Indian J Pure Appl Math (2026) 57:1144–1159
https://doi.org/10.1007/s13226-024-00615-2

ORIGINAL RESEARCH

A generalized relaxed block positive-semidefinite splitting preconditioner for generalized saddle point linear system

Jun Li · Lingsheng Meng · Shu-Xin Miao

Received: 2 July 2023 / Accepted: 27 May 2024 / Published online: 20 June 2024

Abstract In this paper, based on the block positive-semidefinite splitting (BPS) preconditioner studied recently and the relaxation technique, a generalized relaxed BPS (GRBPS) preconditioner is proposed for generalized saddle point linear system. Spectral properties of the GRBPS preconditioned matrix are analyzed in details. Theoretical results show that the eigenvalues of the preconditioned matrix are clustered at only two points when the iteration parameters are close to zero. Finally, a numerical example is provided to verify the efficiency of the GRBPS preconditioner.

Keywords Generalized saddle point linear system · Preconditioning · Matrix splitting · Spectral properties · Krylov subspace method

Mathematics Subject Classification 65F08 · 65F10 · 65F50

1 Introduction

We consider the following large sparse generalized saddle point linear system:

$$\mathscr{A}\mathrm{u} = \begin{pmatrix} A & B^* \\ -B & C \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \equiv b, \tag{1.1}$$

where $A \in \mathbb{C}^{n\times n}$ is a positive-definite matrix (i.e. its Hermitian part $H = \frac{1}{2}(A + A^*)$ is positive-definite), $B \in \mathbb{C}^{m\times n}$ $(m \leq n)$ with rank $r_B \leq m$, $C \in \mathbb{C}^{m\times m}$ is a Hermitian positive-semidefinite matrix and satisfies $N(B^*) \cap N(C) = \{0\}$, and $x, f \in \mathbb{C}^n$, $y, g \in \mathbb{C}^m$. Here, $(\cdot)^*$ represents the conjugate transpose of the corresponding vector or matrix, $N(\cdot)$ denotes the null space of the corresponding matrix. These assumptions guarantee the existence and uniqueness of the solution of the system of linear equation (1.1) [4,5]. Linear system (1.1) has plentiful backgrounds in many scientific computing and engineering applications, such as piezoelectric structure computation [7,17,18], the constrained optimization problems [24], the nonlinear image restoration problem [3], computational fluid dynamics [20] and so on.

Many efficient iteration methods have been studied to solve the generalized saddle point linear system (1.1). To accelerate the convergence rates of these iteration methods, a lot of preconditioners have been constructed based on the special structure of the generalized saddle point matrix. For example, Hermitian and skew-Hermitian splitting (HSS) iteration method [1] and its variants [4,13,25,26], Uzawa-type methods [2,8,35] , the extension

Communicated by NM Bujurke.

L. Meng (✉)· S.-X Miao · J. Li
College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, People's Republic of China
E-mail: menglsh@nwnu.edu.cn

of the positive-definite and skew-Hermitian splitting method [27] and the like. It is noticed that, the relaxed preconditioner firstly studied in [6] have attracted considerable attention in the solution method for saddle point problems, especially for improving the convergence of the GMRES method. Then this idea was extended to relaxed HSS preconditioner [10,13,25] and the relaxed DPSS preconditioner [11,28,31,32]. For the relaxed positive-semidefinite and skew-Hermitian splitting preconditioner, one can see [14,15,33].

Applying positive-semidefinite splitting ideology, Cao [16] gave the following splitting

$$\mathscr{A} = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & B^* \\ -B & C \end{pmatrix} = \mathscr{P}_1 + \mathscr{P}_2. \tag{1.2}$$

It is a fact that both $\mathscr{P}_1$ and $\mathscr{P}_2$ are positive-semidefinite matrices by assumptions. Next, similar to the HSS iteration method, Cao established the block positive-semidefinite splitting (BPS) iteration method which is unconditionally convergent and presented the BPS preconditioner as follows:

$$\mathscr{M}_{BPS} = \frac{1}{2\alpha}(\alpha I + \mathscr{P}_1)(\alpha I + \mathscr{P}_2) = \frac{1}{2\alpha}\begin{pmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & \alpha I + C \end{pmatrix}, \tag{1.3}$$

where α is a real positive constant and I denotes the identity matrix with proper size. Finally, he showed the fact that all eigenvalues of the BPS preconditioned matrix are clustered at only two points as the iteration parameter α is close to zero by theoretical analysis.

In this paper, based on the BPS preconditioner, we propose a generalized relaxed BPS (GRBPS) preconditioner for generalized saddle point linear system. It is the fact that the GRBPS preconditioner can greatly improve the computing efficiency. Besides, theoretical analysis shows that the GRBPS preconditioned matrix also has clustered eigenvalue distribution when the two parameters are closed to zero.

The remainder of this paper is organized as follows. In Section 2, we propose the GRBPS preconditioner on the basis of the BPS preconditioner. In Section 3, we analyze the spectral properties of the GRBPS preconditioned matrix. A strategy for selecting the quasi-optimal parameter of the GRBPS preconditioner is discussed in Section 4. Section 5 is devoted to giving a numerical example that arise from the Navier-Stokes equation to illustrate the effectiveness of the proposed preconditioner, and we briefly state our conclusions in Section 6.

2 Generalized relaxed block positive-semidefinite splitting preconditioner

In this section, we establish a preconditioner for the Krylov subspace methods. First of all, let us recall the block positive-semidefinite splitting (BPS) preconditioner (1.3). When the BPS splitting matrix $\mathscr{M}_{BPS}$ is served as a preconditioner, the pre-factor has no effect on the preconditioned system. For convenience, we replace it by the following preconditioner:

$$\hat{\mathscr{M}}_{BPS} = \frac{1}{\alpha}\begin{pmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & \alpha I + C \end{pmatrix} = \begin{pmatrix} \alpha I + A & B^* + \frac{1}{\alpha}AB^* \\ -B & \alpha I + C \end{pmatrix},$$

and the difference between the preconditioner $\hat{\mathscr{M}}_{BPS}$ and the generalized saddle point matrix $\mathscr{A}$ is

$$\hat{\mathscr{N}}_{BPS} = \hat{\mathscr{M}}_{BPS} - \mathscr{A} = \begin{pmatrix} \alpha I & \frac{1}{\alpha}AB^* \\ 0 & \alpha I \end{pmatrix}.$$

It is observed from $\hat{\mathscr{N}}_{BPS}$ that the diagonal blocks tends to zero matrices as the parameter α tends to zero, while the (1,2)-block goes to infinity. Therefore, we must choose an appropriate parameter α to balance the diagonal and the off-diagonal parts.

We hope that the preconditioner and the coefficient matrix $\mathscr{A}$ should be as close as possible so that the eigenvalue distribution of the preconditioned matrix is more clustered. From this point of view, the BPS preconditioner $\mathscr{M}_{BPS}$ may not be a good approximation to the generalized saddle point matrix. Inspired by the relaxed matrix splitting technique, in order to get a better approximation to the generalized saddle point matrix $\mathscr{A}$, it produces the following relaxed BPS (RBPS) preconditioner

$$\mathscr{M}_{RBPS} = \frac{1}{\alpha}\begin{pmatrix} A & 0 \\ 0 & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & \alpha I + C \end{pmatrix} = \begin{pmatrix} A & \frac{1}{\alpha}AB^* \\ -B & \alpha I + C \end{pmatrix}. \tag{2.1}$$

Next, in order to further improve the performance of the RBPS preconditioner, we propose a generalized RBPS (GRBPS) preconditioner:

$$\mathscr{M}_{GRBPS} = \frac{1}{\alpha}\begin{pmatrix} A & 0 \\ 0 & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & \beta I + C \end{pmatrix} = \begin{pmatrix} A & \frac{1}{\alpha}AB^* \\ -B & \beta I + C \end{pmatrix}. \tag{2.2}$$

Based on the matrix (2.2), it follows that

$$\mathscr{N}_{GRBPS} = \mathscr{M}_{GRBPS} - \mathscr{A} = \begin{pmatrix} 0 & \frac{1}{\alpha}AB^* - B^* \\ 0 & \beta I \end{pmatrix}. \tag{2.3}$$

It is obvious that the preconditioner $\mathscr{M}_{GRBPS}$ is more closer to saddle point matrix $\mathscr{A}$ than the preconditioner $\mathscr{M}_{RBPS}$ as parameter α is constant and $\beta \to 0$.

In practical implementation of the $\mathscr{M}_{GRBPS}$ within Krylov subspace acceleration, we need to solve the residual equation

$$\mathscr{M}_{GRBPS} z = \frac{1}{\alpha}\begin{pmatrix} A & 0 \\ 0 & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & \beta I + C \end{pmatrix} z = r.$$

By using of the matrix factorization:

$$\begin{pmatrix} \alpha I & B^* \\ -B & \beta I + C \end{pmatrix} = \begin{pmatrix} I & 0 \\ -\frac{1}{\alpha}B & I \end{pmatrix}\begin{pmatrix} \alpha I & 0 \\ 0 & \beta I + C + \frac{1}{\alpha}BB^* \end{pmatrix}\begin{pmatrix} I & \frac{1}{\alpha}B^* \\ 0 & I \end{pmatrix}.$$

Then we have the following algorithmic implementation of the GRBPS preconditioner (2.2):

Algorithm 1 (The GRBPS preconditioning): Solution of $\mathscr{M}_{GRBPS} z = r$, where $r = (r_1; r_2)$ and $z = (z_1; z_2)$ are given residual vector and the current vector, respectively; and $r_1,\ z_1 \in \mathbb{C}^n$, $r_2,\ z_2 \in \mathbb{C}^m$, from the following procedures:

(1) solve t_1 from $At_1 = r_1$;
(2) solve z_2 from $(\beta I + C + \frac{1}{\alpha}BB^*)z_2 = r_2 + Bt_1$;
(3) compute $z_1 = t_1 - \frac{1}{\alpha}B^* z_2$.

In analogy with the GRBPS preconditioner, the implementation of the BPS preconditioner with a Krylov subspace methods in [16] can be described as follow:

Algorithm 2 (The BPS preconditioning): Solution of $\mathscr{M}_{BPS} z = r$, where $r = (r_1; r_2)$ and $z = (z_1; z_2)$ are given residual vector and the current vector, respectively; and $r_1,\ z_1 \in \mathbb{C}^n$, $r_2,\ z_2 \in \mathbb{C}^m$, from the following procedures:

(1) solve t_1 from $(\alpha I + A)t_1 = 2\alpha r_1$;
(2) solve z_2 from $(\alpha I + C + \frac{1}{\alpha}BB^*)z_2 = 2r_2 + \frac{1}{\alpha}Bt_1$;
(3) compute $z_1 = \frac{1}{\alpha}(t_1 - B^* z_2)$.

From these algorithms, we see that there are two linear subsystems needed to be solved at steps (1) and (2). Besides, we find that these algorithms almost keep the same computational cost for using the Krylov subspace iteration methods to solve the generalized saddle point linear system (1.1). However, $\mathscr{M}_{GRBPS}$ is much closer to the coefficient matrix $\mathscr{A}$ than the preconditioner $\mathscr{M}_{BPS}$ within the appropriate parameters. That is to say, the GRBPS preconditioner could be having better preconditioning effect.

3 Spectral properties of preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$

The spectral distribution of the preconditioned matrix relates closely to the convergence rate of Krylov subspaces methods. Positive real spectrum or tightly clustered spectrum of the preconditioned matrix are desirable. Next, we analyze some properties of the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$.

The spectral characteristic of $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ is described as follows:

Theorem 3.1 *Assume that $A \in \mathbb{C}^{n\times n}$ is a positive definite matrix and $B \in \mathbb{C}^{m\times n}$ is a rectangular matrix, $C \in \mathbb{C}^{m\times m}$ is a Hermitian positive-semidefinite matrix satisfying $N(B^*) \cap N(C) = \{0\}$. Let α, β be given real positive constants and the GRBPS preconditioner be defined as in (2.2). Then the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ has an eigenvalue 1 with algebraic multiplicity at least n, the remaining eigenvalues are the same as those of the matrix $S_C^{-1}(C + BA^{-1}B^*)$, where $S_C = \beta I + C + \frac{1}{\alpha}BB^*$.*

Proof By direct calculations, we can verify that

$$\mathscr{M}_{GRBPS}^{-1} = \alpha \begin{pmatrix} \frac{1}{\alpha}I - \frac{1}{\alpha^2}B^*S_C^{-1}B & -\frac{1}{\alpha}B^*S_C^{-1} \\ \frac{1}{\alpha}S_C^{-1}B & S_C^{-1} \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & \frac{1}{\alpha}I \end{pmatrix}, \tag{3.1}$$

where $S_C = \beta I + C + \frac{1}{\alpha}BB^*$. Besides, it holds that

$$\mathscr{M}_{GRBPS}^{-1}\mathscr{A} = I - \mathscr{M}_{GRBPS}^{-1}\mathscr{N}_{GRBPS}$$

with

$$\begin{aligned} \mathscr{M}_{GRBPS}^{-1}\mathscr{N}_{GRBPS} &= \alpha \begin{pmatrix} \frac{1}{\alpha}I - \frac{1}{\alpha^2}B^*S_C^{-1}B & -\frac{1}{\alpha}B^*S_C^{-1} \\ \frac{1}{\alpha}S_C^{-1}B & S_C^{-1} \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{\alpha}B^* - A^{-1}B^* \\ 0 & \frac{\beta}{\alpha}I \end{pmatrix} \\ &= \begin{pmatrix} 0 & -A^{-1}B^* + \frac{1}{\alpha}B^*S_C^{-1}(C + BA^{-1}B^*) \\ 0 & I - S_C^{-1}(C + BA^{-1}B^*) \end{pmatrix}. \end{aligned}$$

So we obtain

$$\mathscr{M}_{GRBPS}^{-1}\mathscr{A} = \begin{pmatrix} I & A^{-1}B^* - \frac{1}{\alpha}B^*S_C^{-1}(C + BA^{-1}B^*) \\ 0 & S_C^{-1}(C + BA^{-1}B^*) \end{pmatrix}. \tag{3.2}$$

It is obvious form (3.2) that the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ has an eigenvalue 1 with multiplicity at least n, the rest of eigenvalues are the same as those of matrix $S_C^{-1}(C + BA^{-1}B^*)$. □

Similar to the methods in [16], we can further study the clustering property of the eigenvalues of the GRBPS preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$, which is shown in the following theorem.

Theorem 3.2 *Assume that the conditions in Theorem 3.1 are satisfied. Then the non-unit eigenvalues of the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ are clustered at 0_+ with multiplicity r_B and 1_- with multiplicity $n+m-r_B$ if parameters both α and β are close to zero.*

Proof Based on the definition of the GRBPS preconditioner $\mathscr{M}_{GRBPS}$, we have

$$\mathscr{M}_{GRBPS}^{-1}\mathscr{A} = \alpha \begin{pmatrix} \alpha I & B^* \\ -B & \beta I + C \end{pmatrix}^{-1} \begin{pmatrix} A^{-1} & 0 \\ 0 & \frac{1}{\alpha}I \end{pmatrix} \begin{pmatrix} A & B^* \\ -B & C \end{pmatrix}.$$

It is obvious that $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ is similar to

$$\begin{aligned} \hat{\mathscr{A}} &= \alpha \begin{pmatrix} A^{-1} & 0 \\ 0 & \frac{1}{\alpha}I \end{pmatrix} \begin{pmatrix} A & B^* \\ -B & C \end{pmatrix} \begin{pmatrix} \alpha I & B^* \\ -B & \beta I + C^{ll} \end{pmatrix}^{-1} \\ &= \alpha \begin{pmatrix} A^{-1} & 0 \\ 0 & \frac{1}{\alpha}I \end{pmatrix} \begin{pmatrix} A & B^* \\ -B & C \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha}I - \frac{1}{\alpha^2}B^*S_C^{-1}B & -\frac{1}{\alpha}B^*S_C^{-1} \\ \frac{1}{\alpha}S_C^{-1}B & S_C^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \alpha A^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha}A + \frac{1}{\alpha}(I - \frac{1}{\alpha}A)B^*S_C^{-1}B & (I - \frac{1}{\alpha}A)B^*S_C^{-1} \\ -\frac{\beta}{\alpha}S_C^{-1}B & I - \beta S_C^{-1} \end{pmatrix}, \end{aligned} \tag{3.3}$$

where $S_C = \beta I + C + \frac{1}{\alpha}BB^*$.

Let $B = U\Sigma V^*$ be the singular value decomposition of the matrix B, where $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ are unitary matrices, and Σ is a nonnegative diagonal matrix. Because $B \in \mathbb{C}^{m\times n}$ is a rectangular matrix with rank $r_B \le m$, then the above singular value decomposition can equivalently be reformulated into

$$B = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_{r_B} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix}, \tag{3.4}$$

where $\Sigma_{r_B} = diag(\sigma_1, \cdots, \sigma_{r_B})$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r_B} > 0$ being ordered nonzero singular values, where $U_1 \in \mathbb{C}^{m\times r_B}$, $U_2 \in \mathbb{C}^{m\times(m-r_B)}$; $V_1 \in \mathbb{C}^{n\times r_B}$, $V_2 \in \mathbb{C}^{n\times(n-r_B)}$. Then we have

$$S_C = \beta I + C + \frac{1}{\alpha}BB^* = U(\beta I + U^*CU + \frac{1}{\alpha}\Sigma^2)U^* = USU^*$$

where

$$S = \begin{pmatrix} \beta I + U_1^*CU_1 + \frac{1}{\alpha}\Sigma_{r_B}^2 & U_1^*CU_2 \\ U_2^*CU_1 & \beta I + U_2^*CU_2 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}.$$

Denote by

$$S^{-1} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}^{-1} = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} = T.$$

By direct computations, we obtain

$$\begin{aligned} T_{11} &= S_{11}^{-1} + S_{11}^{-1}S_{12}(S_{22} - S_{21}S_{11}^{-1}S_{12})^{-1}S_{21}S_{11}^{-1} \\ &= S_{11}^{-1} + S_{11}^{-1}U_1^*CU_2(\beta I + U_2^*CU_2 - U_2^*CU_1S_{11}^{-1}U_1^*CU_2)^{-1}U_2^*CU_1S_{11}^{-1}, \end{aligned}$$

$$\begin{aligned} T_{12} &= -S_{11}^{-1}S_{12}(S_{22} - S_{21}S_{11}^{-1}S_{12})^{-1} \\ &= -S_{11}^{-1}U_1^*CU_2(\beta I + U_2^*CU_2 - U_2^*CU_1S_{11}^{-1}U_1^*CU_2)^{-1}, \end{aligned}$$

$$\begin{aligned} T_{21} &= -(S_{22} - S_{21}S_{11}^{-1}S_{12})^{-1}S_{21}S_{11}^{-1} \\ &= -(\beta I + U_2^*CU_2 - U_2^*CU_1S_{11}^{-1}U_1^*CU_2)^{-1}U_2^*CU_1S_{11}^{-1} \end{aligned}$$

and

$$\begin{aligned} T_{22} &= (S_{22} - S_{21}S_{11}^{-1}S_{12})^{-1} \\ &= (\beta I + U_2^*CU_2 - U_2^*CU_1S_{11}^{-1}U_1^*CU_2)^{-1}. \end{aligned}$$

Besides, observing the right matrix in (3.3), we need calculate

$$\begin{aligned} B^*S_C^{-1}B &= V\Sigma^*U^*UTU^*U\Sigma V^* \\ &= \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} \Sigma_{r_B} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \begin{pmatrix} \Sigma_{r_B} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix} \\ &= V_1\Sigma_{r_B}T_{11}\Sigma_{r_B}V_1^*, \end{aligned}$$

$$\begin{aligned} B^*S_C^{-1} &= V\Sigma^*U^*UTU^* \\ &= \begin{pmatrix} V_1 & V_2 \end{pmatrix} \begin{pmatrix} \Sigma_{r_B} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} U^* \\ &= V_1\Sigma_{r_B}T_2U^* \end{aligned}$$

and

$$\begin{aligned} S_C^{-1}B &= UTU^*U\Sigma V^* \\ &= U\begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \begin{pmatrix} \Sigma_{r_B} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix} \\ &= UT_1\Sigma_{r_B}V_1^*, \end{aligned}$$

where $T_1 = (T_{11}^* \;\; T_{21}^*)^*$, $T_2 = (T_{11} \;\; T_{12})$. Then the matrix $\hat{\mathscr{A}}$ can be written as

$$\hat{\mathscr{A}} = \begin{pmatrix} \alpha A^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha}A + \frac{1}{\alpha}(I - \frac{1}{\alpha}A)V_1\Sigma_{r_B}T_{11}\Sigma_{r_B}V_1^* & (I - \frac{1}{\alpha}A)V_1\Sigma_{r_B}T_2U^* \\ -\frac{\beta}{\alpha}UT_1\Sigma_{r_B}V_1^* & I - \beta UTU^* \end{pmatrix}.$$

Let

$$\hat{\mathscr{A}} = \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix} \bar{\mathscr{A}} \begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}^*,$$

where

$$\bar{\mathscr{A}} = \begin{pmatrix} \alpha A^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \frac{1}{\alpha} A + \frac{1}{\alpha}(I - \frac{1}{\alpha}A) V_1 \Sigma_{r_B} T_{11} \Sigma_{r_B} V_1^* & (I - \frac{1}{\alpha}A) V_1 \Sigma_{r_B} T_1^* \\ -\frac{\beta}{\alpha} T_1 \Sigma_{r_B} V_1^* & I - \beta T \end{pmatrix}.$$

Note that $U_2^* C U_2$ is nonsingular. Otherwise, there exists a nonzero vector z such that $U_2^* C U_2 z = 0$. So we know that $U_2 z \in N(C)$. By the singular value decomposition (3.4), we have that $B^* U_2 z = 0$, i.e., $U_2 z \in N(B^*)$. Then we have $U_2 z = 0$ by $N(B^*) \cap N(C) = \{0\}$, which implies $z = 0$. This contradicts the fact that z is a nonzero vector.

Now, letting $\alpha \to 0_+$, $\beta \to 0_+$, it follows that

$$\frac{1}{\alpha} S_{11}^{-1} = (\alpha\beta I + \alpha U_1^* C U_1 + \Sigma_{r_B}^2)^{-1} \to \Sigma_{r_B}^{-2},$$

which means $S_{11}^{-1} \to 0$. Thus, we can get

$$T_{11} \to 0, \quad T_{12} \to 0, \quad T_{21} \to 0 \text{ and } T_{22} \to (U_2^* C U_2)^{-1}.$$

In addition, it is easy to find that

$$\frac{1}{\alpha} T_{11} \to \frac{1}{\alpha} S_{11}^{-1} \to \Sigma_{r_B}^{-2}.$$

Furthermore, if $\alpha \to 0_+$, $\beta \to 0_+$, we obtain

$$\begin{aligned} \bar{\mathscr{A}} &= \begin{pmatrix} I + \frac{1}{\alpha}(\alpha A^{-1} - I) V_1 \Sigma_{r_B} T_{11} \Sigma_{r_B} V_1^* & (\alpha A^{-1} - I) V_1 \Sigma_{r_B} T_1^* \\ -\frac{\beta}{\alpha} T_1 \Sigma_{r_B} V_1^* & I - \beta T \end{pmatrix} \\ &\to \begin{pmatrix} I - V_1 V_1^* & 0 \\ 0 & I \end{pmatrix} = \tilde{\mathscr{A}}. \end{aligned}$$

Because

$$V^*(I - V_1 V_1^*) V = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix},$$

then the matrix $\tilde{\mathscr{A}}$ is similar to

$$\breve{\mathscr{A}} = \begin{pmatrix} 0_{r_B} & 0 & 0 \\ 0 & I_{n-r_B} & 0 \\ 0 & 0 & I_m \end{pmatrix}.$$

Since the GRBPS preconditioned matrix $\mathscr{M}_{GRBPS}^{-1} \mathscr{A}$ is similar to $\bar{\mathscr{A}}$, and $\tilde{\mathscr{A}}$ is similar to $\breve{\mathscr{A}}$, then we can draw conclusions that the eigenvalues of the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1} \mathscr{A}$ are clustered at 0_+ (i.e., gathering at 0 from the right side of 0) with multiplicity r_B and 1_- (i.e., gathering at 1 from the left side of 1) with multiplicity $n + m - r_B$ when both the parameters α and β are close to 0. □

It is well known that the convergence of Krylov subspace methods is not only related to the eigenvalue distribution of the preconditioned matrix, but also related to the number of corresponding linearly independent eigenvectors [30]. The eigenvector distribution of the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1} \mathscr{A}$ is presented in the following theorem.

Theorem 3.3 *Assume that the conditions of Theorem 3.1 hold. Let the GRBPS preconditioner $\mathscr{M}_{GRBPS}$ be defined in (2.2), then the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1} \mathscr{A}$ has $n + i + j$ $(0 \le i + j \le m)$ linearly independent eigenvectors. There are*

(1) n eigenvectors $[u_\ell; 0]$ $(\ell = 1, 2, \ldots, n)$ that correspond to the eigenvalue 1, where u_ℓ $(\ell = 1, 2, \ldots, n)$ are arbitrary linearly independent vectors.

(2) i $(0 \le i \le m)$ *eigenvectors* $[u_\ell^1; v_\ell^1]$ $(1 \le \ell \le i)$ *that correspond to the eigenvalue 1, where* $v_\ell^1 \in N((I - \alpha A^{-1})B^*)$, $i = dim\{N(I - \alpha A^{-1}) \cap R(B^*)\}$ *, and* u_ℓ^1 *are arbitrary vectors,* $R(\cdot)$ *denote the range space of the corresponding matrix..*

(3) j $(0 \le j \le m)$ *eigenvectors* $[u_\ell^2; v_\ell^2]$ $(1 \le \ell \le j)$ *that correspond to eigenvalues* $\lambda_\ell \ne 1$*, where* u_ℓ^2, v_ℓ^2 *satisfy the following equations:* $\lambda_\ell S_C v_\ell^2 = (C + BA^{-1}B^*)v_\ell^2$, $u_\ell^2 = \frac{1}{\lambda_\ell - 1}(A^{-1}B^* - \frac{1}{\alpha}B^* S_C^{-1}(C + BA^{-1}B^*))v_\ell^2$.

Proof Let λ be an eigenvalue of the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ and $[u; v]$ be the corresponding eigenvector. From (3.2), we have

$$\begin{pmatrix} I & A^{-1}B^* - \frac{1}{\alpha}B^* S_C^{-1}(C + BA^{-1}B^*) \\ 0 & S_C^{-1}(C + BA^{-1}B^*) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix}. \tag{3.5}$$

Note that the Eq.(3.5) can equivalently be reformulated into

$$\begin{cases} (1 - \lambda)u + (A^{-1}B^* - \frac{1}{\alpha}B^* S_C^{-1}(C + BA^{-1}B^*))v = 0, \\ (\lambda I - S_C^{-1}(C + BA^{-1}B^*))v = 0. \end{cases} \tag{3.6}$$

Firstly, if $\lambda = 1$, then (3.6) reduces to

$$\begin{cases} (A^{-1}B^* - \frac{1}{\alpha}B^* S_C^{-1}(C + BA^{-1}B^*))v = 0, \\ \{I - S_C^{-1}(C + BA^{-1}B^*)\}v = 0. \end{cases} \tag{3.7}$$

We can find that the Eq.(3.7) are true when $v = 0$. Therefore, there are n linearly independent eigenvectors $[u_\ell; 0]$ $(\ell = 1, 2, \ldots, n)$ corresponding to the eigenvalue 1, where $u_\ell (\ell = 1, 2, \ldots, n)$ are arbitrary linearly independent vectors. Next we discuss other possible cases for v. From the second equation in (3.7), we get that $v = S_C^{-1}(C + BA^{-1}B^*)v$. Substituting it into the first equation in (3.7), it holds that $(I - \alpha A^{-1})B^* v = 0$. Hence, it is obvious that the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$ has i $(0 \le i \le m)$ linearly independent eigenvectors $[u_\ell^1; v_\ell^1]$ $(1 \le \ell \le i)$ that correspond to the eigenvalue 1, where u_ℓ^1 are arbitrary vectors, $v_\ell^1 \in N((I - \alpha A^{-1})B^*)$, $i = dim\{N(I - \alpha A^{-1}) \cap R(B^*)\}$.

Secondly, we consider the case that $\lambda \ne 1$. From the first equation in (3.6), we have

$$u = \frac{1}{\lambda - 1}(A^{-1}B^* - \frac{1}{\alpha}B^* S_C^{-1}(C + BA^{-1}B^*))v. \tag{3.8}$$

If $v = 0$, then $u = 0$, which contradicts with $[u; v]$ being an eigenvector. Hence, v is not equal to zero vector. If there exists v that it satisfies the second and third equations in (3.6), then there will be j $(0 \le j \le m)$ linearly independent eigenvectors $[u_\ell^2; v_\ell^2]$ $(1 \le \ell \le j)$ that correspond to eigenvalues $\lambda \ne 1$. Here, the vectors v_ℓ^2 arises from $\lambda_\ell S_C v_\ell^2 = (C + BA^{-1}B^*)v_\ell^2$, and the vector u_ℓ^2 $(1 \le \ell \le j)$ satisfy (3.8).

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

$$\begin{pmatrix} u_1 & \cdots & u_n 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} k_1^1 \\ \vdots \\ k_n^1 \end{pmatrix} + \begin{pmatrix} u_1^1 & \cdots & u_i^1 \\ v_1^1 & \cdots & v_i^1 \end{pmatrix} \begin{pmatrix} k_1^2 \\ \vdots \\ k_i^2 \end{pmatrix} + \begin{pmatrix} u_1^2 & \cdots & u_j^2 \\ v_1^2 & \cdots & v_j^2 \end{pmatrix} \begin{pmatrix} k_1^3 \\ \vdots \\ k_j^3 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.9}$$

holds only when the vectors k^1, k^2 and k^3 are all zero vectors (denoted as $\mathbf{0}$). We know that the first matrix in Eq.(3.9) consists of the eigenvectors corresponding to the eigenvalue 1 for the case (1), the second matrix arises from the eigenvalue 1 with multiplicity i, and the last matrix produces from the eigenvalue $\lambda_\ell \ne 1$ $(\ell = 1, 2, \ldots, j)$. By multiplying both sides of (3.9) from left with $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$, we obtain

$$\begin{pmatrix} u_1 & \cdots & u_n \\ 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} k_1^1 \\ \vdots \\ k_n^1 \end{pmatrix} + \begin{pmatrix} u_1^1 & \cdots & u_i^1 \\ v_1^1 & \cdots & v_i^1 \end{pmatrix} \begin{pmatrix} k_1^2 \\ \vdots \\ k_i^2 \end{pmatrix} + \begin{pmatrix} u_1^2 & \cdots & u_j^2 \\ v_1^2 & \cdots & v_j^2 \end{pmatrix} \begin{pmatrix} \lambda_1 k_1^3 \\ \vdots \\ \lambda_j k_j^3 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.10}$$

By subtracting (3.10) from (3.9), it holds that

$$\begin{pmatrix} u_1^2 & \cdots & u_j^2 \\ v_1^2 & \cdots & v_j^2 \end{pmatrix} \begin{pmatrix} (\lambda_1 - 1)k_1^3 \\ \vdots \\ (\lambda_j - 1)k_j^3 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.11}$$

Since the left matrix in (3.11) means that there are j linearly independent vectors and the eigenvalues $\lambda_\ell \neq 1$ $(\ell = 1, 2, \ldots, j)$, we know that $k_\ell^3 = 0$ $(\ell = 1, 2, \ldots, j)$. Substituting $k^3 = \mathbf{0}$ into (3.9), we know that k^1 and k^2 are both equal to zero vectors. Therefore, the $n + i + j$ eigenvectors are linearly independent. □

The purpose of preconditioning is to make the eigenvalue distribution more clustered and to reduce the iteration numbers within certain tolerance. The solvers usually use Krylov subspace methods to solve the corresponding problems because of its well properties, such as, the degree of the minimal polynomial is equal to the dimension of the corresponding Krylov subspace, which were found to be extremely useful when combined with an appropriate preconditioner in solving the underlying system [30]. In the following, we study an upper bound of the degree of the minimal polynomial of the preconditioned matrix $\mathscr{M}_{GRBPS}^{-1}\mathscr{A}$.

Theorem 3.4 *Assume that the conditions of Theorem 3.1 hold. Then the dimension of the Krylov subspace $\mathcal{K}(\mathscr{M}_{GRBPS}^{-1}\mathscr{A}, b)$ is at most $m + 1$.*

The proof process is similar to [10, Theorem 3.4] and [12, Theorem 3.3], Therefore we omit it.

4 Parameter selection strategy

In this section, we will discuss the choice strategy of optimal and feasible parameter of the GRBPS preconditioner. Recently, many solvers have studied a lot of effective methods how to estimating the iteration parameter [16,19, 23,29,34]. Through the ideas in [16,23], we can get the new estimation formulas for the parameters α and β of the GRBPS preconditioner.

First of all, through the expression of GRBPS preconditioner, we can find that the difference between $\mathscr{M}_{GRBPS}$ and the generalized saddle point matrix $\mathscr{A}$ is (2.3). To make the preconditioning effect of preconditioner become better, we need to select suitable parameter so that the preconditioner would be as close as possible to coefficient matrix, i.e., $\mathscr{N}_{RBPS} \approx 0$.

We denote

$$\Phi(\alpha, \beta) = \|\mathscr{N}_{GRBPS}\|_F = \left\| \begin{pmatrix} 0 & \frac{1}{\alpha}AB^* - B^* \\ 0 & \beta I \end{pmatrix} \right\|_F .$$

Thus, we have

$$\begin{aligned} \Phi^2(\alpha, \beta) &= \left\| \frac{1}{\alpha}AB^* - B^* \right\|_F^2 + \beta^2 \|I\|_F^2 = \left\| \frac{1}{\alpha}AB^* - B^* \right\|_F^2 + \beta^2 m \\ &= tr\left(\left(\frac{1}{\alpha}AB^* - B^* \right) \left(\frac{1}{\alpha}BA^* - B \right) \right) + \beta^2 m \\ &= \frac{1}{\alpha^2}\|AB^*\|_F^2 - \frac{2}{\alpha}\mathrm{Re}(tr(AB^*B)) + tr(B^*B) + \beta^2 m. \end{aligned}$$

To calculate the quasi-optimal parameters of the GRBPS preconditioner, we can regard parameter β as a constant which is close to 0 and then analyze the value of parameter α. By minimizing the function $\Phi^2(\alpha, \beta)$, We get the following the expression of the quasi-optimal parameter α of the GRBPS preconditioner:

$$\alpha_{est}^0 = \frac{\|AB^*\|_F^2}{\mathrm{Re}(tr(AB^*B))}. \tag{4.1}$$

It is easy to find that the GRBPS preconditioner reduces to the RBPS preconditioner (2.1) as parameters $\alpha = \beta$, in this case, the quasi-optimal parameter α of the GRBPS preconditioner satisfies the quaternary linear equation:

$$\alpha^4 + d\alpha + e = 0$$

with $d = \mathrm{Re}(tr(AB^*B))/m$, $e = -\|AB^*\|_F^2/m$. It is the fact that the parameter α is positive constant, then we can get the expressions of α as follows:

$$\alpha_{est}^1 = -\frac{1}{2}\sqrt{\Delta} + \frac{1}{2}\sqrt{-\Delta + \frac{8d}{4\sqrt{\Delta}}}, \quad \alpha_{est}^2 = \frac{1}{2}\sqrt{\Delta} + \frac{1}{2}\sqrt{-\Delta - \frac{8d}{4\sqrt{\Delta}}}, \tag{4.2}$$

where $\Delta = \frac{\sqrt[3]{2}\Delta_1}{3\sqrt[3]{\Delta_2+\sqrt{-4\Delta_1^3+\Delta_2^2}}} + \frac{\sqrt[3]{\Delta_2+\sqrt{-4\Delta_1^3+\Delta_2^2}}}{3\sqrt[3]{2}}$ with $\Delta_1 = 12e$ and $\Delta_2 = 27d^2$ (It should be noted that the other two roots of the equation are negative values, so we discard them.).

5 Numerical experiment

In this section, we present an example to show the effectiveness of the new preconditioner $\mathscr{M}_{GRBPS}$ for the generalized saddle point linear system (1.1). In actual implementation, the initial guess is chosen to be the zero vector, the exact solution is the vector with all its element being ones and $b = \mathscr{A} \cdot \mathbf{1}$. The iteration is terminated if the current iterations satisfied

$$\mathrm{RES} = \frac{\| b - \mathscr{A}u^k \|}{\| b - \mathscr{A}u^0 \|} \le 10^{-6}$$

or if the number of the prescribed iteration steps (denoted by 'IT') $K_{max} = 1500$ is exceed. All experiments are performed in MATLAB (version R2016b) in double precision on a personal computer with 1.80GHz central processing unit (Intel(R) Core(TM) i7-8565U), 4.00 GB memory and Windows 10 operating system. To see the quality of computed solutions, we also define the relative errors of the computed approximations as ERR, i.e.,

$$\mathrm{ERR} = \frac{\|u^* - u^k\|}{\|u^*\|}$$

are also reported with u^* representing the initially known exact solution.

Besides, we also test DPSS preconditioner [28,31] and RDPSS preconditioner [12,32] for generalized saddle point system, the forms of those are as follows:

$$\mathscr{M}_{DPSS} = \frac{1}{2\alpha}\begin{pmatrix} \alpha I + A & 0 \\ 0 & \alpha I + C \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & \alpha I \end{pmatrix},$$

$$\mathscr{M}_{RDPSS} = \frac{1}{\alpha}\begin{pmatrix} A & 0 \\ 0 & \alpha I \end{pmatrix}\begin{pmatrix} \alpha I & B^* \\ -B & C \end{pmatrix}.$$

Furthermore, to better implement the discussed four preconditioners, we should select suitable parameters to solve the linear sub-systems. We use the quasi-optimal parameter given in [16, Table 1 and 2] for BPS and DPSS preconditioner, optimal parameter α is found by experimentally test for the RDPSS preconditioner, for the GRBPS preconditioner, we use parameter α_{est} given in section 4. The linear sub-systems are solved by the sparse Cholesky factorization if the coefficient matrix is Hermitian positive-definite and the sparse LU factorization if the coefficient matrix is non-Hermitian positive-definite. We use the left preconditioned GMRES iteration method to solve the problem of example.

Example [16]. Consider the following two-dimensional linearized Navier-Stokes equation: Find u and p such that

$$\begin{cases} -\upsilon\Delta u + \omega \cdot \nabla u + \nabla p = f \\ \nabla \cdot u = 0 \end{cases} \quad \text{in } \Omega \tag{5.1}$$

where $\Omega = [0, 1] \times [0, 1] \in \mathbb{R}^2$ is a bounded domain, $\upsilon > 0$ is the kinematic viscosity, u is a vector-valued function representing the velocity, ω is the known velocity from the previous Picard iteration, p is a scalar function representing the pressure, Δ is the componentwise Laplace operator, ∇ and $\nabla\cdot$ denote the gradient and the divergence operators.

Here, we use the IFISS package developed by Elman et al. [21] to generate the test generalized saddle point linear system. In particular, we choose the "leaky" lid-driven cavity problem. In this problem, a unit velocity is specified along the entire top surface and zero velocity on the other surfaces, i.e. the boundary conditions are $u_x = u_y = 0$ on three fixed walls ($x = 0,\ y = 0$ and $x = 0$) and $u_x = 1,\ u_y = 0$ on the moving wall ($y = 1$) [9,22]. The uniform grids of five increasing sizes (8×8, 16×16, 32×32, 64×64 and 128×128) are used to discretize the bounded domain. For each grids, we take two viscosities, i.e., $\upsilon = 1$ and $\upsilon = 0.1$. The $Q1 - P0$ mixed finite element method with local stabilization method is used to discretize the linearized Navier-Stokes equation (5.1). Here, the stabilized parameter is taken as 0.25. Note that the discretized coefficient matrix is singular. To get a nonsingular generalized saddle point linear system, we often drop the last two columns of the matrix B and the last two rows and two columns of C [12,31]. The final generalized saddle point matrix $\mathscr{A}$ satisfies: the (1,1)-block matrix A is nonsymmetric positive-definite, the submatrix B has full row rank and the submatrix C is symmetric positive-semidefinite.

In Tables 1 and 2, we list the values of the parameter α for the preconditioners $\mathscr{M}_{BPS}$, $\mathscr{M}_{DPSS}$, $\mathscr{M}_{RDPSS}$ and $\mathscr{M}_{GRBPS}$, the number of iteration steps 'IT', the CPU times and the corresponding residuals 'RES' and relative errors 'ERR' of PGMRES method incorporated with $\mathscr{M}_{BPS}$, $\mathscr{M}_{DPSS}$, $\mathscr{M}_{RDPSS}$ and $\mathscr{M}_{GRBPS}$. Besides, we also list the iteration steps of the no preconditioning GMRES iteration method (denoted as 'I') to show advantages of the four preconditioners. The difference of these two tables is that Table 1 lists the numerical results for Navier-Stokes equation with $\upsilon = 1$, while Table 2 lists those with $\upsilon = 0.1$. In the column of GRBPS preconditioner under each grid, the parameter α in the first row is calculated by α^0_{est} in (4.1), the parameter α in the second row is calculated by α^1_{est} or α^2_{est} in (4.2) (It should be noted that there are two estimation parameters for the GRBPS preconditioner when $\alpha = \beta$, and we only list the value of parameter with better numerical effect in tables by numerical examples), and optimal α in the third row is found by numerical test. Especially, through the proof analysis of Theorem 3.2, we know that the spectrum of preconditioned matrix $\mathscr{M}^{-1}_{GRBPS}\mathscr{A}$ is more clustered when both the parameters α and β of GRBPS preconditioner tend to zero, see the pictures for details.

From Table 1, it is the fact that the preconditioning effect of the GRBPS preconditioner is more than those of another preconditioners $\mathscr{M}_{BPS}$, $\mathscr{M}_{DPSS}$ and $\mathscr{M}_{RDPSS}$. When the four preconditioners select their own calculated optimal parameters, the preconditioning effects of $\mathscr{M}_{BPS}$ and $\mathscr{M}_{DPSS}$ are almost the same, while the preconditioning effects of $\mathscr{M}_{RDPSS}$ is more than that of $\mathscr{M}_{GRBPS}$ when the grid is less than 64×64. Specifically, the approximate level between the numerical solution and the exact solution obtained by the four preconditioning methods is almost the same when the dimension is small. With the dimension increases, ERR of the preconditioning methods under the preconditioners $\mathscr{M}_{BPS}$ and $\mathscr{M}_{DPSS}$ are getting bigger, which is because the iteration has satisfied the condition RES$< 10^{-6}$ before the iteration solution is well approximate to the exact solution in the iteration process of the two preconditioning methods. Even under the large dimension and stopping criterion, the numerical solutions obtained by the two preconditioning methods with the preconditioners $\mathscr{M}_{RDPSS}$ and $\mathscr{M}_{GRBPS}$ can still approximate the exact solution well. Besides, we find that GRBPS preconditioner with calculated optimal parameters take more time than that of RDPSS preconditioner as grid is bigger. Nevertheless, the effect of GRBPS preconditioned GMRES method still exceeds that of BPS preconditioned GMRES method, and the former can show better performance than RDPSS method under the test of optimal parameters.

From Table 2, we see that all the preconditioners improve the performance of the preconditioned GMRES method significantly. What's more, the preconditioning effect of $\mathscr{M}_{GRBPS}$ is better than that of the another three preconditioners when the actual optimal parameters are selected with grids being 8, 16 and 32. We also find from the value of ERR that all preconditioning methods can obtain good approximate solutions under the given stopping criterion. In addition, it can be found that α^1_{est} in (4.2) is more effective than α^0_{est} in (4.1) for the column of GRBPS preconditioner under each gradient, and when GRBPS preconditioner selects parameter α^1_{est}, its preconditioning effect is better than that of the BPS preconditioner under calculated optimal parameter, which indicates that α^1_{est} is effective. Furthermore, the preconditioning effect of $\mathscr{M}_{GRBPS}$ with parameters obtained from the experimentally test is still advantage over that of another preconditioners with larger grids.

To verify this desirable property, in Figures 1 and 2, we plot the eigenvalue distribution of four preconditioned matrices when $\upsilon = 1$ and $\upsilon = 0.1$, respectively. In addition, for the GRBPS preconditioned matrix, we show two cases, one case is when the iteration parameters $\alpha = \beta$ taken as in Table 1, the other case is when iteration parameters $\alpha \neq \beta$ with α, β taken as close to 0. We observe that the all preconditioners improve the eigenvalue distribution of the original matrix greatly. Besides, in agreement with the theory, the eigenvalue of GRBPS preconditioned matrix is clustered at two points 0_+ and 1_- as the iteration parameters are close to zero.

To further show the impact of the iteration parameters on the iteration counts of different case of the GRBPS preconditioned GMRES iteration methods, we plot the steps as functions of α in Figure 3. It is need to explained that when the parameter α is not equal to β, we fix $\beta = 10^{-7}$ and then depict the iteration curves with varying α. We can find from those figures that, the choice of $\alpha \neq \beta$ is better than that of $\alpha = \beta$ in the case of $\nu = 1$. At this

Table 1 The numerical results for preconditioned GMRES methods with I, $\mathscr{M}_{BPS}$, $\mathscr{M}_{DPSS}$, $\mathscr{M}_{RDPSS}$ and $\mathscr{M}_{GRBPS}$ with different grids for Example with $v = 1$

	I	$\mathscr{M}_{BPS}$					$\mathscr{M}_{DPSS}$					$\mathscr{M}_{RDPSS}$					$\mathscr{M}_{GRBPS}$				
Grid	IT	α	IT	RES	ERR	CPU	α	IT	RES	ERR	CPU	α	IT	RES	ERR	CPU	(α, β)	IT	RES	ERR	CPU
8×8	77	0.1450	33	9.1e-07	1.6e-06	0.0177	0.25	33	8.8e-07	3.2e-06	0.0273	2	11	4.8e-07	4.1e-07	0.02086	$(6.1635,10^{-7})$	15	4.4e-07	2.2e-07	0.01569
																	(1.7762,1.7762)	22	6.9e-07	8.7e-06	0.0063
																	$(2.5,10^{-7})$	11	9.1e-07	7.5e-07	0.0032
16×16	171	0.0776	49	8.5e-07	2.3e-06	0.0529	0.1	49	8.4e-07	2.7e-06	0.0746	0.5	17	9.6e-07	1.9e-06	0.0733	$(6.2706,10^{-7})$	21	8.5e-07	9.5e-07	0.0438
																	(1.3395,1.3395)	25	5.8e-07	7.6e-05	0.0209
																	$(1.5,10^{-7})$	16	8.4e-07	2.7e-06	0.0054
32×32	368	0.0402	71	6.6e-07	2.4e-06	0.2388	0.05	70	9.1e-07	9.8e-06	0.2066	0.5	25	9.3e-07	4.6e-06	0.0883	$(6.3213,10^{-7})$	32	9.9e-07	3.0e-06	0.0801
																	(0.9830,0.9830)	26	6.6e-07	7.9e-03	0.0529
																	$(1.8,10^{-7})$	24	9.4e-07	8.8e-06	0.0428
64×64	404	0.0205	99	9.0e-07	2.9e-05	1.0455	0.04	67	9.6e-07	5.6e-01	0.6056	1	36	9.4e-07	2.5e-05	0.2755	$(6.3460,10^{-7})$	53	5.7e-07	1.8e-06	0.4225
																	(0.7107,0.001)	18	5.8e-07	3.6e-05	0.1376
128×128	657	0.0103	74	9.9e-07	5.7e-01	3.5466	0.008	73	9.4e-07	5.7e-01	3.2398	1.2	50	9.9e-07	3.8e-05	2.1135	$(3.1791,10^{-7})$	69	8.4e-07	3.3e-05	3.2020
																	(0.2547,0.0001)	37	8.7e-07	3.7e-05	1.4372

Table 2 The numerical results for preconditioned GMRES methods with I, $\mathscr{M}_{BPS}$, $\mathscr{M}_{DPSS}$, $\mathscr{M}_{RDPSS}$ and $\mathscr{M}_{GRBPS}$ with different grids for Example with $\upsilon = 0.1$

Grid	I IT	$\mathscr{M}_{BPS}$ α	IT	RES	ERR	CPU	$\mathscr{M}_{DPSS}$ α	IT	RES	ERR	CPU	$\mathscr{M}_{RDPSS}$ α	IT	RES	ERR	CPU	$\mathscr{M}_{GRBPS}$ (α, β)	IT	RES	ERR	CPU
8×8	84	0.0542	20	6.7e-07	8.3e-07	0.0147	0.11	20	8.1e-07	1.0e-06	0.0167	0.1	15	4e-07	4.6e-07	0.0143	$(0.6195,10^{-7})$	16	4.4e-07	6.2e-07	0.0031
																	(0.4457,0.4457)	15	3.8e-07	1.0e-06	0.0027
16×16	157	0.0199	32	9.3e-07	1.5e-06	0.0456	0.041	32	9.7e-07	1.6e-06	0.0552	0.1	21	9.5e-07	1.5e-06	0.0513	$(0.6282,10^{-7})$	24	6.5e-07	1.1e-06	0.0419
																	(0.3682,0.3682)	18	5.5e-07	4.3e-06	0.0062
32×32	275	0.0075	53	7.2e-07	2.1e-06	0.1457	0.013	51	9.8e-07	2.5e-06	0.1079	0.1	31	9.4e-07	2.1e-06	0.0661	$(0.6325,10^{-7})$	36	7.2e-07	2.0e-06	0.0702
																	(0.2802,0.2802)	19	5.0e-07	6.6e-05	0.0254
																	(0.2,0.2)	18	7.7e-07	1.3e-04	0.0209
64×64	539	0.0030	83	9.2e-07	7.1e-06	0.8494	0.004	82	9.7e-07	4.9e-06	0.8444	0.08	45	9.6e-07	3.7e-06	0.3716	$(0.6347,10^{-7})$	52	9.9e-07	9.0e-07	0.4655
																	(0.2095,0.2095)	20	4.1e-07	6.0e-03	0.1465
																	(1,0.01)	19	6.0e-07	2.1e-04	0.1321
128×128	619	0.0013	131	9.0e-07	1.9e-05	7.0239	0.0016	130	9.9e-07	1.7e-05	6.8657	0.02	66	8.7e-07	1.1e-05	2.8650	$(0.3179,10^{-7})$	68	9.1e-07	1.1e-05	3.0114
																	(0.0768,0.0003)	47	8.1e-07	3.2e-05	1.9064

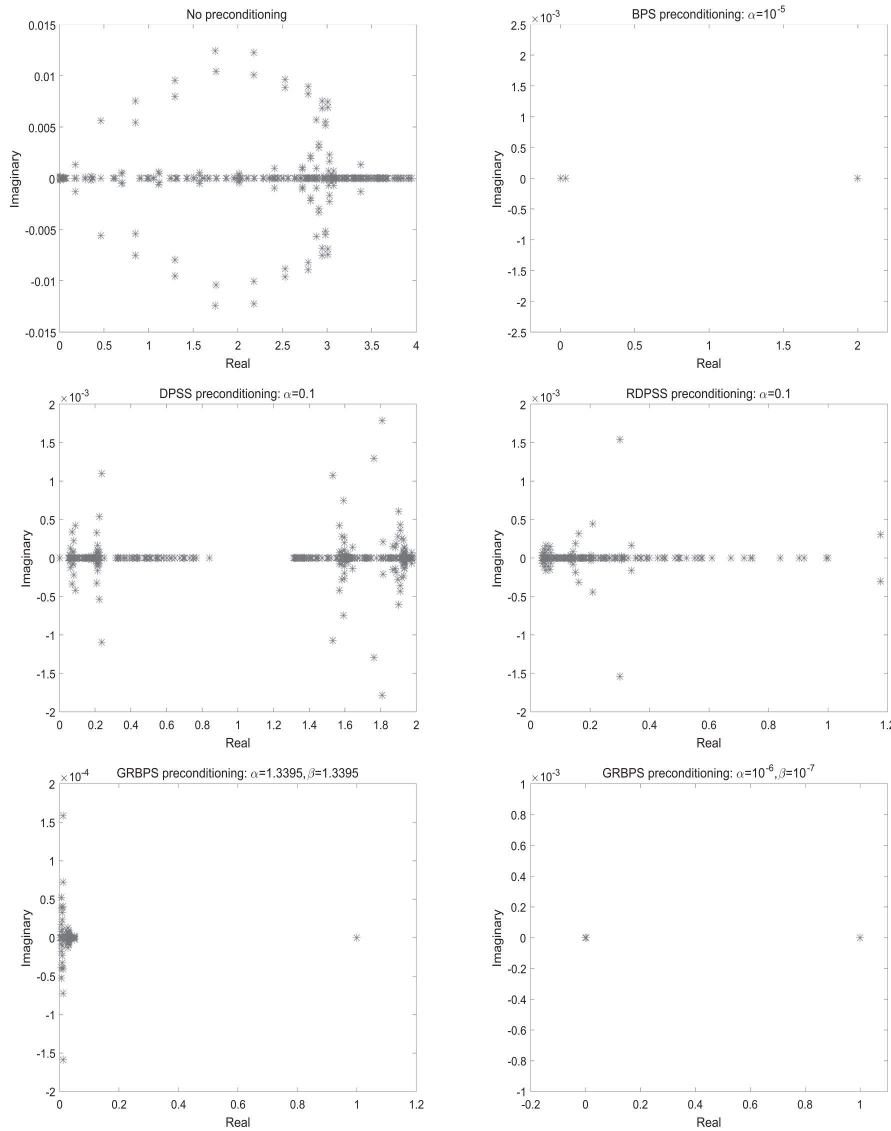

Fig. 1 Eigenvalue distribution of the generalized saddle point matrix with preconditioners for 16×16 grids ($\upsilon = 1$)

time, when α is located in the field of point 1.5, the iteration step of the GRBPS preconditioned GMRES method reach the minimum. On the contrary, the choice of $\alpha = \beta$ is better than that of $\alpha \neq \beta$ as $\nu = 0.1$, and when $\alpha = \beta > 0.6$, the rate of convergence of GRBPS preconditioned GMRES does not appear to be overly sensitive to the choice of α, in the sense that small relative changes in the size of α do not usually cause the number of iterations to change to drastically.

In Figure 4, we plot the values of residuals of the BPS, DPSS, RDPSS and GRBPS preconditioned GMRES methods with iteration steps for linearized Navier-stokes problem with $\upsilon = 1$(left) and $\upsilon = 0.1$(right) in 32×32 grids. In left figure, we choose estimation parameters $\alpha = 0.0402$, $\alpha = 0.05$, $\alpha = 0.5$ and $\alpha = 1.8$, $\beta = 10^{-7}$ for BPS, DPSS, RDPSS and GRBPS preconditioners, respectively; for above four preconditioners, we select parameters $\alpha = 0.0075$, $\alpha = 0.0.013$, $\alpha = 0.1$ and $\alpha = 0.2$, $\beta = 0.2$ in right figure, respectively. It can be observed from Figure 4 that the convergence curves of GRBPS and preconditioned GMRES method are more smoother and faster, on the contray, the convergence curves of the another two preconditioned GMRES methods appear inflection points and the phenomenon of residual errors increasing. Comparing the BPS, DPSS and RDPSS preconditioned GMRES method with the GRBPS preconditioned GMRES methods, we find that latter preconditioner is more effective and practical than another preconditioners.

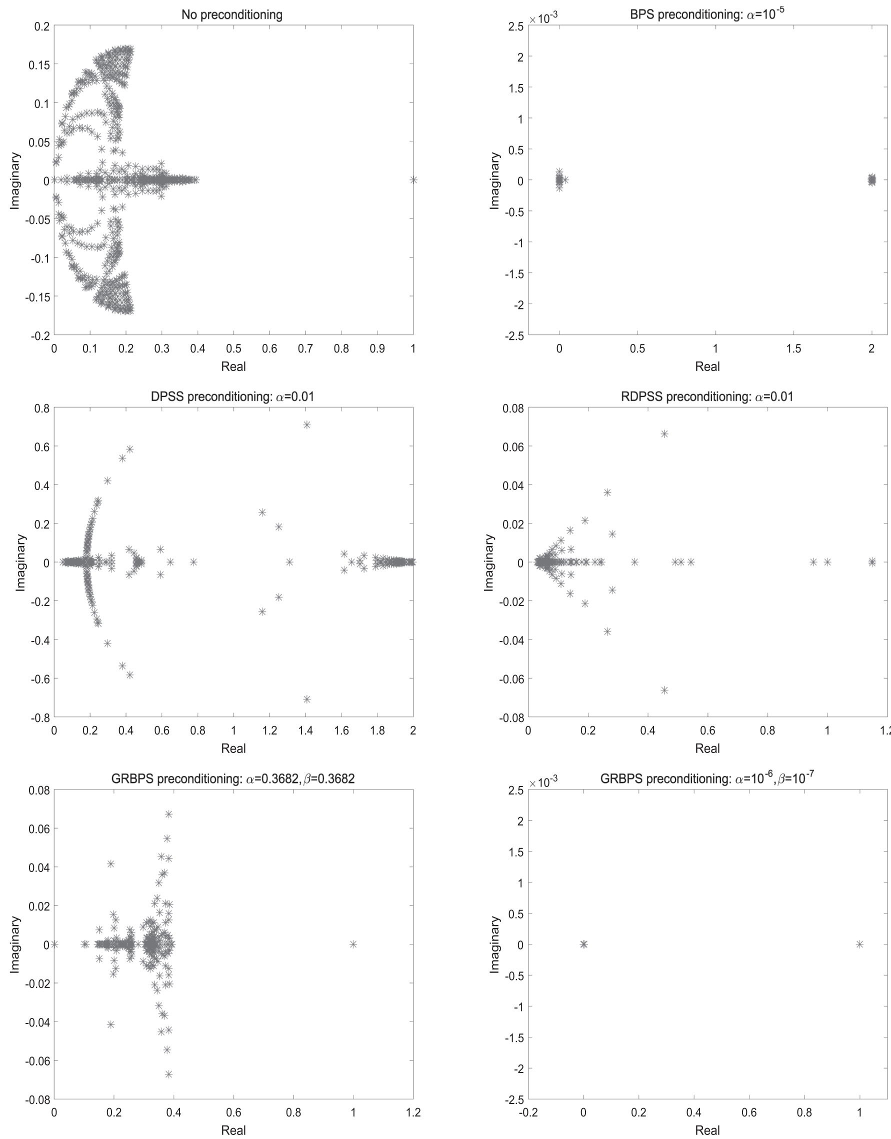

Fig. 2 Eigenvalue distribution of the generalized saddle point matrix with the preconditioners for 16×16 grids ($v = 0.1$)

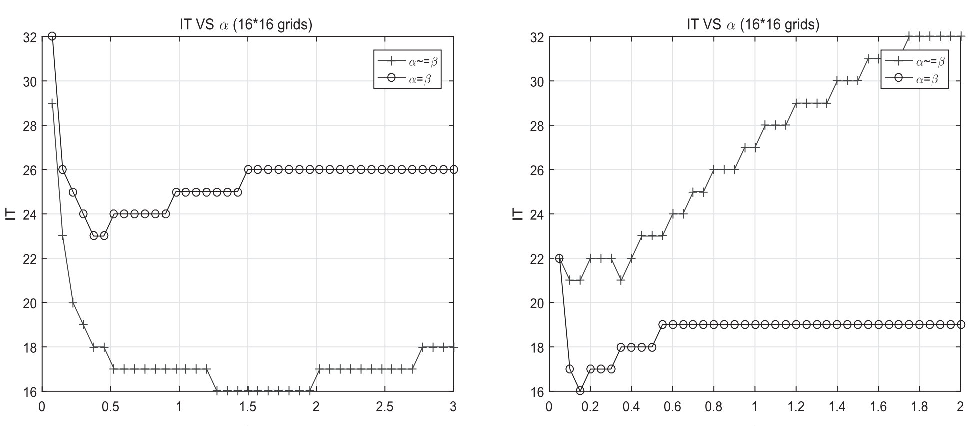

Fig. 3 Number of iteration steps vs. the iteration parameter α for Example 1: $v = 1(left)$ and $v = 0.1$ (right)

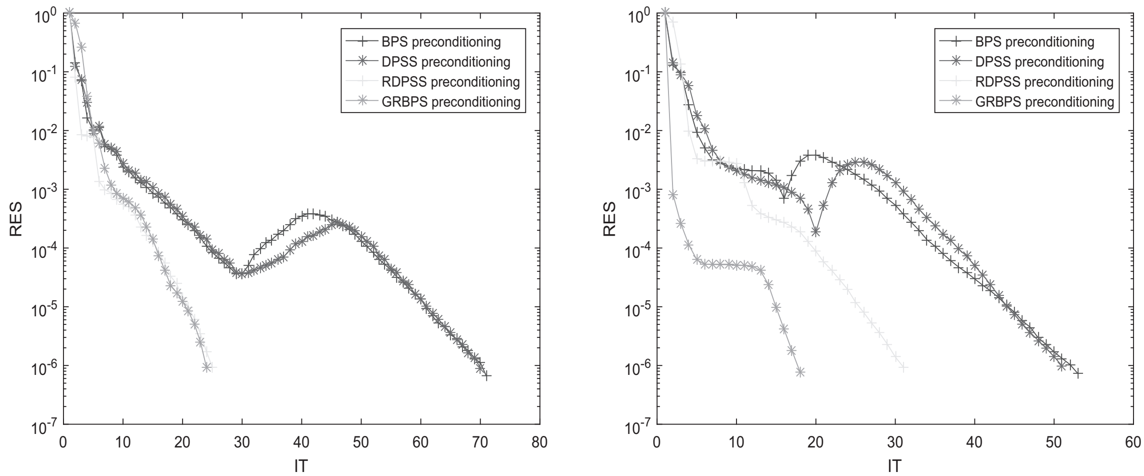

Fig. 4 The values of residuals RES with iteration steps IT for 32×32 grids $v = 1$(left) and $v = 0.1$(right)

From these numerical results, we see that the GRBPS preconditioners perform much better than BPS, DPSS and RDPSS preconditioners. This also shows that the new preconditioner could be more effectively to solve the generalized saddle point linear system (1.1).

6 Conclusions

In this paper, we establish a GRBPS preconditioner for generalized saddle point linear system (1.1). We mainly discuss the spectral properties of the GRBPS preconditioned matrix. Numerical experiments reveal that the proposed preconditioner has great superiority in suitable parameters compared with BPS, DPSS and RDPSS preconditioner under the exact algorithms.

Besides, it is easy to find from numerical experiment that the quasi-optimal parameters of GRBPS preconditioner given in Section 4 is relatively valid (i.e., the values of parameters α_{est} can not achieve the best numerical effect in the case of $v = 1$, while in the case of $v = 0.1$, the value of parameter α_{est}^1 is effective). In fact, it is difficult for the two parameters α and β to show their effective expressions for all problems. Next, we will focus on deriving the practical optimal parameters for GRBPS preconditioner and further discussing the preconditioning effect of the proposed preconditioners under inexact algorithms in our future work.

Acknowledgements We would like to express our sincere thanks to the anonymous reviewer for his/her valuable suggestions which greatly improved the presentation of this paper. 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. 11861059,12361082).

Declarations

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest. This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Z.-Z. Bai, G.H. Golub, M.K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems.* SIAM J. Matrix Anal. Appl. **24** (2003) 603–626.
2. Z.-Z. Bai, Z.-Q. Wang, *On parameterized inexact Uzawa methods for generalized saddle point problems,* Linear Algebra Appl. **428** (2008) 2900–2932.
3. Z.-Z. Bai, *Regularized HSS iteration methods for stabilized saddle-point problems,* IMA J. Numer. Anal. **39** (2019) 1888–1923.
4. M. Benzi, G.H. Golub, *A preconditioner for generalized saddle point problems.* SIAM J. Matrix Anal. Appl. **26** (2004) 20–41.
5. M. Benzi, G.H. Golub, J. Liesen, *Numercial solution of saddle point problems.* Acta Numer. **14** (2005) 1–137.

6. M. Benzi, M. Ng, Q. Niu, Z. Wang, *A relaxed dimensional fractorization preconditioner for the incompressible Navier-Stokes equations,* J. Comput. Phys. **230** (2011) 6185–6202.
7. T.-Q. Bui, M.N. Nguyen, C.Z. Zhang, D.A.K. Pham, *An efficient meshfree method for analysis of two-dimensional piezoelectric structures,* Smart Mater. Struct. **20** (2011) 065016.
8. Z.-H. Cao, *Fast Uzawa algorithm for generalized saddle point problems,* Appl. Numer. Math. **46** (2003) 157–171.
9. Z.-H. Cao, *Positive stable block triangular preconditioners for symmetric saddle point problems,* Appl. Numer. Math. **57** (2007) 899–910.
10. Y. Cao, L.-Q. Yao, M.-Q. Jiang, Q. Niu, *A relaxed HSS preconditioner for saddle point problems from meshfree discretization,* J. Comput. Math. **31** (2013) 398–421.
11. Y. Cao, S.-X. Miao, Y.-S. Cui, *A relaxed splitting preconditioner for generalized saddle point problems,* Comput. Appl. Math. **34** (2015) 865–879.
12. Y. Cao, J.-L. Dong, Y.-M. Wang, *A relaxed deteriorated PSS preconditioner for nonsymmetric saddle point problems from the steady Navier-Stokes equation.* J. Comput. Appl. Math. **273** (2015) 41–60.
13. Y. Cao, Z.-R. Ren, Q. Shi, *A simplified HSS preconditioner for generalized saddle point problems.* BIT Numer. Math. **56** (2016) 423–439.
14. Y. Cao, A. Wang, Y.-J. Chen, *A modified relaxed positive-semidefinite and skew-Hermitian splitting preconditioner for generalized saddle point problems,* East Asain J. Appl. Math. **7** (2017) 192–210.
15. Y. Cao, Z.-R. Ren, L.-Q. Yao, *Improved relaxed positive-definite and skew-Hermitian splitting preconditioners for saddle point problems,* J. Comput. Math. **37** (2019) 95–111.
16. Y. Cao, *A block positive-semidefinite splitting preconditioner for generalized saddle point linear systems,* J. Comput. Appl. Math. **374** (2020) 112787.
17. Y. Cao, Z.-Q. Shi, Q. Shi, *Regularized DPSS preconditioners for generalized saddle point linear systems,* Comput. Math. Appl. **80** (2020) 956-972.
18. Y. Cao, M. Neytcheva, *Cell-by-cell approximate Schur complement technique in preconditioning of meshfree discretized piezoelectric equations,* Numer. Linear Algebra Appl. **28** (2021) e2362.
19. F. Chen, *On choices of iteration parameter in HSS method,* Appl. Math. Comput. **271** (2015) 832–837.
20. H.-C. Elman, D.J. Silvester, A.J. Wathen, *Finite elements and fast Iterative Solvers with Applications in Incompressible Fluid Dynamics,* Oxford University Press, Oxford, UK, 2005.
21. H.-C. Elman, A. Ramage, D.J. Silvester, *IFISS: a matlab toolbox for modeling incompressible flow,* ACM Trans. Math. Software. **33** (2007) 14.
22. U. Guia, K.N. Ghia, C.T Shin, *Hign-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method,* J. Comput. Phys. **48** (1982) 387–411.
23. Y.-M. Huang, *A practical formula for computing optimal parameters in the HSS iteration methods,* J. Comput. Appl. Math. **255** (2014) 142–149.
24. R.-P. Huang, S.J. Qu, X.G. Yang, Z.M. Liu, *Multi-stage distributionally robust optimization with risk aversion,* J. Ind. Manag. Optim. **13:5** (2017).
25. L.-D. Liao, G.-F. Zhang, *A generalized variant of simplified HSS preconditioner for generalized saddle point problems,* Appl. Math. Comput. **346** (2019) 790–799.
26. J. Li, S.-X. Miao, *A generalized SHSS preconditioner for generalized saddle point problem.* Comput. Appl. Math. **39:274** (2020).
27. M. Masoudi, D. K.Salkuyeh, *An extension of the positive-definite and skew-Hermitian splitting method for preconditioning of generalized saddle point problems.* Comput. Math. Appl. **79** (2020) 2304–2321.
28. J.-Y. Pan, M.K. Ng, Z.Z. Bai, *New preconditioners for saddle point problems,* Appl. Math. Comput. **172** (2006) 762–771.
29. Z.-R. Ren, Y. Cao, *An alternating positive-semidefinite splitting preconditioner for saddle point problems from time-harmonic eddy current models,* IMA J. Numer. Anal. **36** (2016) 922–946.
30. Y. Saad, *Iterative Methods for Sparse Linear Systems.* 2nd edn. SIAM. Philadelphia (2003).
31. S.-Q. Shen, *A note on PSS preconditioners for generalized saddle point problems,* Appl. Math. Comput. **237** (2014) 723–729.
32. Q.-Q. Shen, Y. Cao, L. Wang, *Two improvements of the deteriorated PSS preconditioner for generalized saddle point problems,* Numer. Algor. **75** (2017) 33–54.
33. Q.-Q. Shen, Q. Shi, *Inexact modified positive-definite and skew-Hermitian splitting preconditioners for generalized saddle point problems,* Adv. Mech. Eng. **10** (2018) 1–10.
34. A.-L. Yang, *Scaled norm minimization method for computing the parameters of the HSS and the two-parameter HSS preconditioners,* Numer. Linear Algebra Appl. **25** (2018) e2169.
35. Y.-Y. Zhou, G.-F. Zhang, *A generalization of parameterized inexact Uzawa method for generalized saddle point problems,* Appl. Math. Comput. **215** (2009) 599–607.