{e}-10$	$2.3\mathrm{e}-10$	$5.2\mathrm{e}-10$
	CPU	0.0630	0.1790	0.9430	1.9570	4.7710	14.8130
P_{SS}	IT	3	3	3	3	3	3
	RES	$2.2\mathrm{e}-08$	$2.1\mathrm{e}-08$	$2.3\mathrm{e}-08$	$2.7\mathrm{e}-08$	$3.9\mathrm{e}-08$	$6.3\mathrm{e}-08$
	CPU	0.0474	0.2178	0.9138	1.8811	3.6129	10.6621
P_{APSS}	IT	19	22	23	23	25	25
	RES	$6.1\mathrm{e}-07$	$3.1\mathrm{e}-07$	$6.4\mathrm{e}-07$	$5.5\mathrm{e}-07$	$1.3\mathrm{e}-07$	$1.7\mathrm{e}-07$
	CPU	0.0738	0.2501	1.0686	2.0063	3.8605	10.6983
P_{NBT}	IT	7	9	11	12	13	15
	RES	$1.9\mathrm{e}-07$	$7.8\mathrm{e}-07$	$2.2\mathrm{e}-07$	$4.6\mathrm{e}-07$	$1.7\mathrm{e}-08$	$2.6\mathrm{e}-08$
	CPU	0.0594	0.0899	0.2928	0.5382	0.8535	2.1596

Table 3. The numerical results of the preconditioned GMRES method with $\nu = 0.1$ for Example 4.1.

Analysis of numerical results—Example 4.1.

▷ No preconditioned GMRES method is not convergent when $p > 16$. On the contrary, all preconditioned GMRES methods are compelling.

▷ Comparison of iteration steps of different preconditioned GMRES methods:

$$P_{APSS} > P_{NBT} > P_{BD} > P_3 > P_4 = P_{SS} \quad \text{(Table 2)},$$
$$P_{APSS} > P_{NBT} > P_{BD} > P_3 = P_{SS} > P_4 \quad \text{(Table 3)},$$
$$P_{APSS} > P_{NBT} > P_{BD} = P_{SS} > P_3 > P_4 \quad \text{(Table 4)}.$$

▷ Comparison of CPU times of different preconditioned GMRES methods:

$$P_3 > P_{BD} > P_4 > P_{SS} > P_{APSS} > P_{NBT} \quad \text{(Table 2)},$$
$$P_3 > P_{BD} > P_4 > P_{APSS} > P_{SS} > P_{NBT} \quad \text{(Table 3)},$$
$$P_{BD} > P_3 > P_{APSS} > P_4 > P_{SS} > P_{NBT} \quad \text{(Table 4)}.$$

▷ Through the comparison of data in different tables, it is easy to find that the CPU times of the NBT preconditioned GMRES method are smaller if ν is smaller.

Pre.	p	16	32	48	56	64	80
I	IT	561	-	-	-	-	-
	RES	9.2e − 07	-	-	-	-	-
	CPU	1.0762	-	-	-	-	-
P_{BD}	IT	4	4	4	4	5	6
	RES	1.6e − 12	9.0e − 11	8.2e − 10	1.1e − 09	5.8e − 10	2.1e − 08
	CPU	0.0570	0.5960	5.2080	14.1490	36.2650	110.5390
P_3	IT	3	3	3	3	3	3
	RES	5.2e − 13	3.9e − 11	5.1e − 10	4.9e − 10	2.2e − 09	4.6e − 08
	CPU	0.0590	0.5930	5.0890	13.6860	33.5370	109.7490
P_4	IT	2	2	2	2	2	2
	RES	1.6e − 12	2.2e − 11	1.0e − 10	1.7e − 10	2.6e − 10	5.2e − 10
	CPU	0.0620	0.1730	0.9390	1.9870	4.7820	14.6080
P_{SS}	IT	4	4	4	4	4	4
	RES	8.5e − 08	8.2e − 08	7.6e − 08	7.5e − 08	7.5e − 08	8.7e − 08
	CPU	0.0786	0.1993	1.0078	1.8599	3.6743	10.5774
P_{APSS}	IT	42	49	45	43	50	50
	RES	6.0e − 07	4.8e − 08	7.2e − 07	8.1e − 07	7.3e − 07	5.9e − 07
	CPU	0.1275	0.4266	1.6629	2.8772	5.4542	14.2465
P_{NBT}	IT	9	9	11	12	13	14
	RES	1.6e − 08	1.9e − 07	3.9e − 07	3.8e − 08	2.4e − 08	7.6e − 07
	CPU	0.0606	0.0893	0.3151	0.5158	0.8533	2.0978

Table 4. The numerical results of the preconditioned GMRES method with $\nu = 0.01$ for Example 4.1.

▷ It is found by the eigenvalue distributions of the preconditioned matrix that the eigenvalue distributions of the coefficient matrix $\mathcal{A}$ are scattered, but the eigenvalue distributions of all preconditioned matrices are clustered.

▷ The iteration steps of the NBT preconditioned GMRES method increase with the increase of α. Besides, it shows that the NBT preconditioner has the testing optimal parameters, but the calculated value of α is also effective.

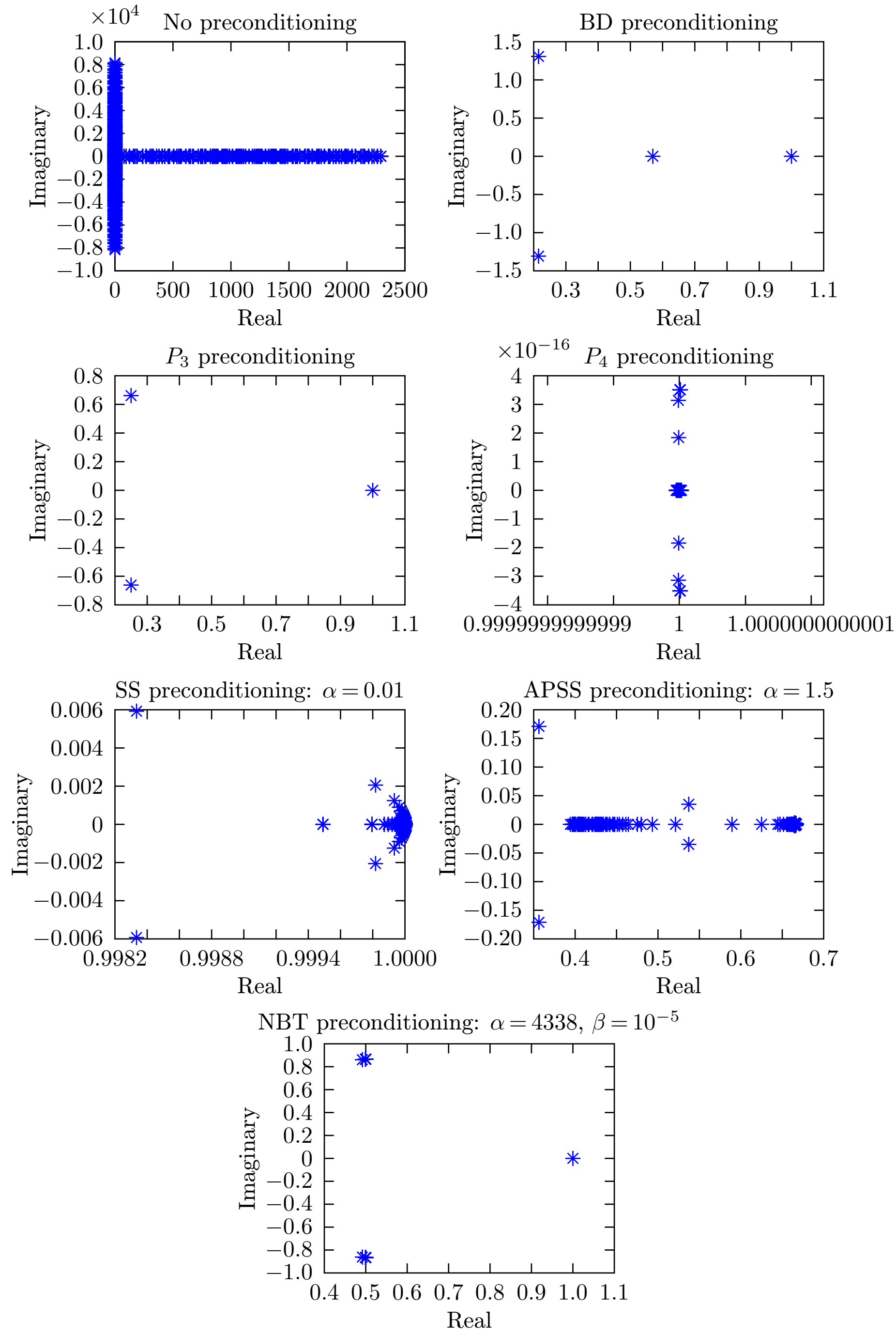

Figure 1. The eigenvalue distribution of the preconditioned matrix with $p = 16$ ($\nu = 1$) for Example 4.1.

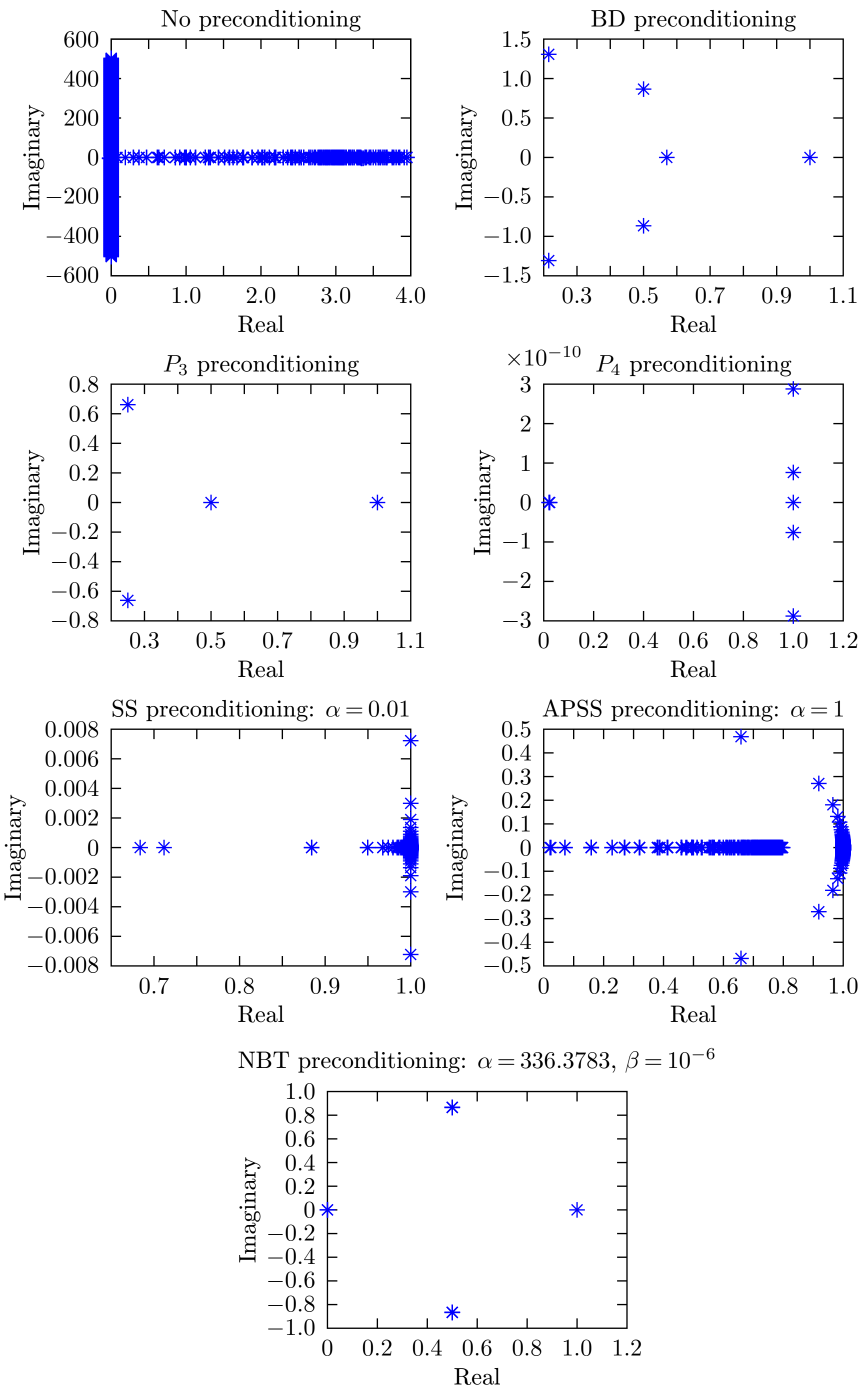

Figure 2. The eigenvalue distribution of the preconditioned matrix with the uniform 16×16 grid for Example 4.2.

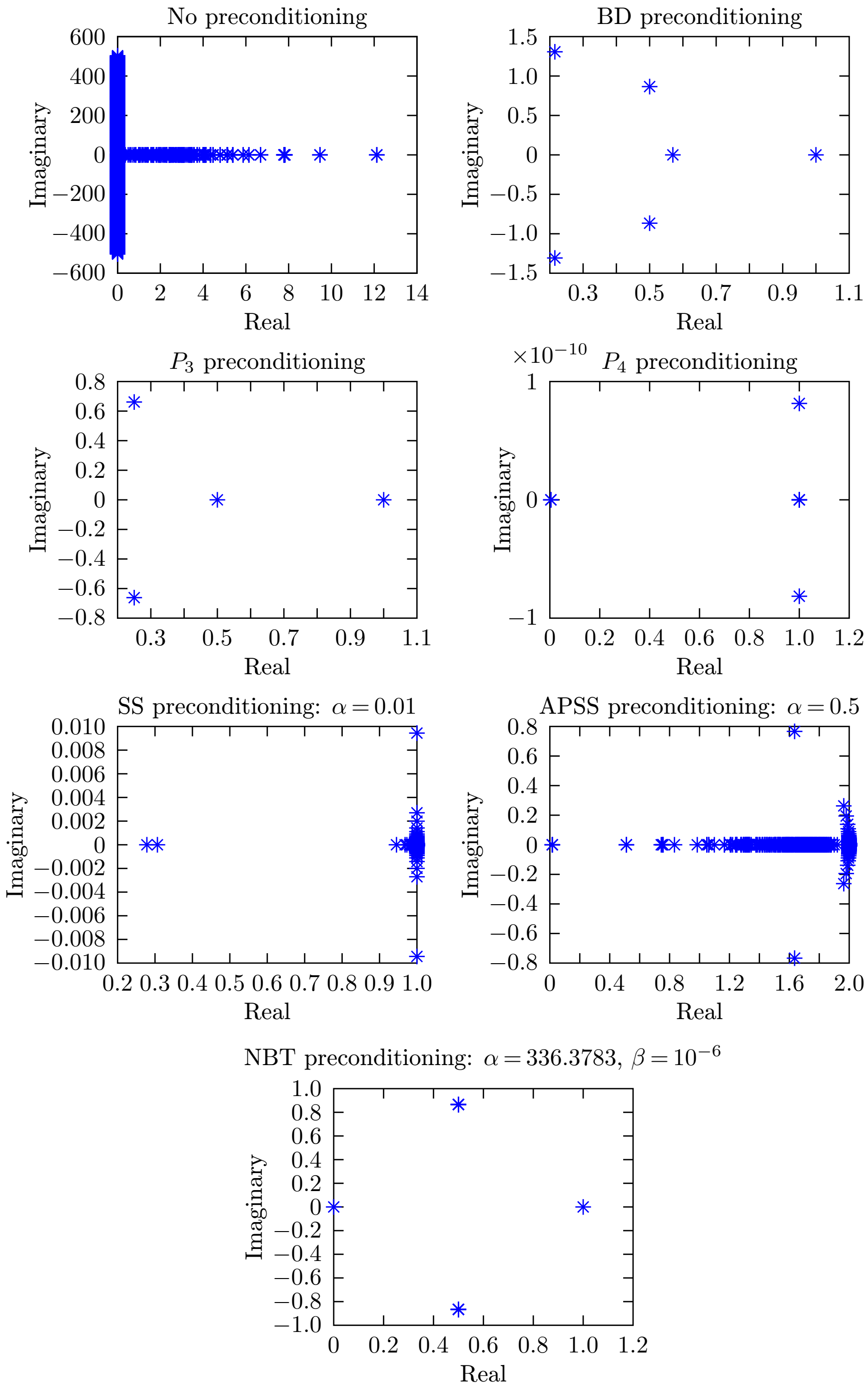

Figure 3. The eigenvalue distribution of the preconditioned matrix with the stretched 16×16 grid for Example 4.2.

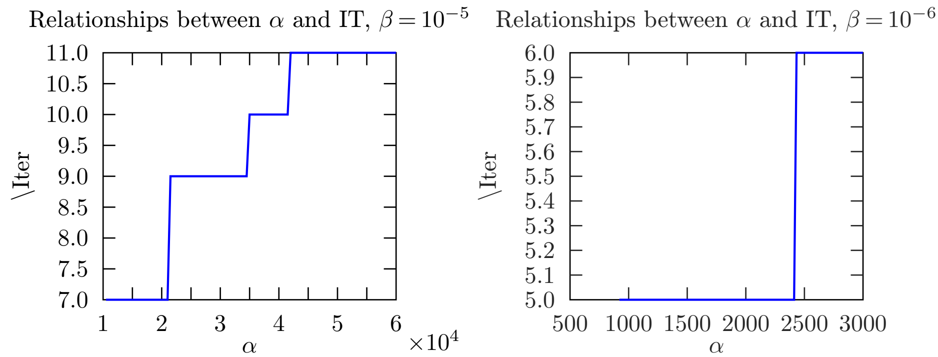

Figure 4. Iteration steps of the NBT preconditioned GMRES method with varying α for Example 4.1 (left: $p = 32$ and $\nu = 1$) and Example 4.2 (right: uniform 32×32 grids).

Analysis of numerical results—Example 4.2.

▷ Without preconditioning, the GMRES method converges slowly, possibly not at all. However, all the preconditioned GMRES methods have good convergence effect.

▷ When the gradient is small, the convergence effects of different preconditioned GMRES methods are almost the same. Furthermore, with the increase of gradient, the following sequences hold:

$$
\begin{aligned}
\text{IT: } & P_{APSS} > P_{BD} \geqslant P_{SS} > P_3 > P_{NBT} > P_4 && \text{(Table 5);}\\
& P_{APSS} > P_{BD} > P_3 \geqslant P_{SS} \geqslant P_{NBT} > P_4 && \text{(Table 6);}\\
\text{CPU: } & P_3 > P_{BD} > P_{SS} > P_{APSS} > P_4 > P_{NBT} && \text{(Table 5);}\\
& P_3 > P_{BD} > P_{SS} > P_{APSS} > P_4 > P_{NBT} && \text{(Table 6).}
\end{aligned}
$$

It is easy to find that the number of iteration steps of the GMRES method with P_4 preconditioner is the least, but its CPU time is larger than that of the NBT preconditioner.

▷ The NBT preconditioner has a stable numerical effect under given testing parameters, which means that the formula for calculating the parameters given in Remark 2.1 is valid.

▷ Figures 2 and 3 show that the spectral distribution of the saddle-point coefficient matrix is relatively scattered. On the contrary, all preconditioners improve the spectral distribution of the saddle-point coefficient matrix $\mathcal{A}$. It is worth mentioning that the eigenvalues of the preconditioned matrix $P_{NBT}^{-1}\mathcal{A}$ are located in the circle with $(\frac{1}{2}, 0)$ as the center and $\frac{1}{2}$ as the radius. This intuitively yields a much more accurate spectral distribution result than Lemma 3.1.

Pre.		I	P_{BD}	P_3	P_4	P_{SS}	P_{APSS}	P_{NBT}
8×8	(α, β)	-	-	-	-	(0.01,-)	(0.5,-)	$(79.5990, 10^{-6})$
	IT	140	6	4	4	4	14	11
	RES	$2.1\mathrm{e}-07$	$3.7\mathrm{e}-10$	$6.1\mathrm{e}-11$	$4.0\mathrm{e}-14$	$5.6\mathrm{e}-08$	$2.0\mathrm{e}-07$	$5.7\mathrm{e}-09$
	CPU	0.1382	0.0470	0.0440	0.0590	0.0511	0.0569	0.0554
16×16	(α, β)	-	-	-	-	(0.01,-)	(1,-)	$(336.3783, 10^{-6})$
	IT	525	6	4	3	4	21	8
	RES	$5.1\mathrm{e}-07$	$4.5\mathrm{e}-07$	$9.8\mathrm{e}-08$	$2.4\mathrm{e}-09$	$9.3\mathrm{e}-07$	$8.9\mathrm{e}-07$	$6.3\mathrm{e}-09$
	CPU	1.0823	0.0850	0.0950	0.0460	0.0694	0.0922	0.0716
32×32	(α, β)	-	-	-	-	(0.01,-)	(1,-)	$(1.4\mathrm{e}+3, 10^{-6})$
	IT	-	8	7	2	7	27	5
	RES	-	$7.3\mathrm{e}-07$	$8.6\mathrm{e}-10$	$1.8\mathrm{e}-07$	$1.1\mathrm{e}-08$	$9.9\mathrm{e}-07$	$1.8\mathrm{e}-07$
	CPU	-	1.3230	1.3820	0.2850	0.6465	0.6436	0.2378
64×64	(α, β)	-	-	-	-	(0.01,-)	(0.1,-)	$(5.5\mathrm{e}+3, 10^{-6})$
	IT	-	9	7	2	9	28	5
	RES	-	$1.3\mathrm{e}-07$	$1.8\mathrm{e}-08$	$7.8\mathrm{e}-09$	$2.3\mathrm{e}-07$	$4.8\mathrm{e}-07$	$9.8\mathrm{e}-07$
	CPU	-	81.3330	82.9800	10.7020	20.9719	10.8527	2.8110

Table 5. The numerical results of the different preconditioned GMRES methods with uniform grids for Example 4.2.

▷ The iteration steps of the NBT preconditioned GMRES method are unchangeable when $\alpha < 2400$, iteration steps increase to 6 when α is greater than 2400, which shows that the calculated parameter α of the NBT preconditioner is valid.

It is common knowledge that the advantage of the preconditioned GMRES iteration method depends not only on the less number of iteration steps, but also on the fewer CPU time. Combined with the analysis of numerical results, it shows that the new block triangular preconditioner is superior to the compared preconditioner for solving three-by-three block saddle-point problem (1.2).

Pre.		I	P_{BD}	P_3	P_4	P_{SS}	P_{APSS}	P_{NBT}
8×8	(α,β)	-	-	-	-	(0.01,-)	(0.25,-)	$(79.5990,10^{-6})$
	IT	142	6	4	3	4	12	8
	RES	$7.6\mathrm{e}-07$	$5.1\mathrm{e}-10$	$1.1\mathrm{e}-10$	$4.1\mathrm{e}-07$	$1.8\mathrm{e}-08$	$6.5\mathrm{e}-07$	$4.4\mathrm{e}-07$
	CPU	0.1703	0.0500	0.0440	0.0450	0.0532	0.0521	0.0450
16×16	(α,β)	-	-	-	-	(0.01,-)	(0.5,-)	$(336.3783,10^{-6})$
	IT	529	6	4	2	6	15	5
	RES	$9.4\mathrm{e}-07$	$2.2\mathrm{e}-07$	$2.4\mathrm{e}-07$	$2.8\mathrm{e}-07$	$4.2\mathrm{e}-09$	$4.7\mathrm{e}-07$	$2.8\mathrm{e}-07$
	CPU	1.1028	0.0790	0.0890	0.0500	0.0783	0.0865	0.0607
32×32	(α,β)	-	-	-	-	(0.01,-)	(0.35,-)	$(1.4\mathrm{e}+3,10^{-6})$
	IT	-	9	7	2	7	12	5
	RES	-	$5.0\mathrm{e}-08$	$1.5\mathrm{e}-08$	$1.3\mathrm{e}-08$	$2.9\mathrm{e}-07$	$7.4\mathrm{e}-07$	$4.4\mathrm{e}-08$
	CPU	-	1.1760	1.1950	0.2350	0.3971	0.5821	0.2224
64×64	(α,β)	-	-	-	-	(0.01,-)	(0.15,-)	$(5.5\mathrm{e}+3,10^{-6})$
	IT	-	13	8	2	6	15	6
	RES	-	$9.7\mathrm{e}-07$	$8.2\mathrm{e}-07$	$1.7\mathrm{e}-10$	$4.7\mathrm{e}-07$	$7.6\mathrm{e}-07$	$4.4\mathrm{e}-09$
	CPU	-	76.1870	87.4730	8.1427	16.9526	8.5019	3.0601

Table 6. The numerical results of the different preconditioned GMRES methods with stretched grids for Example 4.2.

5. Conclusions

In this paper, we established a new block triangular (NBT) preconditioner for solving the three-by-three block saddle-point problem (1.2). Theoretical analysis proved the unconditional convergence of the iteration method produced by the NBT preconditioner. The spectral properties of the NBT preconditioned matrix were also considered. Numerical experiments revealed that the proposed NBT preconditioner has excellent superiority in calculated parameters compared with other testing preconditioners in exact algorithms.

A c k n o w l e d g e m e n t s. The authors are very thankful to the referee for his/her constructive comments and valuable suggestions, which greatly improved the original manuscript of this paper.

References

[1] *M. Abdolmaleki, S. Karimi, D. K. Salkuyeh*: A new block-diagonal preconditioner for a class of 3×3 block saddle point problems. Mediterr. J. Math. *19* (2022), Article ID 43, 15 pages. zbl MR doi

[2] *H. Aslani, D. K. Salkuyeh*: Semi-convergence of the APSS method for a class of nonsymmetric three-by-three singular saddle point problems. Available at `https://arxiv.org/abs/2208.00814` (2022), 17 pages.

[3] *H. Aslani, D. K. Salkuyeh, F. P. A. Beik*: On the preconditioning of three-by-three block saddle point problems. Filomat *15* (2021), 5181–5194. MR doi

[4] *Y. Cao*: Shift-splitting preconditioners for a class of block three-by-three saddle point problems. Appl. Math. Lett. *96* (2019), 40–46. zbl MR doi

[5] *Y. Cao*: A block positive-semidefinite splitting preconditioner for generalized saddle point linear systems. J. Comput. Appl. Math. *374* (2020), Article ID 112787, 15 pages. zbl MR doi

[6] *P. Degond, P.-A. Raviart*: An analysis of the Darwin model of approximation to Maxwell's equations. Forum Math. *4* (1992), 13–44. zbl MR doi

[7] *H. C. Elman, A. Ramage, D. J. Silvester*: Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow. ACM Trans. Math. Softw. *33* (2007), Article ID 14, 18 pages. zbl MR doi

[8] *D. Han, X. Yuan*: Local linear convergence of the alternating direction method of multipliers for quadratic programs. SIAM J. Numer. Anal. *51* (2013), 3446–3457. zbl MR doi

[9] *K. Hu, J. Xu*: Structure-preserving finite element methods for stationary MHD models. Math. Comput. *88* (2019), 553–581. zbl MR doi

[10] *N. Huang*: Variable parameter Uzawa method for solving a class of block three-by-three saddle point problems. Numer. Algorithms *85* (2020), 1233–1254. zbl MR doi

[11] *N. Huang, Y.-H. Dai, Q. Hu*: Uzawa methods for a class of block three-by-three saddle-point problems. Numer. Linear Algebra Appl. *26* (2019), Article ID e2265, 26 pages. zbl MR doi

[12] *N. Huang, C.-F. Ma*: Spectral analysis of the preconditioned system for the 3×3 block saddle point problem. Numer. Algorithms *81* (2019), 421–444. zbl MR doi

[13] *Y.-M. Huang*: A practical formula for computing optimal parameters in the HSS iteration methods. J. Comput. Appl. Math. *255* (2014), 142–149. zbl MR doi

[14] *Z.-G. Huang, L.-G. Wang, Z. Xu, J.-J. Cui*: An efficient preconditioned variant of the PSS preconditioner for generalized saddle point problems. Appl. Math. Comput. *376* (2020), Article ID 125110, 26 pages. zbl MR doi

[15] *L. Meng, J. Li, S.-X. Miao*: A variant of relaxed alternating positive semi-definite splitting preconditioner for double saddle point problems. Japan J. Ind. Appl. Math. *38* (2021), 979–998. zbl MR doi

[16] *P. Monk*: Analysis of a finite element method for Maxwell's equations. SIAM J. Numer. Anal. *29* (1992), 714–729. zbl MR doi

[17] *Y. Saad*: Iterative Methods for Sparse Linear Systems. SIAM, Philadephia, 2003. zbl MR doi

[18] *D. K. Salkuyeh, H. Aslani, Z.-Z. Liang*: An alternating positive semidefinite splitting preconditioner for the three-by-three block saddle point problems. Math. Commun. *26* (2021), 177–195. zbl MR

[19] *L. Wang, K. Zhang*: Generalized shift-splitting preconditioner for saddle point problems with block three-by-three structure. Open Access Library J. *6* (2019), Article ID e5968, 13 pages. doi

[20] *N.-N. Wang, J.-C. Li*: On parameterized block symmetric positive definite preconditioners for a class of block three-by-three saddle point problems. J. Comput. Appl. Math. *405* (2022), Article ID 113959, 15 pages. zbl MR doi

[21] *X. Xie, H.-B. Li*: A note on preconditioning for the 3×3 block saddle point problem. Comput. Math. Appl. *79* (2020), 3289–3296. zbl MR doi

[22] *D. M. Young*: Iterative Solution of Large Linear Systems. Computer Science and Applied Mathematics. Academic Press, New York, 1971. zbl MR doi

[23] *N. Zhang, R.-X. Li, J. Li*: Lopsided shift-splitting preconditioner for saddle point problems with three-by-three structure. Comput. Appl. Math. *41* (2022), Articles ID 261, 16 pages. zbl MR doi

[24] *J.-L. Zhu, Y.-J. Wu, A.-L. Yang*: A two-parameter block triangular preconditioner for double saddle point problem arising from liquid crystal directors modeling. Numer. Algorithms *89* (2022), 987–1006. zbl MR doi

Authors' address: *Jun Li* (corresponding author), *Xiangtuan Xiong*, College of Mathematics and Statistics, Northwest Normal University, 967 Anning E Rd, Anning District, Lanzhou, 730070, P. R. China, e-mail: `junli026430@163.com`, `xiongxt@gmail.com`.