Filomat 35:15 (2021), 5181–5194
https://doi.org/10.2298/FIL2115181A

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

On the Preconditioning of Three-by-Three Block Saddle Point Problems

Hamed Aslani[a], Davod Khojasteh Salkuyeh[a,b], Fatemeh Panjeh Ali Beik[c]

[a]*Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran*
[b]*Center of Excellence for Mathematical Modelling, Optimization and Combinational Computing (MMOCC), University of Guilan, Rasht, Iran*
[c]*Department of Mathematics, Vali-e-Asr University of Rafsanjan, P.O. Box 518, Rafsanjan, Iran*

Abstract. We establish a new iterative method for solving a class of large and sparse linear systems of equations with three-by-three block coefficient matrices having saddle point structure. Convergence properties of the proposed method are studied in details and its induced preconditioner is examined for accelerating the convergence speed of generalized minimal residual (GMRES) method. More precisely, we analyze the eigenvalue distribution of the preconditioned matrix. Numerical experiments are reported to demonstrate the effectiveness of the proposed preconditioner.

1. Introduction

Consider the following three-by-three block system of linear equations,

$$\mathcal{A}\mathbf{x} \equiv \begin{pmatrix} A & B^T & 0 \\ B & 0 & C^T \\ 0 & C & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f \\ g \\ h \end{pmatrix}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{l\times m}$, $f \in \mathbb{R}^n$, $g \in \mathbb{R}^m$ and $h \in \mathbb{R}^l$ are known, and $\mathbf{x} = (x; y; z)$ is an unknown vector to be determined. Here, the MATLAB symbol $(x; y; z)$ is utilized to denote the vector $(x^T, y^T, z^T)^T$.

In the sequel, we assume that the matrix $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite and the matrices $B \in \mathbb{R}^{m\times n}$ and $C \in \mathbb{R}^{l\times m}$ have full row rank, where $n \geq m$ and $m \geq l$. These assumptions guarantee the existence of a unique solution of (1); see [31] for further details. Evidently matrix $\mathcal{A}$ can be regarded as a 2×2 block matrix using the following partitioning strategy,

$$\mathcal{A} = \left(\begin{array}{cc|c} A & B^T & 0 \\ B & 0 & C^T \\ \hline 0 & C & 0 \end{array}\right). \tag{2}$$

As seen, the above block matrix has a saddle point structure. Hence, we call Eq. (1) by three-by-three block saddle point problem.

2020 *Mathematics Subject Classification.* Primary 65F10; Secondary 65F50, 65F08
Keywords. iterative methods, sparse matrices, saddle point, convergence, preconditioning, Krylov methods
Received: 28 December 2020; Revised: 21 June 2021; Accepted: 25 August 2021
Communicated by Dijana Mosić
Email addresses: hamedaslani525@gmail.com (Hamed Aslani), khojasteh@guilan.ac.ir (Davod Khojasteh Salkuyeh), f.beik@vru.ac.ir (Fatemeh Panjeh Ali Beik)

Linear system of the form (1) arises from various practical scientific and engineering application backgrounds, e.g., the discrete finite element methods for solving time-dependent Maxwell equation with discontinuous coefficient [1, 14, 16, 17], the least squares problems [32], the Karush-Kuhn-Tucker (KKT) conditions of a type of quadratic program [20] and so on. In practice, stationary iterative methods for solving (1) tend to converge too slowly or even the produced sequence of approximations may fail to be convergent. For this reason, they are usually combined with acceleration schemes such as the Krylov subspace methods [26]. More precisely, preconditioners extracted from the convergent splittings of coefficient matrix $\mathcal{A}$ in Eq. (1) may be a suitable candidate to speed up the convergence of Krylov subspace methods. Here, we mainly consider on the preconditioned GMRES method.

As seen, the coefficient matrix $\mathcal{A}$ in Eq. (1) can be considered as a two-by-two block form given by (2). The observation was used in the literature for constructing preconditioners to improve the convergence speed of Krylov subspace methods for solving (1), such as block triangular preconditioners [2–5, 7, 13, 18], shift-splitting preconditioners [11, 12, 15, 28–30] and parameterized preconditioners [25]; for more details see also [5, 6, 27]. Recently, Huang and Ma [22] proposed the following block diagonal preconditioner,

$$\mathcal{P}_D = \begin{pmatrix} A & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & CS^{-1}C^T \end{pmatrix}, \tag{3}$$

for solving (1) in which $S = BA^{-1}B^T$. They also derive all the eigenpairs of preconditioned matrix. Xie and Li [31] presented the following three preconditioners

$$\mathcal{P}_1 = \begin{pmatrix} A & 0 & 0 \\ B & -S & C^T \\ 0 & 0 & CS^{-1}C^T \end{pmatrix}, \mathcal{P}_2 = \begin{pmatrix} A & 0 & 0 \\ B & -S & C^T \\ 0 & 0 & -CS^{-1}C^T \end{pmatrix}, \mathcal{P}_3 = \begin{pmatrix} A & B^T & 0 \\ B & -S & 0 \\ 0 & 0 & -CS^{-1}C^T \end{pmatrix},$$

and analyzed spectral properties of corresponding preconditioned matrices in the case $S = BA^{-1}B^T$. The reported numerical results in [31] show that the above preconditioners can significantly improve the convergence speed of GMRES method. It can be observed that the preconditioner $\mathcal{P}_1$ outperforms other preconditioners in terms of both required CPU time and number of iterations for the convergence.

Here, we consider the following equivalent form of (1):

$$\mathcal{B}\mathbf{x} \equiv \begin{pmatrix} A & B^T & 0 \\ -B & 0 & -C^T \\ 0 & C & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f \\ -g \\ h \end{pmatrix} = \mathbf{b}. \tag{4}$$

Although the coefficient matrix of the system (4) is not symmetric, it has some desirable properties. For instance, the matrix $\mathcal{B}$ is positive semidefinite, i.e., $\mathcal{B} + \mathcal{B}^T$ is symmetric positive semidefinite which can have a good effect on the performance of the GMRES method. In fact, the restarted version of GMRES(m) converges for all $m \geq 1$. Recently, some iterative schemes have been extended in the literature for solving (4). For instance, Cao [10] presented the shift-splitting (SS) method for block three-by-three saddle point and discussed on the performance and implementation of the following induced shift-splitting preconditioner. We comment that the preconditioner takes the following form when it is exploited for (4),

$$\mathcal{P}_{SS} = \begin{pmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & -C^T \\ 0 & C & \alpha I \end{pmatrix}, \tag{5}$$

here $\alpha > 0$ is given and I stands for the identity matrix with suitable dimension. In [23, 24], the Uzawa-type methods were developed. In this work, we present a new type of iterative method for solving three-by-three block saddle point problem (4). Next, we extract a preconditioner from the presented iterative method and examine its performance for speeding up the convergence of GMRES.

The remainder of this paper is organized as follows. Before ending this section, we present notations and review some basic preliminaries used in the following sections. In Section 2, we propose a new iterative

method for solving (4) and study its converges properties. In Section 3, we extract a preconditioner from the proposed method and analyze the spectrum of preconditioned matrix. Brief discussions are given in Section 4 about practical implementation of the preconditioner. In Section 5, we report some numerical results and brief concluding remarks are included in Section 6.

Throughout this paper, the identity matrix is denoted by I. The symbol x^* is used for the conjugate transpose of the vector x. For any square matrix A with real eigenvalues, the minimum and maximum eigenvalues of A are indicated by $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, respectively. The notation $\rho(A)$ stands for the spectral radius of A. The matrix $A \in \mathbb{R}^{n\times n}$ is called symmetric positive definite (SPD), if $A^T = A$ and $x^TAx > 0$ for all nonzero $x \in \mathbb{R}^n$. Similarly, the matrix A is called symmetric positive semidefinite (SPSD), if $A^T = A$ and $x^TAx \geqslant 0$ for all $x \in \mathbb{R}^n$. We write $A \succ 0$ ($A \succeq 0$), if A is SPD (SPSD). For two given matrices A and B, $A \succ B$ ($A \succeq B$) means that $A - B \succ 0$ ($A - B \succeq 0$). The matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive (semi-) definite, if $A + A^T$ symmetric positive (semi-) definite. For any matrix W, we shall write its null space as $null(W)$. The norm $\| . \|$ indicates the 2-norm.

2. The proposed iteration scheme

Let us first consider the following splitting for the coefficient matrix $\mathcal{B}$ in (4):

$$\mathcal{B} = \mathcal{P} - \mathcal{R}, \tag{6}$$

where

$$\mathcal{P} = \begin{pmatrix} A & B^T & 0 \\ 0 & S & -C^T \\ 0 & C & 0 \end{pmatrix}, \quad \mathcal{R} = \begin{pmatrix} 0 & 0 & 0 \\ B & S & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

in which S is a given symmetric positive definite matrix. It is not difficult to verify that the matrix $\mathcal{P}$ is nonsingular. The iteration scheme associated with splitting (6) is given by

$$\mathbf{x}^{(k+1)} = \mathcal{G}\mathbf{x}^{(k)} + \mathbf{c}, \quad k = 0, 1, 2, \ldots, \tag{7}$$

where $\mathbf{x}^{(0)}$ is an initial guess, $\mathcal{G} = \mathcal{P}^{-1}\mathcal{R}$ is the iteration matrix and $\mathbf{c} = \mathcal{P}^{-1}\mathbf{b}$.

Now, we present sufficient conditions under which the iterative scheme (7) is convergent. To this end, we first need to recall the following theorem.

Theorem 2.1. *[21, Theorem 7.7.3] Let E and F be two $n \times n$ real symmetric matrices such E is positive definite and F is positive semidefinite. Then $E \succeq F$ if and only if $\rho(E^{-1}F) \leq 1$, and $E \succ F$ if and only if $\rho(E^{-1}F) < 1$.*

Theorem 2.2. *Let $A \succ 0$, $S \succ 0$ and B and C be full row rank matrices. If $2S \succ BA^{-1}B^T$ then the iterative method* (7) *converges to the unique solution of* (4) *for any initial guess.*

Proof. Let λ be an arbitrary eigenvalue of $\mathcal{G} = \mathcal{P}^{-1}\mathcal{R}$ with the corresponding eigenvector $w = (x; y; z)$. Consequently, we have $\mathcal{R}w = \lambda\mathcal{P}w$ which is equivalent to say that

$$\begin{cases} \lambda(Ax + B^Ty) = 0, & (8) \\ \lambda(Sy - C^Tz) = Bx + Sy, & (9) \\ \lambda Cy = 0. & (10) \end{cases}$$

Without loss of generality, we may assume that $\lambda \neq 0$. Obviously $y \neq 0$, otherwise in view of the positive definiteness of A and the assumption that C has full row rank we conclude that x and z are both zero vectors which is in contradiction with the fact that $(x; y; z)$ is an eigenvector. From Eqs. (8) *and* (10) *we can deduce that*

$$x = -A^{-1}B^Ty, \quad y^*C^T = 0.$$

Multiplying both sides of Eq. (9) *on the left by* y^* *and substituting the preceding equalities, we derive*

$$\lambda = 1 - \frac{y^*BA^{-1}B^Ty}{y^*Sy}.$$

This shows that the eigenvalues of $\mathcal{G}$ *are all real. By Theorem 2.1, it is immediate to conclude that* $\lambda_{\max}(S^{-1}BA^{-1}B^T) = \rho(S^{-1}BA^{-1}B^T) < 2$ *if and only if* $2S \succ BA^{-1}B^T$. *This fact together with Courant-Fisher inequality [26] can deduce that*

$$0 < \frac{y^*BA^{-1}B^Ty}{y^*Sy} \leq \lambda_{\max}(S^{-1}BA^{-1}B^T) < 2.$$

Therefore, we have

$$|1 - \frac{y^*BA^{-1}B^Ty}{y^*Sy}| < 1,$$

which completes the proof. □

We complete this section with a remark providing alternative sufficient conditions for convergence of iterative method (7) which are stronger than $2S \succ BA^{-1}B^T$, however, it might be easier to check the following sufficient conditions in some cases. To do so, we first remind the following two lemmas. The first one is a consequence of Weyl's Theorem, see [21, Theorem 4.3.1].

Lemma 2.3. *Suppose that E and F are two Hermitian matrices. Then,*

$$\begin{aligned} \lambda_{\max}(E+F) &\leq \lambda_{\max}(E) + \lambda_{\max}(F), \\ \lambda_{\min}(E+F) &\geq \lambda_{\min}(E) + \lambda_{\min}(F). \end{aligned}$$

Lemma 2.4. *[33] Suppose that E is a Hermitian negative definite matrix and F is Hermitian positive semidefinite. Then the eigenvalues of EF are real and satisfy*

$$\lambda_{\min}(E)\lambda_{\min}(F) \leq \lambda_{\max}(EF) \leq \lambda_{\max}(E)\lambda_{\min}(F),$$

$$\lambda_{\min}(E)\lambda_{\max}(F) \leq \lambda_{\min}(EF) \leq \lambda_{\max}(E)\lambda_{\max}(F).$$

Remark 2.5. *Notice that* $2S \succ BA^{-1}B^T$ *is equivalent to say that all eigenvalues of* $2S - BA^{-1}B^T$ *are positive, i.e.,* $\lambda_{\min}(2S - BA^{-1}B^T) > 0$. *From Lemma 2.3, it can be seen that*

$$\lambda_{\max}\left(BA^{-1}B^T\right) < 2\lambda_{\min}(S), \tag{11}$$

implies that $2S \succ BA^{-1}B^T$. *Using Lemma 2.4, one can deduce that the condition* (11) *is satisfied as soon as*

$$\|B\|^2 < 2\lambda_{\min}(A)\lambda_{\min}(S),$$

which follows from the fact that

$$\lambda_{\max}\left(BA^{-1}B^T\right) = \lambda_{\max}\left(A^{-1}B^TB\right) \leq \lambda_{\max}\left(A^{-1}\right)\lambda_{\max}\left(B^TB\right) = \frac{\|B\|^2}{\lambda_{\min}(A)}.$$

3. The induced preconditioner and its spectral analysis

From the splitting (6) we have

$$\mathcal{P}^{-1}\mathcal{B} = I - \mathcal{P}^{-1}\mathcal{R} = I - \mathcal{G}.$$

Therefore, under the conditions of Theorem 2.2 the eigenvalues of $\mathcal{P}^{-1}\mathcal{B}$ are contained in the interval $(0, 2]$. Thus,

$$\mathcal{P} = \begin{pmatrix} A & B^T & 0 \\ 0 & S & -C^T \\ 0 & C & 0 \end{pmatrix}, \tag{12}$$

can be used as a preconditioner to accelerate the convergence of Krylov subspace methods like GMRES for solving the system (4).

In the succeeding theorem, we investigate the spectral properties of $\mathcal{P}^{-1}\mathcal{B}$ in more details.

Theorem 3.1. *Let A be symmetric positive definite and B and C be of full row rank. Then all the eigenvalues of $\mathcal{P}^{-1}\mathcal{B}$ are real and nonzero. Furthermore, $\lambda = 1$ is an eigenvalue of algebraic multiplicity at least $n + l$ and its corresponding eigenvectors are of the form $(x; -S^{-1}Bx; z)$ where $x \in \mathbb{R}^n$ and $z \in \mathbb{C}^l$ such that x, z are not simultaneously zero. The remaining eigenvalues of $\mathcal{P}^{-1}\mathcal{B}$ are of the form*

$$\lambda = \frac{y^*BA^{-1}B^Ty}{y^*Sy},$$

and the corresponding eigenvectors are of the form $(-A^{-1}B^Ty; y; z)$ for all $0 \neq y \in null(C)$ and arbitrary z.

Proof. *Let λ be an arbitrary eigenvalue of $\mathcal{P}^{-1}\mathcal{B}$ with the corresponding eigenvector $(x; y; z)$, i.e.,*

$$\begin{cases} Ax + B^Ty = \lambda(Ax + B^Ty), & (13) \\ -Bx - C^Tz = \lambda(Sy - C^Tz), & (14) \\ Cy = \lambda Cy. & (15) \end{cases}$$

Let $x = 0$. If $\lambda \neq 1$, then $B^Ty = 0$ by (13) which implies $y = 0$. This along with (14) leads to $C^Tz = 0$. Since C is a full row rank matrix, then $z = 0$. Consequently, we have $(x; y; z) = (0; 0; 0)$ which contradicts with the fact that $(x; y; z)$ is an eigenvector. Now, we assume that $\lambda = 1$. By (14) and the positive definiteness of S, we derive $y = 0$. In fact, $\lambda = 1$ is an eigenvalue of $\mathcal{P}^{-1}\mathcal{B}$ with multiplicity l corresponding eigenvector $(0; 0; z)$ for any arbitrary $0 \neq z \in \mathbb{C}^l$.

In the following, we consider the case that $x \neq 0$. If $y = 0$, then Eqs. (13) and (14) are reduced to

$$Ax = \lambda Ax \quad and \quad -Bx - C^Tz = -\lambda C^Tz, \tag{16}$$

respectively. The first relation shows that $\lambda = 1$. By substituting it into the second equality of (16), we have $Bx = 0$. Therefore, the corresponding eigenvectors are of the form $(x; 0; z)$ with $0 \neq x \in null(B)$ and $z \in \mathbb{R}^l$. Notice that, in general, we can observe that $\lambda = 1$ and $(x; 0; z)$ is an eigenpair of $\mathcal{P}^{-1}\mathcal{B}$ where $x \in null(B)$ and x, z are not simultaneously zero.

In summary, using (14) and in view of the positive definiteness of S, we can conclude that $\lambda = 1$ and $(x; -S^{-1}Bx; z)$ is an eigenpair of $\mathcal{P}^{-1}\mathcal{B}$.

It is immediate to see that if x and y are both zero vectors then $\lambda = 1$ and z must be a nonzero vector. In rest of the proof, we assume that $x \neq 0$ and $y \neq 0$. If $\lambda \neq 1$, then from Eqs. (13) and (15), we observe that $x = -A^{-1}B^Ty$ and $Cy = 0$, respectively. Pre-multiplying both sides of (14) from left by y^* and substituting deduced x and z into (14), we get

$$\lambda = -\frac{y^*Bx}{y^*Sy} = \frac{y^*BA^{-1}B^Ty}{y^*Sy}.$$

Hence, the corresponding eigenvectors are of the form $(-A^{-1}B^Ty; y; z)$ for all $0 \neq y \in null(C) \subseteq \mathbb{R}^m$ and arbitrary z. □

Remark 3.2. *Let S be an arbitrary symmetric positive definite matrix. From Theorem 3.1, we see that the non-unit eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{B}$ satisfies*

$$0 < \frac{\lambda_{\min}(BA^{-1}B^T)}{\lambda_{\max}(S)} \leq \lambda = \frac{y^*BA^{-1}B^Ty}{y^*Sy} \leq \frac{\lambda_{\max}(BA^{-1}B^T)}{\lambda_{\min}(S)}.$$

Theorem 3.3. *Under the assumptions of Theorem 3.1, if $S = BA^{-1}B^T$, then all the eigenvalues of preconditioned matrix $\mathcal{H} = \mathcal{P}^{-1}\mathcal{B}$ are equal to 1 and its minimal polynomial is of degree 2.*

Proof. Consider the matrix $\mathcal{P}_D$ defined in Eq. (3) with $S = BA^{-1}B^T$. Obviously, $\mathcal{P}_D$ is symmetric positive definite, therefore there is a symmetric positive definite matrix $\mathcal{P}_D^{\frac{1}{2}}$ such that $\mathcal{P}_D = \mathcal{P}_D^{\frac{1}{2}}\mathcal{P}_D^{\frac{1}{2}}$. Similar to the proof of Theorem 3.1 in [31], we see that the matrix $\mathcal{H}$ is similar to the matrix

$$\begin{aligned}
\hat{\mathcal{H}} &:= \mathcal{P}_D^{\frac{1}{2}}\mathcal{H}\mathcal{P}_D^{-\frac{1}{2}} && (17)\\
&= \mathcal{P}_D^{\frac{1}{2}}\mathcal{P}^{-1}\mathcal{B}\mathcal{P}_D^{-\frac{1}{2}} && (18)\\
&= \left(\mathcal{P}_D^{-\frac{1}{2}}\mathcal{P}\mathcal{P}_D^{-\frac{1}{2}}\right)^{-1}\left(\mathcal{P}_D^{-\frac{1}{2}}\mathcal{B}\mathcal{P}_D^{-\frac{1}{2}}\right) && (19)\\
&= \begin{pmatrix} I & M^T & 0 \\ 0 & I & -N^T \\ 0 & N & 0 \end{pmatrix}^{-1}\begin{pmatrix} I & M^T & 0 \\ -M & 0 & -N^T \\ 0 & N & 0 \end{pmatrix}, && (20)
\end{aligned}$$

where $M = S^{-\frac{1}{2}}BA^{-\frac{1}{2}}$ and $N = (CS^{-1}C^T)^{-\frac{1}{2}}CS^{-\frac{1}{2}}$. It straightforward to verify that $MM^T = I$, $NN^T = I$ and

$$\hat{\mathcal{H}} = I + \begin{pmatrix} M^T(I - N^TN)M & M^T(I - N^TN) & 0 \\ (N^TN - I)M & N^TN - I & 0 \\ NM & N & 0 \end{pmatrix}.$$

Direct computation reveals that $(\hat{\mathcal{H}} - I)^2 = 0$. This shows that the minimal polynomial of $\hat{\mathcal{H}}$, as well as $\mathcal{H}$ is 2. □

Remark 3.4. *Theorem 3.3 shows that the complete version of the GMRES method for solving the system $\mathcal{P}^{-1}\mathcal{B}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$ will converge in two iterations in exact arithmetic.*

4. Implementation of the preconditioner

In the implementation of the preconditioner $\mathcal{P}$ in a Krylov subspace method like GMRES, in each iteration, a vector of the form $v = \mathcal{P}^{-1}w$ should be computed. To this end, all we need is to solve $\mathcal{P}v = w$ for v. If we set $v = (v_1; v_2; v_3)$ and $w = (w_1; w_2; w_3)$ in which $v_1, w_1 \in \mathbb{R}^n$, $v_2, w_2 \in \mathbb{R}^m$ and $v_3, w_3 \in \mathbb{R}^l$, then we need to solve the system

$$\begin{pmatrix} A & B^T & 0 \\ 0 & S & -C^T \\ 0 & C & 0 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}.$$

The following algorithm is given for solving the above linear system of equations.

Algorithm 4.1. *Computation of $(v_1; v_2; v_3) = \mathcal{P}^{-1}(w_1; w_2; w_3)$.*
1: Set $t_1 = w_3 - CS^{-1}w_2$;
2: Solve $(CS^{-1}C^T)v_3 = t_1$ using the Cholesky factorization of $CS^{-1}C^T$;
3: Set $t_2 = w_2 + C^Tv_3$;
4: Solve $Sv_2 = t_2$;
5: Set $t_3 = w_1 - B^Tv_2$;
6: Solve $Av_1 = t_3$ by the Cholesky factorization of A.

In the implementation of the preconditioner $\mathcal{P}$ using the above algorithm, two systems with the coefficient matrix S and two systems with the coefficient matrices $CS^{-1}C^T$ and A should be solved. Whereas, in the implementation of the peconditioners $\mathcal{P}_D$, $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$ three systems with the coefficient matrices S, $CS^{-1}C^T$ and A need to be solved. On the other hand, the preconditioner $\mathcal{P}$ differs by the original matrix $\mathcal{B}$ in less blocks in comparison with the preconditioners such as $\mathcal{P}_D$, $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. It is natural to expect that among the above mentioned preconditioners, the one given by $\mathcal{P}$ tend to provide a better approximation for $\mathcal{B}$. Therefore, using a suitable approximation of S, one can expect that the new proposed preconditioner outperforms the preconditioners $\mathcal{P}_D$, $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. This is the main advantage of our proposed preconditioner.

We end this section by pointing out to the choice of SPD matrix S. As seen, Remark 3.4 shows that $S = BA^{-1}B^T$ leads to an ideal case. However, by this choice, the resulting algorithm can be costly in general cases. Basically, a preconditioner is called "optimal", if the number of preconditioned iterations is independent of the size of the problem and the amount of work per iteration scales linearly with the size of the problem. Notice that for our test problems, total work (and, approximately, the corresponding CPU-time) should grow by a factor of 4 each time the value of p doubles.

In view of Remark 3.4 and the above discussions, in the numerical experiments, we are particularly inspired to set $S = I$, with I being the identity matrix or $S = \text{diag}(B\ \text{diag}(A)^{-1}B^T)$. For these choices, the proposed preconditioners, while not quite optimal, scales well with increasing the size of problem for our test examples. For more details, we further set $S = I$ or $S = \text{diag}(B\ \text{diag}(A)^{-1}B^T)$ while working with the preconditioners $\mathcal{P}_D$ and $\mathcal{P}_1$. In this work, we examine the exact versions of preconditioners in conjunction with complete version of GMRES.

In general cases, for approximating $BA^{-1}B^T$ by S, similar to [2–4], one can possibly avoid forming $BA^{-1}B^T$ and $CS^{-1}C^T$. Instead, using a prescribed tolerance, few steps of the (P)CG method can be used for the actions of A^{-1}, $(BA^{-1}B^T)^{-1}$ and $(CS^{-1}C^T)^{-1}$. For this inexact implementation, the preconditioner should be used in conjunction with flexible GMRES (FGMRES). For some problems, we may have access to the sparse matrix M, being spectrally equivalent to $BA^{-1}B^T$. In this case we can set $S = M$ and implement the preconditioner either exactly in conjunction with GMRES or inexactly in conjunction with FGMRES.

5. Numerical experiments

In this section, we numerically solve the three-by-three saddle point problem (4) to examine the performance of proposed preconditioner in Section 3. In order to compare the performance of our preconditioner with the recently proposed ones in the literature, test problems are taken from [22, 23, 31]. In all the test examples, we use the complete version of GMRES method with right preconditioning. All runs were started from the initial zero vector and terminated once the relative residual 2-norm satisfies,

$$\text{Res} := \frac{\|\mathbf{b} - \mathcal{B}\mathbf{x}^{(k)}\|}{\|\mathbf{b}\|} < 10^{-7},$$

where $\mathbf{x}^{(k)}$ is the current approximate solution. The maximum number of iterations is set to be 5000. For the preconditioner $\mathcal{P}_{SS}$, we set $\alpha = 0.01$ (see [10]). In all the tests, the right-hand side vector $\mathbf{b}$ is set to be $\mathbf{b} = \mathcal{B}e$, where $e \in \mathbb{R}^{\mathbf{n}}$ is a vector of all ones and $\mathbf{n} = n+m+l$. Numerical results are presented in the tables in which "IT" and "CPU" denote the number of iterations and elapsed CPU times in second, respectively. The symbols "†" and "‡" are used to indicate that the method has not converged in the maximum number of iterations and in 500 seconds, respectively. To show the accuracy of the methods we also report the values

$$\text{Err} := \frac{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|}{\|\mathbf{x}^*\|},$$

in the tables, where $\mathbf{x}^*$ stands for the exact solution of the system (4). All runs were performed in MATLAB R2017a with a personal computer with 2.40 GHz central processing unit (Intel(R) Core(TM) i7-5500), 8 GB memory and Windows 10 operating system.

Table 1: Numerical results for Example 5.1 with $S = I$.

Precon.	p **n**	64 16384	128 65536	256 262144	512 1048576
I	IT	†	†	†	†
	CPU	-	-	-	-
	Err	-	-	-	-
	Res	-	-	-	-
$\mathcal{P}_D$	IT	36	39	41	47
	CPU	0.38	1.90	13.73	114.90
	Err	1.46e-05	1.33e-05	1.08e-04	1.13e-04
	Res	5.16e-08	7.92e-07	2.14e-05	5.70e-05
$\mathcal{P}_1$	IT	28	30	30	32
	CPU	0.26	1.24	10.99	86.42
	Err	2.08e-06	6.50e-06	2.95e-05	5.79e-05
	Res	6.56e-08	1.60e-08	3.81e-06	6.12e-07
$\mathcal{P}_{SS}$	IT	3	3	3	-
	CPU	0.98	3.55	39.30	‡
	Err	2.05e-04	1.50e-04	3.05e-03	-
	Res	3.15e-8	1.61e-07	1.12e-7	-
$\mathcal{P}$	IT	2	2	2	6
	CPU	0.06	0.39	2.61	34.05
	Err	1.16e-11	6.50e-11	6.84e-10	5.02e-09
	Res	7.62e-13	2.37e-12	4.01e-12	7.44e-10

Example 5.1. [22, 31] Consider the saddle point problem (4) with

$$A = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2 \times 2p^2},$$

$B = (I \otimes F \quad F \otimes I) \in \mathbb{R}^{p^2 \times 2p^2}$ and $C = E \otimes F \in \mathbb{R}^{p^2 \times p^2}$ where

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

and $E = \text{diag}\left(1, p+1, 2p+1, \ldots, p^2 - p + 1\right)$ in which $\otimes$ denotes the Kronecker product and $h = 1/(p+1)$ stands for the discretization meshsize.

In this example, we set $S = I$, where I is the identity matrix. Table 1 shows the iteration counts and the elapsed CPU time for the GMRES method with the preconditioner $\mathcal{P}_D$, $\mathcal{P}_1$, $\mathcal{P}_{SS}$ and $\mathcal{P}$. To see the effectiveness of preconditioners, we have also reported the numerical results of the GMRES method without preconditioning. Numerical results illustrate that the preconditioners can significantly reduce the number of iterations and elapsed CPU time of the GMRES method without preconditioning. As seen, $\mathcal{P}$ is superior to the other examined preconditioners. An interesting observation which can be posed here is that the GMRES method with the preconditioner $\mathcal{P}$ gives the best accuracy among the preconditioners. As we see in many cases the GMRES of MATLAB stagnated for other preconditioners and two consecutive iterates are the same. In these cases the reported value of Res is greater than 10^{-7} (the chosen tolerance). We also see that, $S = I$ presents a good approximation of the matrix $S = BA^{-1}B^T$.

Table 2: Numerical results for Example 5.2 for the first choice with $S = I$.

Precon.	p **n**	32 8256	48 9216	64 32896	128 131328
I	IT	557	1180	1815	2128
	CPU	5.22	38.50	133.15	209.50
	Err	5.22e-06	5.67e-05	1.84e-04	6.61e-04
	Res	9.89e-08	9.99e-08	2.66e-07	3.90e-07
$\mathcal{P}_D$	IT	348	314	284	197
	CPU	10.97	13.51	14.84	21.75
	Err	4.87e-06	2.34e-05	7.51e-05	1.14e-03
	Res	9.92e-08	9.84e-08	9.84e-08	9.62e-07
$\mathcal{P}_1$	IT	171	159	144	103
	CPU	2.73	3.74	4.09	7.24
	Err	3.48e-06	2.19e-5	7.43e-05	1.07e-03
	Res	9.96e-08	9.26e-08	9.26e-08	9.47e-08
$\mathcal{P}_{SS}$	IT	9	7	7	-
	CPU	18.25	64.21	188.42	‡
	Err	1.19e-06	4.92e-06	5.12e-06	-
	Res	3.40e-08	2.54e-08	4.23e-08	-
$\mathcal{P}$	IT	2	2	2	2
	CPU	0.08	0.17	0.06	0.40
	Err	5.64e-09	1.00e-08	2.06e-08	1.82e-08
	Res	2.15e-10	7.65e-10	2.69e-10	1.98e-10

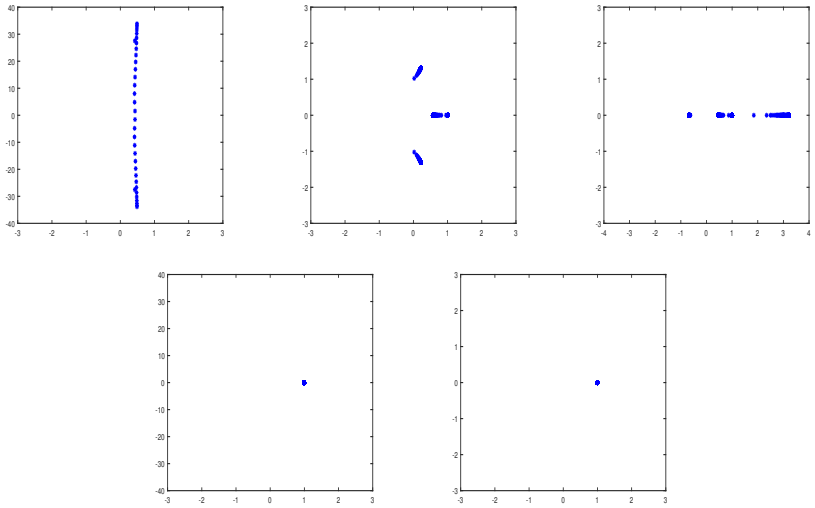

Figure 1: Eigenvalue distributions of $\mathcal{B}$, $\mathcal{P}_D^{-1}\mathcal{B}$, $\mathcal{P}_1^{-1}\mathcal{B}$, $\mathcal{P}_{SS}^{-1}\mathcal{B}$ and $\mathcal{P}^{-1}\mathcal{B}$ (from the left to right) with $S = I$ and $p = 16$ for Example 5.1.

Figure 1 plots the eigenvalues of the matrices $\mathcal{B}$, $\mathcal{P}_D^{-1}\mathcal{B}$, $\mathcal{P}_1^{-1}\mathcal{B}$, $\mathcal{P}_{SS}$ and $\mathcal{P}^{-1}\mathcal{B}$ for $p = 16$ with $S = I$. It is seen that the eigenvalues of $\mathcal{P}^{-1}\mathcal{B}$ are more clustered than the others.

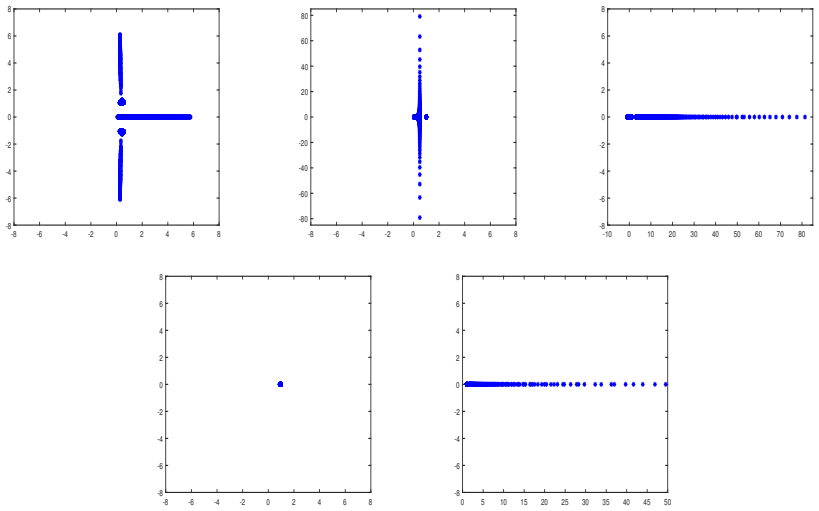

Figure 2: Eigenvalue distributions of $\mathcal{B}$, ${\mathcal{P}_D}^{-1}\mathcal{B}$, $\mathcal{P}_1^{-1}\mathcal{B}$, $\mathcal{P}_{SS}^{-1}\mathcal{B}$ and $\mathcal{P}^{-1}\mathcal{B}$ for the first choice with $S = I$ and $p = 16$ for Example 5.2.

Example 5.2. [22, 31] Consider the three-by-three block saddle point problem (1) for which

$$A = \text{bldiag}\left(2W^TW + D_1, D_2, D_3\right) \in \mathbb{R}^{n\times n},$$

is a block-diagonal matrix,

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

are both full row-rank matrices where $\tilde{p} = p^2$, $\hat{p} = p(p+1)$; $D_1 = I_{\widehat{p}}$ is an identity matrix; $D_i = \text{diag}(d_j^{(i)}) \in \mathbb{R}^{2\tilde{p}\times 2\tilde{p}}$, $i = 2, 3$, are diagonal matrices, with

$$d_j^{(2)} = \begin{cases} 1, & \text{for} \quad 1 \le j \le \tilde{p}, \\ 10^{-5}(j-\tilde{p})^2, & \text{for } \tilde{p}+1 \le j \le 2\tilde{p}, \end{cases}$$
$$d_j^{(3)} = 10^{-5}(j+\tilde{p})^2 \text{ for } 1 \le j \le 2\tilde{p},$$

and

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

Moreover, $W = vv^T \in \mathbb{R}^{\hat{p}\times\hat{p}}$, where $v \in \mathbb{R}^{\hat{p}}$ is an arbitrary vector. According to the above definitions, we have $n = \hat{p} + 4\tilde{p}$, $m = 2\tilde{p}$ and $l = \hat{p}$. We consider two choices for the vector v. In the first choice, the ith entry of the vector v is set to be $v_i = e^{-2(i/3)^2}$, $i = 1, 2, \ldots, l$, and in the second one the vector v is set to be a random sparse vector of order l with approximately $0.05l$ uniformly distributed nonzero entries (such a vector can be generated using the "`sprand`" command of MATLAB).

For both of the choices we set $S = I$. Numerical results for the first choice are presented in Table 2 and for the second choice in Table 3. All the other notations are as the previous example. As seen, the proposed preconditioner outperforms the others in terms of the iteration counts, the elapsed CPU time

Table 3: Numerical results for Example 5.2 for the second choice with $S = I$.

Precon.	p **n**	64 32896	128 131328	256 524800	512 2098176
I	IT	†	†	†	†
	CPU	-	-	-	-
	Err	-	-	-	-
	Res	-	-	-	-
$\mathcal{P}_D$	IT	279	193	125	119
	CPU	14.36	21.92	93.29	201.56
	Err	1.31e-04	2.19e-03	2.19e-02	3.72e-02
	Res	5.22e-05	8.42e-05	4.71e-04	3.92e-04
$\mathcal{P}_1$	IT	143	103	70	59
	CPU	4.08	8.02	56.08	149.11
	Err	1.30e-04	1.94e-03	2.17e-04	2.12e-03
	Res	6.33e-05	2.74e-05	5.84e-05	1.40e-4
$\mathcal{P}_{SS}$	IT	3	3	-	-
	CPU	108.51	146.19	‡	‡
	Err	2.20e-08	7.01e-08	-	-
	Res	3.11e-09	3.34e-09	-	-
$\mathcal{P}$	IT	2	2	2	4
	CPU	0.08	0.78	30.92	66.42
	Err	1.33e-09	4.09e-09	3.20e-09	2.41e-09
	Res	2.20e-10	5.51e-10	2.04e-10	1.68e-10

and the accuracy of computed solution. Figures 2 and 3 display the eigenvalue distribution of the original coefficient matrix, $\mathcal{P}_D^{-1}\mathcal{B}$, $\mathcal{P}_1^{-1}\mathcal{B}$, $\mathcal{P}_{SS}$ and $\mathcal{P}^{-1}\mathcal{B}$ for $S = I$ and $p = 16$ for the two choices, respectively. As observed, eigenvalues of $\mathcal{P}_{SS}$ are more clustered around the point $(1, 0)$ than the others.

Example 5.3. We consider the three-by-three block saddle point problem (4) with (see [23, 31])

$$\begin{aligned} &\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^l} \frac{1}{2}x^TAx + r^Tx + q^Ty \\ &s.t.: \quad Bx + C^Ty = b, \end{aligned} \tag{21}$$

where $r \in \mathbb{R}^n$ and $q \in \mathbb{R}^l$. To solve the above problem we define the Lagrange function

$$L(x, y, \lambda) = \frac{1}{2}x^TAx + r^Tx + q^Ty + \lambda^T(Bx + C^Ty - b),$$

where the vector $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier. Then the Karush-Kuhn-Tucker necessary conditions of (21) are as follows (see [9])

$$\nabla_x L(x, y, \lambda) = 0, \quad \nabla_y L(x, y, \lambda) = 0 \quad \text{and} \quad \nabla_\lambda L(x, y, \lambda) = 0.$$

It is easy to see that these equations give a system of linear equations of the form (1). In this example, we chose the matrices A, B and C from the CUTEr collection [19]. To do so, we have selected four matrices. In this example, we set $S = \text{diag}(B\ \text{diag}(A)^{-1}B^T)$ (see [8]). Numerical results are presented in Table 4. As we see the proposed preconditioner outperforms the others from the iteration counts, elapsed CPU time and accuracy of the computed solution point of view.

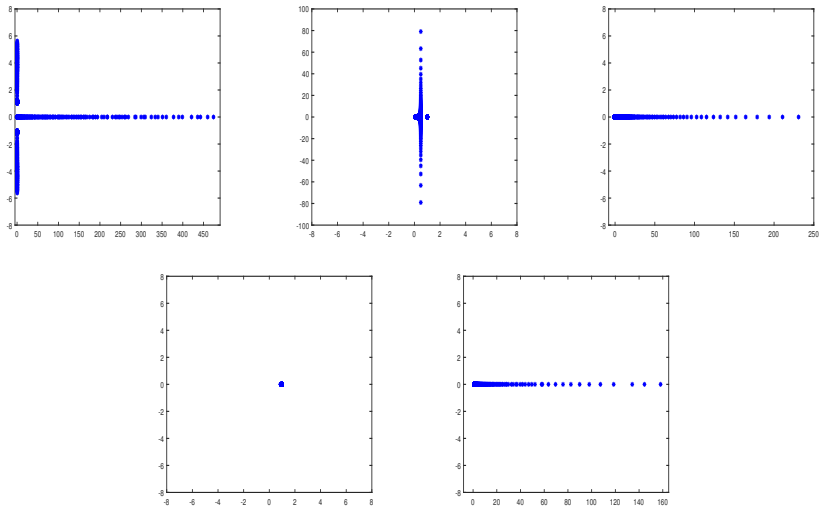

Figure 3: Eigenvalue distributions of $\mathcal{B}$, ${\mathcal{P}_D}^{-1}\mathcal{B}$, $\mathcal{P}_1^{-1}\mathcal{B}$, $\mathcal{P}_{SS}^{-1}\mathcal{B}$ and $\mathcal{P}^{-1}\mathcal{B}$ for the second choice with $S = I$ and $p = 16$ for Example 5.2.

Table 4: Numerical results for Example 5.3 with $S = \text{diag}(B\ \text{diag}(A)^{-1}B^T)$.

Precon.	Matrix **n**	AUG2DC 50400	AUG3DC 8746	LISWET12 30004	YAO 6004
I	IT	94	99	92	99
	CPU	2.12	0.93	1.52	0.67
	Err	4.82e-07	2.37e-07	5.80e-07	5.93e-07
	Res	9.86e-08	8.13e-08	9.32e-08	9.54e-08
$\mathcal{P}_D$	IT	101	136	52	57
	CPU	2.62	1.85	0.66	0.31
	Err	3.41e-07	1.84e-07	4.40e-07	3.11e-07
	Res	9.13e-08	9.80e-08	9.84e-08	6.79e-08
$\mathcal{P}_1$	IT	55	80	34	37
	CPU	0.92	0.65	0.32	0.14
	Err	3.68e-07	1.90e-7	3.88e-07	3.10e-09
	Res	8.44e-08	6.14e-08	9.33e-08	7.29e-09
$\mathcal{P}_{SS}$	IT	10	12	9	11
	CPU	13.56	0.91	0.07	0.08
	Err	1.08e-07	9.29e-8	8.13e-8	6.10e-8
	Res	9.34e-08	1.39e-08	8.85e-8	7.52e-08
$\mathcal{P}$	IT	22	29	4	4
	CPU	0.30	0.14	0.07	0.04
	Err	1.33e-07	1.14e-7	1.68e-14	1.69e-14
	Res	9.28e-08	6.14e-08	3.60e-15	2.89e-16

6. Conclusions

A new stationary iterative method was constructed for solving a class of three-by-three block saddle point problems. We analyzed the convergence properties of the elaborated stationary method. We further

examined the performance of induced preconditioner from the proposed method. More precisely, the eigenvalue distribution of the preconditioned matrix was studied. Our numerical tests illustrated that the proposed preconditioner is more effective than the other tested preconditioners in the literature.

Acknowledgments

The authors would like to thank the referees for their useful comments and suggestions. The work of the second author is partially supported by University of Guilan.

References

[1] F. Assous, P. Degond, E. Heintze, P.A. Raviart, J. Segre, On a finite-element method for solving the three-dimensional Maxwell equations, J. Comput. Phys. 109 (1993) 222–237.
[2] F. P. A. Beik and M. Benzi, Block preconditioners for saddle point systems arising from liquid crystal directors modeling, CALCOLO 55 (2018) 29.
[3] F. P. A. Beik and M. Benzi, Iterative methods for double saddle point systems, SIAM J. Matrix Anal. Appl. 39 (2018) 902–921.
[4] M. Benzi and F. P. A. Beik, Uzawa-type and augmented lagrangian methods for double saddle point systems, Structur Matrices in Numerical Linear Algebra (Prof. Dario Andrea Bini, Prof. Fabio Di Benedetto, Prof. Eugene Tyrtyshnikov, and Prof. Marc Van Barel, eds.), Springer International Publishing, 2019.
[5] M. Benzi, G. H. Golub and J. Liesen, Numerical Solution of Saddle Point Problems, Acta Numer. 14 (2005) 1–137.
[6] M. Benzi, G.H. Golub, A preconditioner for generalized saddle point problems, SIAM J. Matrix Anal. Appl. 26 (2004) 20-41.
[7] M. Benzi, M.K. Ng, Q. Niu, Z. Wang, A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations, J. Comput. Phys. 230 (2011) 6185–6202.
[8] L. Bergamaschi, J. Gondzio, G. Zilli, Preconditioning indefinite systems in interior point methods for optimization, Comput. Optim. Appl. 28 (2004) 149–171.
[9] D. P. Bertsekas, Nonlinear Programming, (2nd edition), Athena Scientific, 1999.
[10] Y. Cao, Shift-splitting preconditioners for a class of block three-by-three saddle point problems, Appl. Math. Lett. 96 (2019) 40–46.
[11] Y. Cao, J. Du, Q. Niu, Shift-splitting preconditioners for saddle point problems, J. Comput. Appl. Math. 270 (2014) 239-250.
[12] Y. Cao, Sen Li, L. Yao, A class of generalized shift-splitting preconditioners for nonsymmetric saddle point problems, Appl. Math. Lett. 49 (2015) 20-27.
[13] Z.-H. Cao, Positive stable block triangular preconditioners for symmetric saddle point problems, Appl. Numer. Math. 57 (2007) 899–910.
[14] Z.-M. Chen, Q. Du, J. Zou, Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients, SIAM J. Numer Anal. 37 (2000) 1542–1570.
[15] C.-R. Chen, C.-F. Ma, A generalized shift-splitting preconditioner for singular saddle point problems, Appl. Math. Comput. 269 (2015) 947–955.
[16] Z.-M. Chen, Q. Du, J. Zou, Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients, SIAM J. Numer. Anal. 37 (2000) 1542–1570.
[17] P. Ciarlet, J. Zou, Finite element convergence for the Darwin model to Maxwell's equations, RAIRO Math. Modelling Numer. Anal. 31 (1997) 213–249.
[18] H.C. Elman, D.J. Silvester, A.J. Wathen, Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations, Numer. Math. 90 (2002) 665–688.
[19] N. I. M. Gould, D. Orban, P. L. Toint, CUTEr and SifDec, a constrained and unconstrained testing environment, revisited, ACM Trans. Math. Softw. 29 (2003) 373–394.
[20] D.R. Han, X.M. Yuan, Local linear convergence of the alternating direction method of multipliers for quadratic programs, SIAM J. Numer. Anal. 51 (2013) 3446–3457.
[21] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, UK, 1985.
[22] N. Huang, C.-F. Ma, Spectral analysis of the preconditioned system for the 3×3 block saddle point problem, Numer. Algor. 81 (2019) 421–444.
[23] N. Huang, Variable parameter Uzawa method for solving a class of block three-by-three saddle point problems, Numer. Algor. 85 (2020) 1233-1254.
[24] N. Huang, Y.-H. Dai, Q. Hu, Uzawa methods for a class of block three-by-three saddle-point problems, Numer Linear Algebra Appl. 26 (2019) e2265.
[25] Y.-F. Ke, C.-F. Ma, The parameterized preconditioner for the generalized saddle point problems from the incompressible Navier-Stokes equations, J. Comput. Appl. Math. 37 (2018) 3385–3398.
[26] Y. Saad, Iterative Methods for Sparse Linear Systems, (2nd edition), Society for Industrial and Applied Mathematics, Philadelphia, 2003.
[27] D.K. Salkuyeh, M. Masoudi, A new relaxed HSS preconditioner for saddle point problems, Numer. Algor. 74 (2017) 781–795.
[28] D.K. Salkuyeh, M. Rahimian, A modification of the generalized shift-splitting method for singular saddle point problems, Comput. Math. Appl. 74 (2017) 2940–2949.
[29] D.K. Salkuyeh, M. Masoudi, D. Hezari, On the generalized shift-splitting preconditioner for saddle point problems, Appl. Math. Lett. 48 (2015) 55–61.

[30] Q.-Q. Shen, Q. Shi, Generalized shift-splitting preconditioners for nonsingular and singular generalized saddle point problems, Comput. Math. Appl. 72 (2016) 632-641.
[31] X. Xie, H.-B. Li, A note on preconditioning for the 3×3 block saddle point problem, Comput. Math. Appl. 79 (2020) 3289–3296.
[32] J.-Y. Yuan, Numerical methods for generalized least squares problems, J. Comput. Appl. Math. 66 (1996) 571–584.
[33] F. Zhang, Q. Zhang, Eigenvalue inequalities for matrix product, IEEE Trans. Automat. Control 51 (2006) 1506–1509.