

A MODIFIED ALTERNATING POSITIVE SEMIDEFINITE SPLITTING PRECONDITIONER FOR BLOCK THREE-BY-THREE SADDLE POINT PROBLEMS*

FANG CHEN[†] AND BI-CONG REN[†]

Abstract. We propose a modified alternating positive semidefinite splitting (MAPSS) preconditioner for solving block three-by-three saddle point problems that arise in linear programming and the finite element discretization of Maxwell equations. Spectral properties of the MAPSS-preconditioned matrix are discussed and analyzed in detail. As the efficiency of the MAPSS preconditioner depends on its parameters, we derive fast and effective formulas to compute the quasi-optimal values of these parameters. Numerical examples show that the MAPSS preconditioner performs better than the APSS preconditioner.

Key words. saddle point problem, convergence analysis, Krylov subspace iteration methods, preconditioned matrix

AMS subject classifications. 65F10, 65F08

1. Introduction. We consider the following block three-by-three saddle point problem:

$$(1.1) \quad \mathcal{A}\mathbf{u} \equiv \begin{bmatrix} A & B^T & 0 \\ -B & 0 & -C^T \\ 0 & C & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f \\ -g \\ h \end{bmatrix} \equiv \mathbf{b},$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{l \times m}$ are full row-rank matrices, $x, f \in \mathbb{R}^n$, $y, g \in \mathbb{R}^m$, and $z, h \in \mathbb{R}^l$. Here, B^T and C^T denote the transposes of the matrices B and C , respectively. It is easy to verify that the coefficient matrix \mathcal{A} is nonsingular so that the linear system (1.1) has a unique solution [6, 14, 16].

The linear system (1.1) arises in many areas of scientific computing and engineering applications, such as in the Karush-Kuhn-Tucker conditions of quadratic programming problems, in finite element methods for solving time-dependent Maxwell equations with discontinuous coefficients, in least-squares problems [1, 2, 5, 8, 14, 36, 49], and so on.

Although the linear system (1.1) has a block three-by-three structure, it can be regarded as a block two-by-two matrix as in the following formulas:

$$\mathcal{A} = \left[\begin{array}{c|cc} A & B^T & 0 \\ \hline -B & 0 & -C^T \\ 0 & C & 0 \end{array} \right] \quad \text{and} \quad \mathcal{A} = \left[\begin{array}{cc|c} A & B^T & 0 \\ \hline -B & 0 & -C^T \\ 0 & C & 0 \end{array} \right].$$

Obviously, these matrices have block structures similar to the standard and generalized saddle point problems [8, 14, 22]. As it is well-known, there are many effective iterative methods and algebraic preconditioners for solving these saddle point problems, such as Uzawa-type methods [17, 31, 38, 50], *successive overrelaxation-type* (SOR-type) methods [15, 33, 34, 35, 44, 48], *Hermitian and skew-Hermitian splitting-type* (HSS-type) methods [9, 10, 12, 13, 19, 21, 23, 24, 25, 28, 30, 32, 40, 41, 43, 47, 51, 52], *shift-splitting-type* (SS-type) methods [18, 27, 53], and so on [2, 4, 5, 6, 11]. However, because of the complicated structure of the coefficient matrix of the linear system (1.1), these iterative methods cannot be straightforwardly applied

*Received February 16, 2022. Accepted July 8, 2022. Published online on November 23, 2022. Recommended by Zhong-Zhi Bai. Supported by R&D Program of Beijing Municipal Education Commission (No. KM202011232019), P.R. China.

[†]School of Applied Science, Beijing Information Science and Technology University, Beijing 100192, P.R. China (chenfreesky@126.com, study_rbc@163.com)

to solve it, and they often show poor numerical performance. Therefore, there is a great need to find specific iterative methods to solve the three-by-three saddle point problem (1.1).

Fortunately, some iteration methods and preconditioning techniques have been proposed for the linear system (1.1) in recent years. Based on the *shift splitting* (SS) technique, first proposed and discussed in [18], Cao [26] proposed the SS and the *relaxed SS* (RSS) preconditioner. Later, Wang and Zhang [45] proposed the *generalized SS* (GSS) preconditioner, and Xie and Li [46] presented the block diagonal-type preconditioners. Furthermore, Bai, Golub, Lu, and Yin [11] first proposed the *positive-definite and skew-Hermitian splitting* (PSS) iteration method for solving non-Hermitian positive definite linear systems, and by further extending this idea, Liang and Zhang [39] presented the *alternating positive semidefinite splitting* (APSS) preconditioner for solving double saddle point problems. The APSS preconditioning can be considered as a special case of the BASI preconditioning in [5]. Also, Salkuyeh, Aslani, and Liang [42] applied the APSS preconditioner in conjunction with Krylov subspace iteration methods to solve the linear system (1.1).

When the APSS preconditioner is used to accelerate Krylov subspace iteration methods, it is often difficult to choose the parameter α in the APSS preconditioner for different numerical examples in the extreme case when $\alpha \rightarrow 0_+$ or $\alpha \rightarrow +\infty$. In order to overcome this shortcoming, in this paper we give a useful formula for choosing such a parameter. Moreover, to further improve the computational efficiency of the APSS preconditioner, we propose a *modified APSS* (MAPSS) preconditioner for the block three-by-three saddle point problem (1.1). By removing the term αI and introducing another positive parameter β , we then obtain the MAPSS preconditioner. We analyze and discuss spectral properties of the MAPSS-preconditioned matrix and give practical formulas for choosing the parameters involved in the APSS- and MAPSS-preconditioned matrices. In addition, numerical results are reported to show the effectiveness of the MAPSS preconditioner.

The paper is organized as follows. In Section 2, we introduce the MAPSS preconditioner and give its algorithmic implementation for accelerating Krylov subspace iteration methods. In Section 3, we analyze spectral properties of the preconditioned matrix with respect to the MAPSS preconditioner. Efficient practical formulas for computing the optimal values of the parameters involved in the MAPSS and APSS preconditioners are derived in Section 4. In Section 5, several examples are used to verify that the numerical performance of the MAPSS preconditioner outperforms the APSS preconditioner. Finally, in Section 6, we give concluding remarks.

2. The MAPSS preconditioner. First, we introduce the APSS preconditioner [42]. Based on the idea of the HSS iteration method [12], the matrix \mathcal{A} is split into

$$\mathcal{A} = (\alpha I + \mathcal{A}_1) - (\alpha I - \mathcal{A}_2) = (\alpha I + \mathcal{A}_2) - (\alpha I - \mathcal{A}_1),$$

where

$$\mathcal{A}_1 = \begin{bmatrix} A & B^T & 0 \\ -B & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -C^T \\ 0 & C & 0 \end{bmatrix},$$

and $\alpha > 0$ is a parameter and I is the identity matrix. On basis of these splittings and the HSS iteration spirit, the authors in [42] proposed the following APSS preconditioner:

$$\tilde{\mathcal{P}}_{\text{APSS}} = \frac{1}{2\alpha} (\alpha I + \mathcal{A}_1)(\alpha I + \mathcal{A}_2) = \frac{1}{2} \begin{bmatrix} \alpha I + A & B^T & -\frac{1}{\alpha} B^T C^T \\ -B & \alpha I & -C^T \\ 0 & C & \alpha I \end{bmatrix}.$$

Since the factor $\frac{1}{2}$ in the above matrix has no effect on the preconditioned linear system, we can simply drop it and write the preconditioning matrix $\mathcal{P}_{\text{APSS}} = \frac{1}{\alpha}(\alpha I + \mathcal{A}_1)(\alpha I + \mathcal{A}_2)$ in the decomposed form

$$\mathcal{P}_{\text{APSS}} = \frac{1}{\alpha} \begin{bmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I & -C^T \\ 0 & C & \alpha I \end{bmatrix},$$

in which the first matrix admits the LU decomposition

$$\begin{bmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix} = \begin{bmatrix} \alpha I + A + \frac{1}{\alpha} B^T B & B^T & 0 \\ 0 & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ -\frac{1}{\alpha} B & I & 0 \\ 0 & 0 & I \end{bmatrix},$$

while the second matrix admits the LU decomposition

$$\begin{bmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I & -C^T \\ 0 & C & \alpha I \end{bmatrix} = \begin{bmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I + \frac{1}{\alpha} C^T C & -C^T \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{1}{\alpha} C & I \end{bmatrix}.$$

It follows that

$$(2.1) \quad \mathcal{P}_{\text{APSS}} = \frac{1}{\alpha} \begin{bmatrix} \alpha I + A + \frac{1}{\alpha} B^T B & B^T & 0 \\ 0 & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ -\frac{1}{\alpha} B & I & 0 \\ 0 & 0 & I \end{bmatrix} \\ \times \begin{bmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I + \frac{1}{\alpha} C^T C & -C^T \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{1}{\alpha} C & I \end{bmatrix}.$$

In this way, we obtain an efficient method for solving a linear system of the form $\mathcal{P}_{\text{APSS}} w = r$ with respect to the unknown w and the right-hand side r . Indeed, when we apply the APSS preconditioner to accelerate Krylov methods, such as the *generalized minimum residual* (GMRES) method, we need to solve a linear system of the form $\mathcal{P}_{\text{APSS}} w = r$, where $w = [w_1^T; w_2^T; w_3^T]^T$ and $r = [r_1^T; r_2^T; r_3^T]^T$. Based on the matrix decomposition in (2.1), we obtain an algorithmic implementation of the APSS preconditioner as follows.

ALGORITHM 1 (APSS Preconditioning).

Step 1: Solve $\left(\alpha I + A + \frac{1}{\alpha} B^T B\right) w_1 = r_1 - \frac{1}{\alpha} B^T r_2$ to obtain w_1 .

Step 2: Solve $\left(\alpha I + \frac{1}{\alpha} C^T C\right) w_2 = r_2 + \frac{1}{\alpha} C^T r_3 + B w_1$ to obtain w_2 .

Step 3: Compute $w_3 = \frac{1}{\alpha} (r_3 - C w_2)$.

In Algorithm 1, the matrices $\alpha I + A + \frac{1}{\alpha} B^T B$ and $\alpha I + \frac{1}{\alpha} C^T C$ are symmetric positive definite, and the factors α and $\frac{1}{\alpha}$ appear in all coefficient matrices of Algorithm 1. This makes the choice of the parameter α significantly difficult. Therefore, we turn to consider removing the term αI in Step 1 and introducing another positive parameter β in Step 2. In this way, we obtain the so-called MAPSS preconditioner.

Note that the factor $\frac{1}{2}$ has no substantial effect on the preconditioner. We construct the MAPSS preconditioner as follows:

$$\begin{aligned}
 \mathcal{P}_{\text{MAPSS}} &= \frac{1}{\alpha} \begin{bmatrix} A + \frac{1}{\alpha} B^T B & B^T & 0 \\ 0 & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ -\frac{1}{\alpha} B & I & 0 \\ 0 & 0 & I \end{bmatrix} \\
 &\quad \times \begin{bmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I + \frac{1}{\beta} C^T C & -C^T \\ 0 & 0 & \beta I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & \frac{1}{\beta} C & I \end{bmatrix} \\
 &= \frac{1}{\alpha} \begin{bmatrix} A & B^T & 0 \\ -B & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & 0 & 0 \\ 0 & \alpha I & -C^T \\ 0 & C & \beta I \end{bmatrix} \\
 &= \begin{bmatrix} A & B^T & -\frac{1}{\alpha} B^T C^T \\ -B & \alpha I & -C^T \\ 0 & C & \beta I \end{bmatrix},
 \end{aligned}$$

where α and β are given positive parameters.

Based on the above matrix decompositions, we solve a linear system of the form $\mathcal{P}_{\text{MAPSS}} w = r$ by the following algorithm.

ALGORITHM 2 (MAPSS Preconditioning).

Step 1: Solve $\left(A + \frac{1}{\alpha} B^T B \right) w_1 = r_1 - \frac{1}{\alpha} B^T r_2$ to obtain w_1 .

Step 2: Solve $\left(\alpha I + \frac{1}{\beta} C^T C \right) w_2 = B w_1 + r_2 + \frac{1}{\beta} C^T r_3$ to obtain w_2 .

Step 3: Compute $w_3 = \frac{1}{\beta} (r_3 - C w_2)$.

From a comparison of Algorithm 1 and Algorithm 2 we deduce the following observations. In Algorithm 1, two linear subsystems with the coefficient matrices $\alpha I + A + \frac{1}{\alpha} B^T B$ and $\alpha I + \frac{1}{\alpha} C^T C$ need to be solved at each iteration step, while in Algorithm 2 the two linear subsystems to be solved are characterized by the coefficient matrices $A + \frac{1}{\alpha} B^T B$ and $\alpha I + \frac{1}{\beta} C^T C$. The factors α and $\frac{1}{\alpha}$ appear in the same coefficient matrices in Steps 1 and 2 of Algorithm 1, while Algorithm 2 avoids this unfavorable case. Consequently, it is much easier to estimate the parameters α and β in Algorithm 2 than just choosing the parameter α in Algorithm 1.

3. Analysis of the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1} \mathcal{A}$. In this section, we discuss and analyze spectral properties of the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1} \mathcal{A}$. We have the following result.

THEOREM 3.1. *Let $A \in \mathbb{R}^{n \times n}$ be symmetric positive definite and $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{l \times m}$ be of full row-rank. Then, the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1} \mathcal{A}$ has an eigenvalue 1 with algebraic multiplicity at least n , and the remaining eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_{m+l}$ satisfy the condition*

$$\sum_{i=1}^{m+l} \lambda_i = m + \sum_{i=1}^m \eta_i,$$

where $\eta_1, \eta_2, \dots, \eta_m$ are the eigenvalues of the matrix

$$T = \hat{S}M_2^{-1} - \frac{1}{\alpha\beta}\hat{S}C^T C M_2^{-1},$$

with

$$M_1 = A + \frac{1}{\alpha}B^T B, \quad M_2 = \alpha I + \frac{1}{\beta}C^T C, \quad S_1 = B M_1^{-1} B^T, \quad \text{and} \quad \hat{S} = S_1 - \alpha I.$$

Proof. The difference between the preconditioner $\mathcal{P}_{\text{MAPSS}}$ and the coefficient matrix \mathcal{A} , denoted by $\mathcal{R}_{\text{MAPSS}}$, is given by

$$(3.1) \quad \mathcal{R}_{\text{MAPSS}} = \mathcal{P}_{\text{MAPSS}} - \mathcal{A} = \begin{bmatrix} 0 & 0 & -\frac{1}{\alpha}B^T C^T \\ 0 & \alpha I & 0 \\ 0 & 0 & \beta I \end{bmatrix}.$$

By using the block decomposition of the matrix $\mathcal{P}_{\text{MAPSS}}$, we get

$$\begin{aligned} \mathcal{P}_{\text{MAPSS}}^{-1} &= \alpha \begin{bmatrix} \frac{1}{\alpha}I & 0 & 0 \\ 0 & M_2^{-1} & \frac{1}{\beta}M_2^{-1}C^T \\ 0 & -\frac{1}{\beta}C M_2^{-1} & \frac{1}{\beta}I - \frac{1}{\beta^2}C M_2^{-1}C^T \end{bmatrix} \\ &\quad \times \begin{bmatrix} M_1^{-1} & -\frac{1}{\alpha}M_1^{-1}B^T & 0 \\ \frac{1}{\alpha}B M_1^{-1} & -\frac{1}{\alpha^2}S_1 + \frac{1}{\alpha}I & 0 \\ 0 & 0 & \frac{1}{\alpha}I \end{bmatrix} \\ &= \begin{bmatrix} M_1^{-1} & -\frac{1}{\alpha}M_1^{-1}B^T & 0 \\ M_2^{-1}B M_1^{-1} & -\frac{1}{\alpha}M_2^{-1}\hat{S} & \frac{1}{\beta}M_2^{-1}C^T \\ -\frac{1}{\beta}C M_2^{-1}B M_1^{-1} & \frac{1}{\alpha\beta}C M_2^{-1}\hat{S} & \frac{1}{\beta}I - \frac{1}{\beta^2}C M_2^{-1}C^T \end{bmatrix}. \end{aligned}$$

Hence, straightforward computations yield

$$(3.2) \quad \begin{aligned} \mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A} &= I - \mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{R}_{\text{MAPSS}} \\ &= \begin{bmatrix} I & M_1^{-1}B^T & \frac{1}{\alpha}M_1^{-1}B^T C^T \\ 0 & I + M_2^{-1}\hat{S} & \frac{1}{\alpha}M_2^{-1}\hat{S}C^T \\ 0 & -\frac{1}{\beta}C M_2^{-1}\hat{S} & -\frac{1}{\alpha\beta}C M_2^{-1}\hat{S}C^T \end{bmatrix}. \end{aligned}$$

It is then clear that the matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ has an eigenvalue 1 with algebraic multiplicity at least n , while the remaining eigenvalues are the same as those of the matrix

$$(3.3) \quad Z = \begin{bmatrix} I + M_2^{-1}\hat{S} & \frac{1}{\alpha}M_2^{-1}\hat{S}C^T \\ -\frac{1}{\beta}C M_2^{-1}\hat{S} & -\frac{1}{\alpha\beta}C M_2^{-1}\hat{S}C^T \end{bmatrix}.$$

Denote by

$$Z = \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix} + \tilde{T},$$

where

$$\tilde{T} = \begin{bmatrix} M_2^{-1}\hat{S} & \frac{1}{\alpha}M_2^{-1}\hat{S}C^T \\ -\frac{1}{\beta}CM_2^{-1}\hat{S} & -\frac{1}{\alpha\beta}CM_2^{-1}\hat{S}C^T \end{bmatrix} \equiv XY,$$

with

$$X = \begin{bmatrix} M_2^{-1} \\ -\frac{1}{\beta}CM_2^{-1} \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} \hat{S} & \frac{1}{\alpha}\hat{S}C^T \end{bmatrix}.$$

Moreover, let $T = YX$. Then, \tilde{T} and T have the same nonzero eigenvalues [14, Theorem 1.19], and it holds that

$$\text{tr}(Z) = \text{tr} \left\{ \begin{bmatrix} I_m & 0 \\ 0 & 0 \end{bmatrix} + \tilde{T} \right\} = m + \text{tr}(\tilde{T}) = m + \text{tr}(T).$$

Here $\text{tr}(Z)$ stands for the trace of the matrix Z . Now, the final claim of the theorem is immediately obtained. \square

REMARK 3.2. Under the conditions of Theorem 3.1, we have

$$\begin{aligned} \sum_{i=1}^m \eta_i &= \text{tr}(T) \\ &= \text{tr}(\hat{S}M_2^{-1}) - \frac{1}{\alpha\beta}\text{tr}(\hat{S}C^T C M_2^{-1}) = \text{tr}(M_2^{-1}\hat{S}) - \frac{1}{\alpha\beta}\text{tr}(M_2^{-1}\hat{S}C^T C) \\ &= \text{tr} \left[M_2^{-1}\hat{S} \left(I - \frac{1}{\alpha\beta}C^T C \right) \right] = \text{tr} \left\{ M_2^{-1}\hat{S} \left[I - \frac{1}{\alpha}(\alpha I + \frac{1}{\beta}C^T C) + I \right] \right\} \\ &= \text{tr} \left[M_2^{-1}\hat{S} \left(2I - \frac{1}{\alpha}M_2 \right) \right] = 2\text{tr}(M_2^{-1}\hat{S}) - \frac{1}{\alpha}\text{tr}(\hat{S}). \end{aligned}$$

As it is well-known, the convergence of Krylov subspace iteration methods depends not only on the distributions of the eigenvalues of the preconditioned matrix, but also on the number of the corresponding linearly independent eigenvectors [3, 7, 14]. In the following theorem, we investigate the properties of the linearly independent eigenvectors of the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$.

THEOREM 3.3. *If the conditions of Theorem 3.1 hold, then the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ has the following $n + j$ linearly independent eigenvectors:*

1. n eigenvectors of the form $[u_\ell^T; 0^T; 0^T]^T$ ($\ell = 1, 2, \dots, n$) corresponding to the eigenvalue 1, where $u_\ell \in \mathbb{R}^n$ are arbitrary linearly independent vectors;
2. j eigenvectors of the form $[\hat{u}_\ell^T; \hat{v}_\ell^T; \hat{w}_\ell^T]^T$ ($\ell = 1, 2, \dots, j$) corresponding to the eigenvalues $\lambda \neq 1$, where $\hat{v} \neq 0$ satisfies

$$\left[I + M_2^{-1}\hat{S} \left(\frac{1}{1-\lambda}I + \frac{1}{\alpha\beta\lambda}C^T C \right) \right] \hat{v} = 0,$$

and \hat{w} and \hat{u} are given by

$$\hat{w} = \frac{1-\lambda}{\lambda\beta}C\hat{v}, \quad \hat{u} = M^{-1}B^T \left(\frac{1}{\lambda-1}I - \frac{1}{\alpha\beta\lambda}C^T C \right) \hat{v}.$$

Proof. Let λ be an eigenvalue of the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ and $[u^T; v^T; w^T]^T$ be the corresponding eigenvector. By (3.2) we obtain

$$\begin{bmatrix} I & M_1^{-1}B^T & \frac{1}{\alpha}M_1^{-1}B^TC^T \\ 0 & I + M_2^{-1}\hat{S} & \frac{1}{\alpha}M_2^{-1}\hat{S}C^T \\ 0 & -\frac{1}{\beta}CM_2^{-1}\hat{S} & -\frac{1}{\alpha\beta}CM_2^{-1}\hat{S}C^T \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ w \end{bmatrix}.$$

After straightforward calculations, we can rewrite the above equations as

$$\begin{aligned} (\lambda - 1)u - M_1^{-1}B^T \left(v + \frac{1}{\alpha}C^T w \right) &= 0, \\ (3.4) \quad \left[(1 - \lambda)I + M_2^{-1}\hat{S} \right] v + \frac{1}{\alpha}M_2^{-1}\hat{S}C^T w &= 0, \\ \frac{1}{\beta}CM_2^{-1}\hat{S}v + \left(\lambda I + \frac{1}{\alpha\beta}CM_2^{-1}\hat{S}C^T \right) w &= 0. \end{aligned}$$

If $\lambda = 1$, then the equations in (3.4) are reduced to

$$\begin{aligned} (3.5) \quad M_1^{-1}B^T \left(v + \frac{1}{\alpha}C^T w \right) &= 0, \\ M_2^{-1}\hat{S} \left(v + \frac{1}{\alpha}C^T w \right) &= 0, \\ \frac{1}{\beta}CM_2^{-1}\hat{S}v + \left(I + \frac{1}{\alpha\beta}CM_2^{-1}\hat{S}C^T \right) w &= 0. \end{aligned}$$

Multiplying the second equation in (3.5) from the left by $-\frac{1}{\beta}C$, we have

$$-\frac{1}{\beta}CM_2^{-1}\hat{S} \left(v + \frac{1}{\alpha}C^T w \right) = 0,$$

which together with the third equation in (3.5) gives $w = 0$. Substituting $w = 0$ into the first equation in (3.5), we obtain $M_1^{-1}B^T v = 0$. By the positive definiteness of M_1^{-1} and the full row-rank property of B we have $v = 0$. Therefore, there are n linearly independent eigenvectors $[u_\ell^T; 0^T; 0^T]^T$ ($\ell = 1, 2, \dots, n$) corresponding to the eigenvalue 1, where u_ℓ ($\ell = 1, 2, \dots, n$) are arbitrary linearly independent vectors.

If $\lambda \neq 1$, then, combining the second and third equations in (3.4), we obtain

$$w = \frac{(1 - \lambda)}{\lambda\beta} C v.$$

Substituting the above equation into the first and the second equations in (3.4), we have

$$u = M^{-1}B^T \left(\frac{1}{\lambda - 1} I - \frac{1}{\alpha\beta\lambda} C^T C \right) v$$

and

$$(3.6) \quad \left[I + M_2^{-1}\hat{S} \left(\frac{1}{1 - \lambda} I + \frac{1}{\alpha\beta\lambda} C^T C \right) \right] v = 0.$$

If $v = 0$, then $w = 0$ and $u = 0$, which contradicts the fact that $[u^T; v^T; w^T]^T$ is an eigenvector. So, it holds that $v \neq 0$. Hence, when there exists an $v \neq 0$ that satisfies (3.6), there will be j linearly independent eigenvectors $[\hat{u}_\ell^T; \hat{v}_\ell^T; \hat{w}_\ell^T]^T$ ($\ell = 1, 2, \dots, j$) that correspond to the eigenvalue $\lambda \neq 1$.

Finally, we show that the above mentioned $n + j$ eigenvectors are linearly independent. That is to say, we need to prove that

$$(3.7) \quad \begin{bmatrix} u_1 & \cdots & u_n \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} + \begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_j \\ \hat{v}_1 & \cdots & \hat{v}_j \\ \hat{w}_1 & \cdots & \hat{w}_j \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \vdots \\ \hat{\theta}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

holds only when the vectors $\theta = [\theta_1, \dots, \theta_n]^T$ and $\hat{\theta} = [\hat{\theta}_1, \dots, \hat{\theta}_j]^T$ are zero. Obviously, the first matrix in (3.7) consists of the eigenvectors corresponding to the eigenvalue 1 for case 1 in Theorem 3.3, and the second matrix is formed from the eigenvectors corresponding to the eigenvalues $\lambda_\ell \neq 1$ ($\ell = 1, 2, \dots, j$). By left-multiplying $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ on both sides of equality (3.7), we obtain

$$\begin{bmatrix} u_1 & \cdots & u_n \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} + \begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_j \\ \hat{v}_1 & \cdots & \hat{v}_j \\ \hat{w}_1 & \cdots & \hat{w}_j \end{bmatrix} \begin{bmatrix} \lambda_1 \hat{\theta}_1 \\ \vdots \\ \lambda_j \hat{\theta}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Subtracting the above equations from (3.7), we see that

$$\begin{bmatrix} \hat{u}_1 & \cdots & \hat{u}_j \\ \hat{v}_1 & \cdots & \hat{v}_j \\ \hat{w}_1 & \cdots & \hat{w}_j \end{bmatrix} \begin{bmatrix} (\lambda_1 - 1)\hat{\theta}_1 \\ \vdots \\ (\lambda_j - 1)\hat{\theta}_j \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Since $[\hat{u}_\ell^T; \hat{v}_\ell^T; \hat{w}_\ell^T]^T$ ($\ell = 1, 2, \dots, j$) are j linearly independent vectors and they correspond to the eigenvalues $\lambda_\ell \neq 1$ ($\ell = 1, 2, \dots, j$), we obtain $\hat{\theta} = 0$. Substituting $\hat{\theta} = 0$ into (3.7), we obtain $\theta = 0$. Hence, the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ has $n + j$ linearly independent eigenvectors. \square

An upper bound for the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ is given in the following theorem.

THEOREM 3.4. *Let the conditions of Theorem 3.1 be satisfied. Then, the dimension of the Krylov subspace $\mathcal{K}(\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}, \mathbf{b})$ is at most $m + l + 1$.*

Proof. The characteristic polynomial of the matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ defined in (3.2) is given by

$$\psi(\lambda) = \det(\lambda I - \mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}) = (\lambda - 1)^n \prod_{\ell=1}^{m+l} (\lambda - \lambda_\ell),$$

where λ_ℓ ($\ell = 1, 2, \dots, m + l$) are the eigenvalues of the matrix Z in (3.3). Using the Cayley-Hamilton Theorem [14, Theorem 1.21], we have $\prod_{\ell=1}^{m+l} (Z - \lambda_\ell I) = 0$.

Let us define the polynomial

$$\phi(\lambda) = (\lambda - 1) \prod_{\ell=1}^{m+l} (\lambda - \lambda_\ell).$$

Then, we obtain

$$\phi(\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}) = (\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A} - I) \prod_{\ell=1}^{m+l} (\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A} - \lambda_{\ell}I) = 0.$$

Hence, the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}$ is at most $m+l+1$. Consequently, the dimension of the corresponding Krylov subspace $\mathcal{K}(\mathcal{P}_{\text{MAPSS}}^{-1}\mathcal{A}, \mathbf{b})$ is at most $m+l+1$. \square

4. The strategy of parameter selection. It is worth noting that the efficiency of the MAPSS preconditioner largely depends on the choice of the parameters α and β . In this section, we discuss and analyze how to choose the parameters α and β involved in the MAPSS and APSS preconditioners.

The choice of parameters is very complicated for most preconditioners, such as, for instance, SS-type preconditioners [26, 27, 45, 53]. Hence, it is often necessary to perform many experiments to experimentally choose optimal parameters. Fortunately, efficient formulas for obtaining nearly optimal parameters can be obtained by adopting certain approximation strategies; see [4, 8, 9, 20, 29, 37] for more details. Motivated by the idea of Huang [37], we give a practical way for choosing the parameters in the MAPSS preconditioner.

We first define a function $f(\alpha, \beta) = \|\mathcal{R}_{\text{MAPSS}}\|_F^2$ depending on α and β and then select these parameters by minimizing $f(\alpha, \beta)$, where $\mathcal{R}_{\text{MAPSS}}$ is given by (3.1) and $\|\cdot\|_F$ denotes the Frobenius norm. From straightforward computations we obtain

$$f(\alpha, \beta) = \|\mathcal{R}_{\text{MAPSS}}\|_F^2 = \text{tr}(\mathcal{R}_{\text{MAPSS}}^T \mathcal{R}_{\text{MAPSS}}) = \alpha^2 m + \frac{1}{\alpha^2} \text{tr}(CBB^T C^T) + \beta^2 l.$$

For convenience, we let β be a constant and only analyze the parameter α . By minimizing the function $f(\alpha, \beta)$ with respect to α , we obtain the quasi-optimal parameter α in the MAPSS preconditioner as follows:

$$(4.1) \quad \alpha_{\text{MAPSS}} = \sqrt[4]{\frac{\text{tr}(CBB^T C^T)}{m}} = \sqrt[4]{\frac{\text{tr}(BB^T C^T C)}{m}}.$$

We use a similar idea to estimate the parameter α in the APSS preconditioner. Since

$$\mathcal{R}_{\text{APSS}} = \mathcal{P}_{\text{APSS}} - \mathcal{A} = \begin{bmatrix} \alpha I & 0 & -\frac{1}{\alpha} B^T C^T \\ 0 & \alpha I & 0 \\ 0 & 0 & \alpha I \end{bmatrix},$$

we define

$$g(\alpha) = \|\mathcal{R}_{\text{APSS}}\|_F^2 = \text{tr}(\mathcal{R}_{\text{APSS}}^T \mathcal{R}_{\text{APSS}}).$$

Then, we see that

$$g(\alpha) = \alpha^2(m+n+l) + \frac{1}{\alpha^2} \text{tr}(BB^T C^T C).$$

By minimizing the function $g(\alpha)$, we obtain the quasi-optimal value α_{APSS} in the APSS preconditioner as follows:

$$(4.2) \quad \alpha_{\text{APSS}} = \sqrt[4]{\frac{\text{tr}(BB^T C^T C)}{m+n+l}}.$$

5. Numerical experiments. In this section, we use several examples to investigate the feasibility and effectiveness of the MAPSS preconditioner for solving the block three-by-three saddle point problem (1.1).

The numerical experiments were performed using MATLAB (version R2020a) on a personal computer with a 2.40 GHz central processing unit (Intel(R) Core(TM) i5-1135G7 CPU), 16GB memory, and the Windows 10 operating system. The number of iteration steps (IT) and the elapsed CPU time in seconds (CPU) are measured for the GMRES and preconditioned GMRES iteration methods. All experiments are run five times, and CPU and IT denotes the corresponding averaged quantities, with the value of IT being rounded to the nearest integer.

When preconditioners are used to accelerate the convergence rate of the GMRES method, we solve several linear subsystems at each iteration step using direct methods, such as the Cholesky factorization in combination with an *approximate minimum degree* (AMD) reordering.

In actual computations, the initial vector is chosen to be $u^{(0)} = 0 \in \mathbb{R}^{n+m+l}$, and the iteration process is terminated if the current iterate $u^{(t)}$ satisfies

$$\text{RES} := \frac{\|\mathcal{A}u^{(t)} - b\|_2}{\|b\|_2} < 10^{-6}$$

or if the number of iteration steps satisfies $\text{IT} > 3000$. We use the symbol “-” to denote that the value of IT of the corresponding iteration method exceeds the value 3000, and we set the right-hand side vector $b \in \mathbb{R}^{n+m+l}$ according to the following two cases:

Case I: the vector $b \in \mathbb{R}^{n+m+l}$ is such that the exact solution of the linear system (1.1) is $u_* = [1, 1, \dots, 1]^T \in \mathbb{R}^{n+m+l}$.

Case II: the vector $b \in \mathbb{R}^{n+m+l}$ is such that the exact solution of the linear system (1.1) is $u_* = \text{rand}(n+m+l, 1) \in \mathbb{R}^{n+m+l}$, where $\text{rand}(\cdot, 1)$ denotes a column vector whose components are generated randomly.

For the MAPSS preconditioner, the theoretical quasi-optimal value α_{MAPSS} of the parameter α is computed according to formula (4.1), and the parameter β is chosen to be 10^{-4} for both Cases I and II. For the APSS preconditioner, the theoretical quasi-optimal value α_{APSS} of the parameter α is computed according to the formula (4.2). Furthermore, we obtain the experimentally optimal parameter α_{exp} of the APSS preconditioner and the experimentally optimal parameters α_{exp} and β_{exp} of the MAPSS preconditioner according to the minimum number of iteration steps.

EXAMPLE 5.1 ([4]). Consider the block three-by-three saddle point problem (1.1), whose coefficient matrix has the following block structure:

$$A = \text{Diag}(2W^T W + D_1, D_2, D_3) \in \mathbb{R}^{n \times n}$$

is a block-diagonal matrix,

$$B = [E, -I_{2p^2}, I_{2p^2}] \in \mathbb{R}^{m \times n} \quad \text{and} \quad C = E^T \in \mathbb{R}^{l \times m}$$

are both full row-rank matrices, where $W = [w_{ij}] \in \mathbb{R}^{p(p+1) \times p(p+1)}$ has entries $w_{ij} = e^{-2((i/3)^2 + (j/3)^2)}$. $D_1 = I_{p(p+1)}$ is the identity matrix, and the diagonal matrices $D_i = \text{diag}(d_1^{(i)}, \dots, d_{2p^2}^{(i)}) \in \mathbb{R}^{2p^2 \times 2p^2}$ ($i = 2, 3$) are defined by

$$d_j^{(2)} = \begin{cases} 1, & \text{for } 1 \leq j \leq p^2, \\ 10^{-5}(j - p^2)^2 & \text{for } p^2 + 1 \leq j \leq 2p^2, \end{cases}$$

$$d_j^{(3)} = 10^{-5}(j + p^2)^2, \quad \text{for } 1 \leq j \leq 2p^2.$$

TABLE 5.1
Numerical results for Example 5.1 (Case I).

	p	16	32	64	128	256
GMRES	IT	207	452	1272	-	-
	CPU	0.0690	2.0124	44.3523	-	-
APSS-GMRES(α_{APSS})	IT	43	34	23	13	5
	CPU	0.0246	0.0628	0.1773	0.5120	1.7554
APSS-GMRES(α_{exp})	IT	36	29	18	8	3
	CPU	0.0163	0.0528	0.1633	0.4113	1.3186
MAPSS-GMRES(α_{MAPSS})	IT	15	13	8	4	3
	CPU	0.0085	0.0255	0.0812	0.2547	1.4031
MAPSS-GMRES($\alpha_{\text{exp}}, \beta_{\text{exp}}$)	IT	14	13	8	4	2
	CPU	0.0053	0.0256	0.0755	0.2763	1.1983

Finally

$$E = \begin{bmatrix} \check{E} \otimes I_p \\ I_p \otimes \check{E} \end{bmatrix}, \quad \check{E} = \begin{bmatrix} 2 & -1 & & & \\ & 2 & -1 & & \\ & & \ddots & \ddots & \\ & & & 2 & -1 \end{bmatrix} \in \mathbb{R}^{p \times (p+1)},$$

where \otimes denotes the Kronecker product.

TABLE 5.2
Numerical results for Example 5.1 (Case II).

	p	16	32	64	128	256
GMRES	IT	265	530	1517	-	-
	CPU	0.1165	2.7710	67.5061	-	-
APSS-GMRES(α_{APSS})	IT	57	41	29	18	6
	CPU	0.0307	0.0787	0.2063	0.7427	1.7077
APSS-GMRES(α_{exp})	IT	44	35	25	15	5
	CPU	0.0227	0.0991	0.2247	0.6427	1.5768
MAPSS-GMRES(α_{MAPSS})	IT	18	18	15	7	3
	CPU	0.0093	0.0339	0.1183	0.3703	1.2305
MAPSS-GMRES($\alpha_{\text{exp}}, \beta_{\text{exp}}$)	IT	17	16	13	7	3
	CPU	0.0064	0.0313	0.1078	0.3143	1.2020

From Tables 5.1 and 5.2, we see that GMRES converges very slowly in both cases. However, for the APSS-preconditioned and MAPSS-preconditioned GMRES methods, the value of IT becomes very small and that of CPU is also not large when the problem size becomes very large. Whether using experimentally optimal values or quasi-optimal values, the MAPSS preconditioner is more efficient than the APSS preconditioner in accelerating the convergence rate of the GMRES method in terms of both IT and CPU. Moreover, it can be seen that the numerical results of the APSS and MAPSS preconditioners with quasi-optimal parameters are almost the same as the results with optimal parameter values. It implies that our strategy of parameter selection is effective.

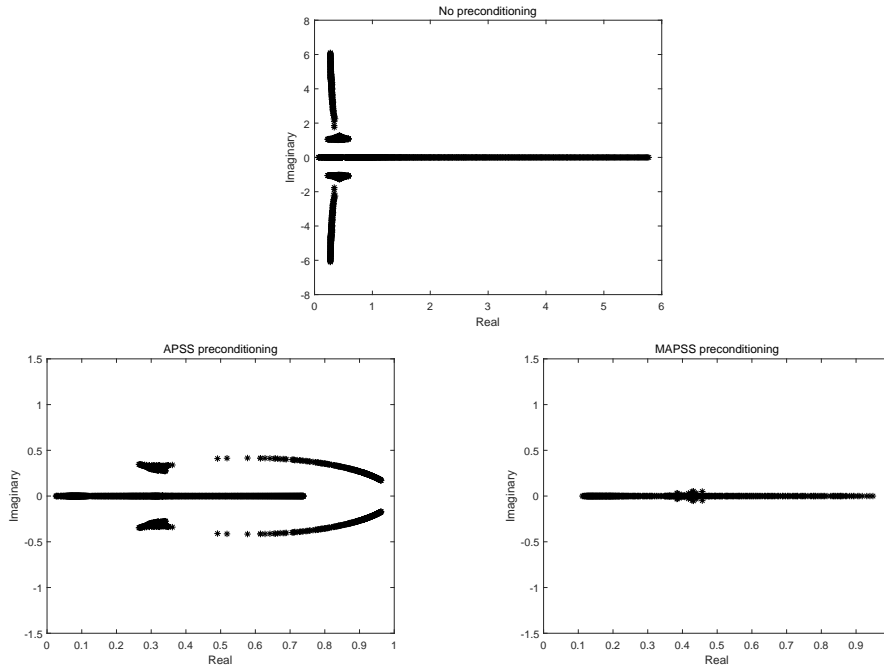


FIG. 5.1. Eigenvalue distribution of the preconditioned matrices with $p = 16$ for Example 5.1.

EXAMPLE 5.2. Consider the block three-by-three saddle point problem (1.1) whose coefficient matrix has the following block structure:

$$\begin{aligned}
 A &= \frac{1}{10} \text{sprandsym}(n, 0.01) + W_n, \\
 B &= [W_m, \text{sprand}(m, n - m, 0.05)], \\
 C &= [W_l, \text{sprand}(l, m - l, 0.05)],
 \end{aligned}$$

where $W_\ell = \text{diag}(1, 2, \dots, \ell)$ ($\ell = l, m, n$) is a diagonal matrix, $\text{sprandsym}(n, 0.01) \in \mathbb{R}^{n \times n}$ is a sparse symmetric random matrix with approximately $0.01n^2$ nonzeros and each entry is the sum of one or more normally distributed random samples, and $\text{sprand}(m, l, 0.05) \in \mathbb{R}^{m \times l}$ is a sparse random matrix with approximately $0.05ml$ uniformly distributed nonzero entries. In this example, we choose $m = 0.8n$ and $l = 0.6n$.

For the two different right-hand sides b , the values of IT and CPU for the preconditioned GMRES methods applied to Example 5.2 are presented in Tables 5.3 and 5.4, respectively. Comparing these tables with the numerical results of Example 5.1, we see that for this example there is a great difference in IT and CPU for all tested methods. We notice that both the GMRES method and the APSS-preconditioned GMRES method with quasi-optimal parameter converge very slowly for this example, while the values of IT and CPU of the MAPSS-preconditioned GMRES method with quasi-optimal parameter are always small, even when the problem size n becomes larger and larger. The results of the MAPSS-preconditioned GMRES method with quasi-optimal parameter are almost the same as that with an experimentally optimal parameter. So it is appropriate for the MAPSS preconditioner to use our formula for choosing parameters. We can also observe that the numerical results of the MAPSS preconditioner with quasi-optimal parameter are better than those produced by the APSS preconditioner with the optimal parameter.

TABLE 5.3
Numerical results for Example 5.2 (Case I).

	n	500	1000	2000	3000	4000
GMRES	IT	1029	2024	-	-	-
	CPU	1.1499	9.4198	-	-	-
APSS-GMRES(α_{APSS})	IT	332	581	995	1360	1699
	CPU	0.5570	4.2169	28.5816	83.6772	181.8035
APSS-GMRES(α_{exp})	IT	56	71	89	103	113
	CPU	0.0275	0.1238	0.7426	1.8593	3.7577
MAPSS-GMRES(α_{MAPSS})	IT	14	14	14	16	17
	CPU	0.0138	0.0544	0.2490	0.5997	1.1653
MAPSS-GMRES($\alpha_{\text{exp}}, \beta_{\text{exp}}$)	IT	10	11	12	13	14
	CPU	0.0153	0.0324	0.1833	0.4803	0.9341

TABLE 5.4
Numerical results for Example 5.2 (Case II).

	n	500	1000	2000	3000	4000
GMRES	IT	1023	2024	-	-	-
	CPU	1.1639	9.6227	-	-	-
APSS-GMRES(α_{APSS})	IT	335	581	1011	1349	1703
	CPU	0.6345	3.4261	30.6293	99.8449	184.3075
APSS-GMRES(α_{exp})	IT	56	71	90	103	113
	CPU	0.0204	0.1203	0.7519	1.8305	3.7833
MAPSS-GMRES(α_{MAPSS})	IT	14	15	15	17	18
	CPU	0.0114	0.0414	0.2493	0.7051	1.1741
MAPSS-GMRES($\alpha_{\text{exp}}, \beta_{\text{exp}}$)	IT	10	10	12	13	14
	CPU	0.0088	0.0583	0.1877	0.4553	0.9745

EXAMPLE 5.3 ([13]). Consider the block three-by-three saddle point problem (1.1) whose coefficient matrix has the following block structure:

$$\begin{aligned}
 A &= \begin{bmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{bmatrix} \in \mathbb{R}^{2p^2 \times 2p^2}, \\
 B &= [I \otimes F \quad F \otimes I] \in \mathbb{R}^{p^2 \times 2p^2}, \\
 C &= E \otimes F \in \mathbb{R}^{p^2 \times p^2},
 \end{aligned}$$

where

$$\begin{aligned}
 T &= \frac{1}{h^2} \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{p \times p}, \\
 F &= \frac{1}{h} \cdot \text{tridiag}(0, 1, -1) \in \mathbb{R}^{p \times p}, \\
 E &= \text{diag}(1, p+1, 2p+1, 3p+1, \dots, (p-1)p+1) \in \mathbb{R}^{p \times p},
 \end{aligned}$$

and $h = \frac{1}{p+1}$ is the discretization mesh size.

We display the numerical results of Example 5.3 in Tables 5.5 and 5.6. We note that these results are slightly different from those of Example 5.1. For this example, both the

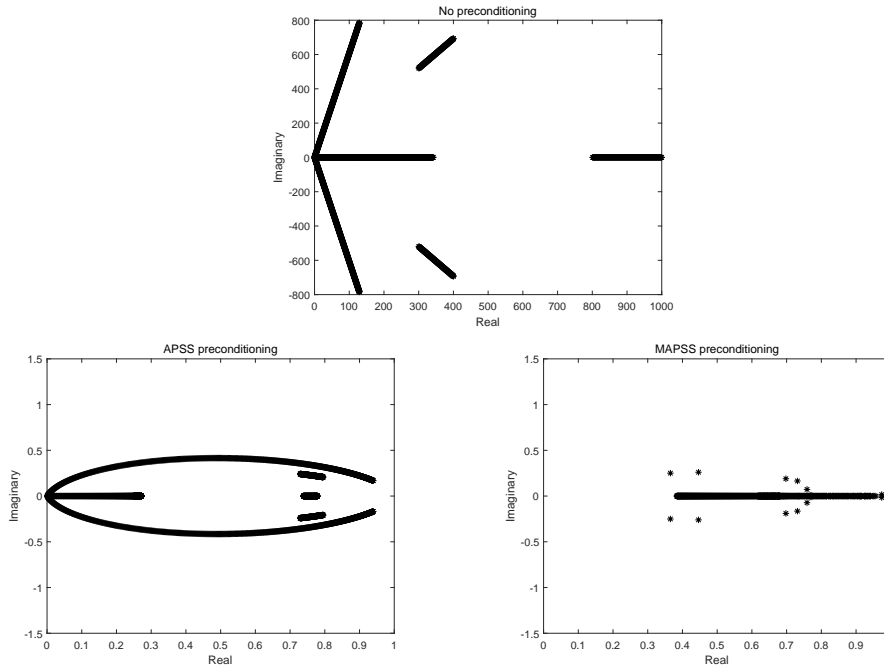


FIG. 5.2. Eigenvalue distribution of the preconditioned matrices with $n = 1000$ for Example 5.2.

GMRES and APSS-preconditioned GMRES method with quasi-optimal parameter converge slowly, as in Example 5.2. In addition, it can be seen in Tables 5.5 and 5.6 that the APSS preconditioner with optimal parameter performs good. However, all numerical results of the MAPSS preconditioner are better than those of the APSS preconditioner in terms of both IT and CPU, which implies that our proposed method is the best of all tested methods and that our selection strategy of the parameter is appropriate.

In Figures 5.1, 5.2, and 5.3, we depict the eigenvalue distribution of the original coefficient matrices of the three examples and of the same matrices preconditioned with the APSS and MAPSS preconditioners. It is evident that the APSS preconditioner greatly improves the eigenvalue distribution, while the MAPSS-preconditioned matrix has a tighter spectrum that leads to a stronger stability and a better numerical performance.

Finally, numerical results show that our parameter selection strategy is not very good for the APSS preconditioner in Example 5.1, but it is very effective for the MAPSS preconditioner in all examples. Furthermore, the MAPSS preconditioner can achieve a higher computational efficiency than the APSS preconditioner in accelerating the convergence rate of the GMRES method.

6. Conclusion. When Krylov subspace iteration methods are used to solve block three-by-three linear systems, the use of good preconditioners is very important [3, 5, 6, 7, 14]. In this paper, we have presented an MAPSS preconditioner for solving block three-by-three saddle point problems and analyzed spectral properties of the MAPSS-preconditioned matrix. Moreover, we have provided fast and effective formulas for computing the quasi-optimal values of the parameters involved in the MAPSS preconditioner. Numerical experiments show that the MAPSS preconditioner with quasi-optimal parameter is an efficient and robust solver for the block three-by-three linear system (1.1).

TABLE 5.5
Numerical results for Example 5.3 (Case I).

	p	16	32	64	96	128
GMRES	IT	994	-	-	-	-
	CPU	0.9550	-	-	-	-
APSS-GMRES(α_{APSS})	IT	295	914	-	-	-
	CPU	0.3759	15.5429	-	-	-
APSS-GMRES(α_{exp})	IT	10	10	10	10	10
	CPU	0.0026	0.0127	0.0569	0.1392	0.2833
MAPSS-GMRES(α_{MAPSS})	IT	6	6	7	7	8
	CPU	0.0028	0.0107	0.0466	0.1112	0.2175
MAPSS-GMRES($\alpha_{\text{exp}}, \beta_{\text{exp}}$)	IT	5	5	4	4	4
	CPU	0.0024	0.0082	0.0316	0.0831	0.1494

TABLE 5.6
Numerical results for Example 5.3 (Case II).

	p	16	32	64	96	128
GMRES	IT	986	-	-	-	-
	CPU	0.9373	-	-	-	-
APSS-GMRES(α_{APSS})	IT	294	822	-	-	-
	CPU	0.3725	12.7610	-	-	-
APSS-GMRES(α_{exp})	IT	10	10	9	8	8
	CPU	0.0044	0.0125	0.0505	0.1298	0.2338
MAPSS-GMRES(α_{MAPSS})	IT	5	6	6	7	8
	CPU	0.0027	0.0119	0.0462	0.1066	0.2092
MAPSS-GMRES($\alpha_{\text{exp}}, \beta_{\text{exp}}$)	IT	5	4	4	3	3
	CPU	0.0017	0.0073	0.0341	0.0599	0.1352

REFERENCES

- [1] Z.-Z. BAI, *The convergence of parallel iteration algorithms for linear complementarity problems*, Comput. Math. Appl., 32 (1996), pp. 1–17.
- [2] ———, *A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations*, Adv. Comput. Math., 10 (1999), pp. 169–186.
- [3] ———, *Sharp error bounds of some Krylov subspace methods for non-Hermitian linear systems*, Appl. Math. Comput., 109 (2000), pp. 273–285.
- [4] ———, *Optimal parameters in the HSS-like methods for saddle-point problems*, Numer. Linear Algebra Appl., 16 (2009), pp. 447–479.
- [5] ———, *Block alternating splitting implicit iteration methods for saddle-point problems from time-harmonic eddy current models*, Numer. Linear Algebra Appl., 19 (2012), pp. 914–936.
- [6] ———, *Eigenvalue estimates for saddle point matrices of Hermitian and indefinite leading blocks*, J. Comput. Appl. Math., 237 (2013), pp. 295–306.
- [7] ———, *Motivations and realizations of Krylov subspace methods for large sparse linear systems*, J. Comput. Appl. Math., 283 (2015), pp. 71–78.
- [8] ———, *Regularized HSS iteration methods for stabilized saddle-point problems*, IMA J. Numer. Anal., 39 (2019), pp. 1888–1923.
- [9] Z.-Z. BAI AND M. BENZI, *Regularized HSS iteration methods for saddle-point linear systems*, BIT, 57 (2017), pp. 287–311.
- [10] Z.-Z. BAI, M. BENZI, AND F. CHEN, *Modified HSS iteration methods for a class of complex symmetric linear systems*, Computing, 87 (2010), pp. 93–111.

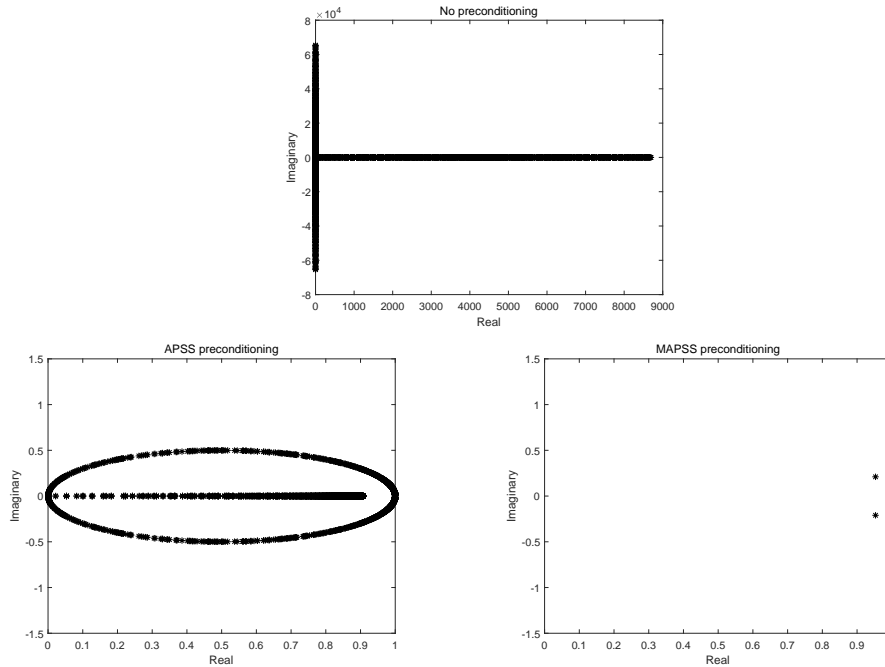


FIG. 5.3. Eigenvalue distribution of the preconditioned matrices with $p = 32$ for Example 5.3.

- [11] Z.-Z. BAI, G. H. GOLUB, L.-Z. LU, AND J.-F. YIN, *Block triangular and skew-Hermitian splitting methods for positive-definite linear systems*, SIAM J. Sci. Comput., 26 (2005), pp. 844–863.
- [12] Z.-Z. BAI, G. H. GOLUB AND M. K. NG, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.
- [13] Z.-Z. BAI, G. H. GOLUB, AND J.-Y. PAN, *Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems*, Numer. Math., 98 (2004), pp. 1–32.
- [14] Z.-Z. BAI AND J.-Y. PAN, *Matrix Analysis and Computations*, SIAM, Philadelphia, 2021.
- [15] Z.-Z. BAI, B. N. PARLETT, AND Z.-Q. WANG, *On generalized successive overrelaxation methods for augmented linear systems*, Numer. Math., 102 (2005), pp. 1–38.
- [16] Z.-Z. BAI AND M. TAO, *Rigorous convergence analysis of alternating variable minimization with multiplier methods for quadratic programming problems with equality constraints*, BIT, 56 (2016), pp. 399–422.
- [17] Z.-Z. BAI AND Z.-Q. WANG, *On parameterized inexact Uzawa methods for generalized saddle point problems*, Linear Algebra Appl., 428 (2008), pp. 2900–2932.
- [18] Z.-Z. BAI, J.-F. YIN, AND Y.-F. SU, *A shift-splitting preconditioner for non-Hermitian positive definite matrices*, J. Comput. Math., 24 (2006), pp. 539–552.
- [19] F. P. A. BEIK, M. BENZI, AND S.-H. A. CHAPARPORDI, *On block diagonal and block triangular iterative schemes and preconditioners for stabilized saddle point problems*, J. Comput. Appl. Math., 326 (2017), pp. 15–30.
- [20] M. BENZI, S. DEPARIS, G. GRANDPERRIN, AND A. QUARTERONI, *Parameter estimates for the relaxed dimensional factorization preconditioner and application to hemodynamics*, Comput. Methods Appl. Mech. Engrg., 300 (2016), pp. 129–145.
- [21] M. BENZI AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
- [22] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
- [23] M. BENZI AND X.-P. GUO, *A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations*, Appl. Numer. Math., 61 (2011), pp. 66–76.
- [24] M. BENZI AND M. K. NG, *Preconditioned iterative methods for weighted Toeplitz least squares problems*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1106–1124.
- [25] M. BENZI, M. K. NG, Q. NIU, AND Z. WANG, *A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations*, J. Comput. Phys., 230 (2011), pp. 6185–6202.

- [26] Y. CAO, *Shift-splitting preconditioners for a class of block three-by-three saddle point problems*, Appl. Math. Lett., 96 (2019), pp. 40–46.
- [27] Y. CAO, J. DU, AND Q. NIU, *Shift-splitting preconditioners for saddle point problems*, J. Comput. Appl. Math., 272 (2014), pp. 239–250.
- [28] Y. CAO, Z. REN, AND L. YAO, *Improved relaxed positive-definite and skew-Hermitian splitting preconditioners for saddle point problems*, J. Comput. Math., 37 (2019), pp. 95–111.
- [29] F. CHEN, *On choices of iteration parameter in HSS method*, Appl. Math. Comput., 271 (2015), pp. 832–837.
- [30] F. CHEN, T.-Y. LI, AND K.-Y. LU, *Updated preconditioned Hermitian and skew-Hermitian splitting-type iteration methods for solving saddle-point problems*, Comput. Appl. Math., 39 (2020), Art. 162, 14 pages.
- [31] H. C. ELMAN AND G. H. GOLUB, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.
- [32] C. GREIF, *Structured shifts for skew-symmetric matrices*, Electron. Trans. Numer. Anal., 55 (2022), pp. 455–468.
<https://etna.ricam.oeaw.ac.at/vol.55.2022/pp455-468.dir/pp455-468.pdf>
- [33] G. H. GOLUB, X. WU, AND J.-Y. YUAN, *SOR-like methods for augmented systems*, BIT, 41 (2001), pp. 71–85.
- [34] X.-P. GUO AND A. HADJIDIMOS, *Optimal accelerated SOR-like (ASOR) method for singular symmetric saddle point problems*, J. Comput. Appl. Math., 370 (2020), Art. 112662, 12 pages.
- [35] A. HADJIDIMOS, *Successive overrelaxation (SOR) and related methods*, Comput. Appl. Math, 123 (2000), pp. 177–199.
- [36] D. HAN AND X. YUAN, *Local linear convergence of the alternating direction method of multipliers for quadratic programs*, SIAM J. Numer. Anal., 51 (2013), pp. 3446–3457.
- [37] Y.-M. HUANG, *A practical formula for computing optimal parameters in the HSS iteration methods*, J. Comput. Appl. Math., 255 (2014), pp. 142–149.
- [38] X. LI, Y.-J. WU, A.-L. YANG, AND J.-Y. YUAN, *Modified accelerated parameterized inexact Uzawa method for singular and nonsingular saddle point problems*, Appl. Math. Comput., 244 (2014), pp. 552–560.
- [39] Z.-Z. LIANG AND G.-F. ZHANG, *Alternating positive semidefinite splitting preconditioners for double saddle point problems*, Calcolo, 56 (2019), Art. 26, 17 pages.
- [40] L.-D. LIAO AND G.-F. ZHANG, *A generalized variant of simplified HSS preconditioner for generalized saddle point problems*, Appl. Math. Comput., 346 (2019), pp. 790–799.
- [41] M. K. NG AND J. PAN, *Weighted Toeplitz regularized least squares computation for image restoration*, SIAM J. Sci. Comput., 36 (2014), pp. B94–B121.
- [42] D. K. SALKUYEH, H. ASLANI, AND Z.-Z. LIANG, *An alternating positive semidefinite splitting preconditioner for the three-by-three block saddle point problems*, Math. Commun., 26 (2021), pp. 177–195.
- [43] V. SIMONCINI AND M. BENZI, *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004/05), pp. 377–389.
- [44] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, 1962.
- [45] L.-N. WANG AND K. ZHANG, *Generalized shift-splitting preconditioner for saddle point problems with block three-by-three structure*, Open Access Lib. J., 6 (2019), Art. e5968, 13 pages.
- [46] X. XIE AND H.-B. LI, *A note on preconditioning for the 3×3 block saddle point problem*, Comput. Math. Appl., 79 (2020), pp. 3289–3296.
- [47] A.-L. YANG, X. LI, AND Y.-J. WU, *On semi-convergence of the Uzawa-HSS method for singular saddle-point problems*, Appl. Math. Comput., 252 (2015), pp. 88–98.
- [48] D. M. YOUNG, *Iterative Solutions of Large Linear Systems*, Academic Press, New York, 1971.
- [49] J. Y. YUAN, *Numerical methods for generalized least squares problems*, J. Comput. Appl. Math., 66 (1996), pp. 571–584.
- [50] G.-F. ZHANG, J.-L. YANG, AND S.-S. WANG, *On generalized parameterized inexact Uzawa method for a block two-by-two linear system*, J. Comput. Appl. Math., 255 (2014), pp. 193–207.
- [51] J.-H. ZHANG AND H. DAI, *A new block preconditioner for complex symmetric indefinite linear systems*, Numer. Algorithms, 74 (2017), pp. 889–903.
- [52] J.-L. ZHANG, *An efficient variant of HSS preconditioner for generalized saddle point problems*, Numer. Linear Algebra Appl., 25 (2018), Art. e2166, 14 pages.
- [53] S.-W. ZHOU, A.-L. YANG, Y. DOU, AND Y.-J. WU, *The modified shift-splitting preconditioners for nonsymmetric saddle-point problems*, Appl. Math. Lett., 59 (2016), pp. 100–114.