

A New Uzawa-Type Iteration Method for Non-Hermitian Saddle-Point Problems

Yan Dou, Ai-Li Yang and Yu-Jiang Wu*

School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, P.R. China.

Received 29 August 2016; Accepted (in revised version) 13 January 2017.

Abstract. Based on a preconditioned shift-splitting of the $(1, 1)$ -block of non-Hermitian saddle-point matrix and the Uzawa iteration method, we establish a new Uzawa-type iteration method. The convergence properties of this iteration method are analyzed. In addition, based on this iteration method, a preconditioner is proposed. The spectral properties of the preconditioned saddle-point matrix are also analyzed. Numerical results are presented to verify the robustness and the efficiency of the new iteration method and the preconditioner.

AMS subject classifications: 65F08, 65F10, 65F20

Key words: Saddle-point problems, Uzawa method, preconditioned shift-splitting, convergence, preconditioner.

1. Introduction

Consider the following large, sparse non-Hermitian saddle-point linear system

$$\mathcal{A}x := \begin{pmatrix} A & B \\ -B^* & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ -g \end{pmatrix} =: b, \quad (1.1)$$

where $A \in \mathbb{C}^{n \times n}$ is a non-Hermitian positive definite matrix, i.e., its Hermitian part $H = (1/2)(A + A^*)$ is positive definite, $B \in \mathbb{C}^{n \times m}$ is a rectangular matrix of full column rank with $n \geq m$, and $b \in \mathbb{C}^{n+m}$ is a given vector. Here, $(\cdot)^*$ denotes the conjugate transpose of a matrix. This kind of systems of linear equations arises in a variety of scientific and engineering applications, such as computational fluid dynamics, optimal control, constrained optimization, weighted least-squares problems, electronic networks, computer graphic, mixed or hybrid finite element discretization of second-order elliptic problems and meshfree discretization of some partial differential equations; see [10, 13, 15, 17, 21, 24, 27, 28] and the references therein.

In order to solve the nonsingular saddle-point linear system (1.1), many effective iteration methods and preconditioning techniques have been proposed in the literature;

*Corresponding author. *Email address:* myjaw@lzu.edu.cn (Y.-J. Wu)

see [3–7, 9, 11, 13, 19, 20, 23] and the references therein. As one of the most important iteration methods, Uzawa method [1] can be described as follows:

$$\begin{cases} u_{k+1} = A^{-1}(f - Bv_k), \\ v_{k+1} = v_k + \tau(B^*u_{k+1} - g), \end{cases} \quad (1.2)$$

where $\tau > 0$ is a relaxation parameter. Owing to its simple form, minimal computer memory requirements and good numerical performance, the Uzawa method has received wide attention in the literature. Many variants of the Uzawa method, including preconditioned Uzawa [19] and parameterized Uzawa [10] methods, have been proposed to improve the efficiency of the original Uzawa method.

In each step of the Uzawa method, a linear system with coefficient matrix A needs to be solved, which is the most expensive computation in the algorithms. With this in mind, an approximation matrix Q_A , i.e., a preconditioner of the matrix A , has been introduced in the inexact and the parameterized inexact Uzawa methods [11, 14]. Hence, the first iteration step of (1.2) becomes

$$u_{k+1} = u_k + Q_A^{-1}(f - Au_k - Bv_k). \quad (1.3)$$

From both the theoretical analysis and numerical results in Refs. [11, 14], we know that a good approximation matrix Q_A may lead to an efficient inexact Uzawa method. Hence, how to choose a good preconditioner Q_A for matrix A becomes an important problem.

When A is Hermitian positive definite, many efficient preconditioners have been discussed; see Refs. [31, 32] for the details. When A is non-Hermitian positive definite, Yang and Wu [29, 30] employed the Hermitian and skew-Hermitian splitting (HSS) preconditioner of matrix A to accelerate the convergence of Uzawa method, which leads to the following Uzawa–HSS method:

$$\begin{cases} u_{k+1} = u_k + 2\alpha(\alpha I + S)^{-1}(\alpha I + H)^{-1}(f - Au_k - Bv_k), \\ v_{k+1} = v_k + \tau Q^{-1}(B^*u_{k+1} - g), \end{cases} \quad (1.4)$$

where $Q \in \mathbb{R}^{m \times m}$ is an Hermitian positive definite matrix, $H = (1/2)(A + A^*)$ and $S = (1/2)(A - A^*)$ are the Hermitian and skew-Hermitian parts of matrix A , respectively. A similar idea, i.e., using positive definite and skew-Hermitian splitting preconditioner to accelerate the convergence of Uzawa method, can be seen in Ref. [18]. In addition, a modified local HSS (MLHSS) iteration method and the corresponding MLHSS preconditioner proposed by Jiang and Cao [22] are also very efficient for solving the saddle-point problems. However, for these iteration methods referred above, the restrictions on the involved iteration parameters to ensure convergence are complicated. It is hard to verify whether the convergence conditions are satisfied.

In the saddle-point linear system (1.1) or the iteration scheme (1.2), matrix A is non-Hermitian positive definite. For the non-Hermitian positive definite system of linear equations $Az = c$, Bai et al. [12] introduced an efficient shift-splitting iteration method of the form

$$(\alpha I + A)z_{k+1} = (\alpha I - A)z_k + 2c, \quad (1.5)$$

where α is a positive iteration parameter and z_k is the approximation solution of $Az = c$ at the k -th iteration step. This shift-splitting iteration method is unconditionally convergent since its iteration matrix $T(\alpha) = (\alpha I + A)^{-1}(\alpha I - A)$ yields [12]

$$\rho(T(\alpha)) \leq \|(\alpha I + A)^{-1}(\alpha I - A)\| < 1, \quad \forall \alpha > 0. \quad (1.6)$$

Owing to the high efficiency of the shift-splitting iteration method, Cao et al. [16] first employed it to solve the saddle-point problem whose coefficient matrix has Hermitian positive definite $(1, 1)$ -block.

In this work, based on the shift-splitting technique, we propose a preconditioned shift-splitting (PSS) preconditioner for the $(1, 1)$ -block A of the non-Hermitian saddle-point matrix. Combining this PSS preconditioner with the Uzawa iteration method, we establish a new Uzawa-type iteration method, abbreviated as UPSS, to solve the non-Hermitian saddle-point problems. As will be indicated, the new method is effective and its convergent conditions are much easier to be satisfied than the methods in [18, 22, 30]. Moreover, a preconditioner for the saddle-point matrix \mathcal{A} involved in the linear system (1.1) is constructed. Spectral properties of the corresponding preconditioned matrix are also analyzed.

The remainder of this paper is organized as follows. In Section 2, we introduce the UPSS iteration method. In Section 3, the convergence properties of this new iteration method are discussed. In Section 4, we further discuss the spectral properties of the UPSS preconditioned saddle-point matrix. Numerical results are presented in Section 5 to show the effectiveness of the UPSS iteration method and the UPSS preconditioned GMRES method. Finally, in Section 6, we give a brief conclusion of this paper.

2. The UPSS Iteration Method

Let P be an Hermitian positive definite approximation matrix of A , and denote $\bar{A} = P^{-1/2}AP^{-1/2}$, $\bar{z} = P^{1/2}z$ and $\bar{c} = P^{-1/2}c$. Then the linear system $Az = c$ is equivalent to $\bar{A}\bar{z} = \bar{c}$. The shift-splitting iteration scheme for $\bar{A}\bar{z} = \bar{c}$ is

$$\frac{1}{2}(\alpha I + \bar{A})\bar{z}_{k+1} = \frac{1}{2}(\alpha I - \bar{A})\bar{z}_k + \bar{c}. \quad (2.1)$$

Multiplying $P^{1/2}$ for the two sides of (2.1) from left and noticing that $\bar{z} = P^{1/2}z$, the above iteration scheme becomes

$$\frac{1}{2}(\alpha P + A)z_{k+1} = \frac{1}{2}(\alpha P - A)z_k + c.$$

We name this scheme as preconditioned shift-splitting (PSS) iteration scheme. Hence, the matrix

$$M(\alpha) = \frac{1}{2}(\alpha P + A) \quad (2.2)$$

can be viewed as a preconditioner, named as PSS preconditioner, of the non-Hermitian positive definite matrix A .

Now, choosing PSS preconditioner $M(\alpha)$ as Q_A , i.e., $Q_A = M(\alpha)$, in the iteration scheme (1.3), and using the second step of iteration scheme (1.4), we can derive the following UPSS iteration method.

Method 2.1 (The UPSS Method). Given initial vectors $u_0 \in \mathbb{C}^n$, $v_0 \in \mathbb{C}^m$, and two relaxation parameters $\alpha, \tau > 0$. For $k = 0, 1, 2, \dots$, until the iteration sequence converges, compute

$$\begin{cases} u_{k+1} = u_k + 2(\alpha P + A)^{-1}(f - Au_k - Bv_k), \\ v_{k+1} = v_k + \tau Q^{-1}(B^*u_{k+1} - g), \end{cases} \quad (2.3)$$

where $P \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{m \times m}$ are Hermitian positive definite approximation matrices of A and Schur complement matrix $B^*A^{-1}B$, respectively.

The UPSS method can also be induced from the matrix splitting $\mathcal{A} = \mathcal{M}(\alpha, \tau) - \mathcal{N}(\alpha, \tau)$, where

$$\begin{aligned} \mathcal{M}(\alpha, \tau) &= \begin{pmatrix} (1/2)(\alpha P + A) & 0 \\ -B^* & (1/\tau)Q \end{pmatrix}, \\ \mathcal{N}(\alpha, \tau) &= \begin{pmatrix} (1/2)(\alpha P - A) & -B \\ 0 & (1/\tau)Q \end{pmatrix}. \end{aligned} \quad (2.4)$$

This is because, if we denote $x_k = (u_k^*, v_k^*)^*$, the iteration scheme of the form

$$x_{k+1} = \mathcal{T}(\alpha, \tau)x_k + \mathcal{M}(\alpha, \tau)^{-1}b \quad (2.5)$$

is equivalent to (2.3), where the iteration matrix is defined as

$$\begin{aligned} \mathcal{T}(\alpha, \tau) &= \mathcal{M}(\alpha, \tau)^{-1}\mathcal{N}(\alpha, \tau) \\ &= \begin{pmatrix} I - 2(\alpha P + A)^{-1}A & -2(\alpha P + A)^{-1}B \\ \tau Q^{-1}B^*(I - 2(\alpha P + A)^{-1}A) & I - 2\tau Q^{-1}B^*(\alpha P + A)^{-1}B \end{pmatrix}. \end{aligned} \quad (2.6)$$

3. Convergence Results

To analyze the convergence properties of the iteration scheme (2.5), we first introduce several useful lemmas.

Lemma 3.1 ([11, 25]). *Both roots of the complex quadratic equation $\lambda^2 - \phi\lambda + \psi = 0$ have modulus less than one if and only if $|\phi - \bar{\phi}\psi| + |\psi|^2 < 1$, where $\bar{\phi}$ denotes the complex conjugate of ϕ .*

Lemma 3.2. *Let $A \in \mathbb{C}^{n \times n}$ be non-Hermitian positive definite and $B \in \mathbb{C}^{n \times m}$ be of full column rank. If λ is an eigenvalue of the iteration matrix $\mathcal{T}(\alpha, \tau)$, then $\lambda \neq 1$.*

Proof. Let $(x^*, y^*)^*$ be an eigenvector of the iteration matrix $\mathcal{T}(\alpha, \tau)$ corresponding to eigenvalue λ , then we have

$$\begin{pmatrix} \frac{1}{2}(\alpha P - A) & -B \\ 0 & \frac{1}{\tau}Q \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} \frac{1}{2}(\alpha P + A) & 0 \\ -B^* & \frac{1}{\tau}Q \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

or equivalently,

$$\begin{cases} (\alpha P - A)x - 2By = \lambda(\alpha P + A)x, \\ Qy = \lambda(-\tau B^*x + Qy). \end{cases} \quad (3.1)$$

If $\lambda = 1$, we have $Ax + By = 0$ and $B^*x = 0$, which indicates $(x^*, y^*)^* = 0$ by noticing that matrix A is nonsingular and matrix B is of full column rank. This is in contradiction with the fact that $(x^*, y^*)^* \in \mathbb{C}^{n+m}$ is an eigenvector. \square

Lemma 3.3. *Let $A \in \mathbb{C}^{n \times n}$ be non-Hermitian positive definite and $B \in \mathbb{C}^{n \times m}$ be of full column rank. If λ is an eigenvalue of the iteration matrix $\mathcal{T}(\alpha, \tau)$, and $(x^*, y^*)^* \in \mathbb{C}^{n+m}$ is the corresponding eigenvector, then $x \neq 0$. Moreover, when $y = 0$, we have $|\lambda| < 1$.*

Proof. If $x = 0$, using (3.1) gives $y = 0$, which contradicts with the fact that $(x^*, y^*)^* \in \mathbb{C}^{n+m}$ is an eigenvector.

If $y = 0$, from the first equality of (3.1), we obtain $(\alpha P + A)^{-1}(\alpha P - A)x = \lambda x$. So λ is also an eigenvalue of the matrix $(\alpha P + A)^{-1}(\alpha P - A)$. Now we only need to discuss the eigenvalues of the matrix

$$P^{\frac{1}{2}}(\alpha P + A)^{-1}P^{\frac{1}{2}}P^{-\frac{1}{2}}(\alpha P - A)P^{-\frac{1}{2}} = (\alpha I + P^{-\frac{1}{2}}AP^{-\frac{1}{2}})^{-1}(\alpha I - P^{-\frac{1}{2}}AP^{-\frac{1}{2}}),$$

which is similar to $(\alpha P + A)^{-1}(\alpha P - A)$. Note that the matrix $P^{-1/2}AP^{-1/2}$ is non-Hermitian positive definite, so $|\lambda| < 1$ by using (1.6). \square

Using the above conclusions, we have the following result that characterizes the property of the eigenpairs of the iteration matrix $\mathcal{T}(\alpha, \tau)$.

Lemma 3.4. *Let $A \in \mathbb{C}^{n \times n}$ be non-Hermitian positive definite and $B \in \mathbb{C}^{n \times m}$ be of full column rank. Matrices $P \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{m \times m}$ are Hermitian positive definite approximations of A and the Schur complement matrix $B^*A^{-1}B$, respectively. Assume that λ is an eigenvalue of iteration matrix $\mathcal{T}(\alpha, \tau)$ and $(x^*, y^*)^* \in \mathbb{C}^{n+m}$ is the corresponding eigenvector. Denote by*

$$\mu_1 + \iota\mu_2 := \mu_1(x) + \iota\mu_2(x) = \frac{x^*Ax}{x^*Px}, \quad \eta := \eta(x) = \frac{x^*BQ^{-1}B^*x}{x^*Px}, \quad (3.2)$$

where μ_1, μ_2 and η are real-value functions of the complex vector $x \in \mathbb{C}^n$. Then λ satisfies the following equation

$$\lambda^2 - \frac{2(\alpha - \tau\eta)}{\alpha + \mu_1 + \iota\mu_2}\lambda + \frac{\alpha - \mu_1 - \iota\mu_2}{\alpha + \mu_1 + \iota\mu_2} = 0. \quad (3.3)$$

Proof. Using the second equality of (3.1) and noticing the conclusion $\lambda \neq 1$ in Lemma 3.2, we have

$$y = \frac{\lambda\tau}{\lambda - 1}Q^{-1}B^*x. \quad (3.4)$$

Substituting (3.4) into the first equality of (3.1) derives

$$(\alpha P - A)x - 2\frac{\lambda\tau}{\lambda - 1}BQ^{-1}B^*x = \lambda(\alpha P + A)x,$$

or equivalently,

$$(1 - \lambda)\alpha Px - (1 + \lambda)Ax = 2\frac{\lambda\tau}{\lambda - 1}BQ^{-1}B^*x. \tag{3.5}$$

Owing to $x \neq 0$ from Lemma 3.3, multiplying $x^*/(x^*Px)$ to the two sides of (3.5) from left gives

$$(1 - \lambda)\alpha - (1 + \lambda)\frac{x^*Ax}{x^*Px} = \frac{2\lambda\tau}{\lambda - 1}\frac{x^*BQ^{-1}B^*x}{x^*Px}. \tag{3.6}$$

Using the definitions (3.2), the equation (3.6) can be written as

$$(1 - \lambda)\alpha - (1 + \lambda)(\mu_1 + \mu_2i) = \frac{2\lambda\tau}{\lambda - 1}\eta.$$

Simple calculation gives the quadratic equation (3.3). □

Remark 3.1. For a particular case of equation (3.4), we know that if $x \in \text{null}(B^*)$, i.e., x is a vector in the null space of B^* , then $y = 0$. Thus the eigenvalues of the iteration matrix $\mathcal{T}(\alpha, \tau)$ satisfy

$$(1 - \lambda)\alpha - (1 + \lambda)(\mu_1 + \mu_2i) = 0.$$

By simple calculations, the above equation can be proved to be a special form of the equation (3.3) since $\eta = 0$ and $\lambda \neq 1$. Therefore, in the proof of the Lemma 3.4, we did not discuss this special case.

Using the above Lemmas, we can derive a sufficient condition to guarantee the convergence of iteration scheme (2.5).

Theorem 3.1. *Let $A \in \mathbb{C}^{n \times n}$ be non-Hermitian positive definite and $B \in \mathbb{C}^{n \times m}$ be of full column rank. Matrices $P \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{m \times m}$ are Hermitian positive definite approximations of A and the Schur complement matrix $B^*A^{-1}B$, respectively. Then the UPSS method is convergent for any $\alpha > 0$ if parameter τ satisfies*

$$0 < \tau < \frac{2\alpha}{\lambda_{\max}(Q^{-1}B^*P^{-1}B)}. \tag{3.7}$$

Proof. Note that the iteration scheme (2.5) is convergent if and only if all the eigenvalues of the iteration matrix $\mathcal{T}(\alpha, \tau)$ satisfy $|\lambda| < 1$. For the case $y = 0$, we have $|\lambda| < 1$ using Lemma 3.3. In the following, we only consider the case $y \neq 0$, i.e., $x \notin \text{null}(B^*)$. Denote the quadratic equation in (3.3) by $\lambda^2 - \phi\lambda + \psi = 0$, where

$$\phi = \frac{2(\alpha - \tau\eta)}{\alpha + \mu_1 + \mu_2i}, \quad \text{and} \quad \psi = \frac{\alpha - \mu_1 - \mu_2i}{\alpha + \mu_1 + \mu_2i}.$$

By using Lemma 3.1, we need to solve the following equivalent inequality

$$\begin{aligned} |\phi - \bar{\phi}\psi| + |\psi|^2 &= \left| \frac{2(\alpha - \tau\eta)}{\alpha + \mu_1 + i\mu_2} - \frac{2(\alpha - \tau\eta)}{\alpha + \mu_1 - i\mu_2} \cdot \frac{\alpha - \mu_1 - i\mu_2}{\alpha + \mu_1 + i\mu_2} \right| + \frac{(\alpha - \mu_1)^2 + \mu_2^2}{(\alpha + \mu_1)^2 + \mu_2^2} \\ &= \frac{|4(\alpha - \tau\eta)\mu_1| + (\alpha - \mu_1)^2 + \mu_2^2}{(\alpha + \mu_1)^2 + \mu_2^2} \\ &< 1, \end{aligned}$$

which holds if and only if

$$|(\alpha - \tau\eta)\mu_1| < \alpha\mu_1. \quad (3.8)$$

Since matrix A is positive definite, matrices P and Q are Hermitian positive definite and $x \notin \text{null}(B^*)$, we know that $\mu_1 > 0$, $\eta > 0$. Then the inequality (3.8) holds for any $\alpha > 0$ if parameter τ yields

$$0 < \tau < \frac{2\alpha}{\eta}. \quad (3.9)$$

Let $z = P^{1/2}x$, the function η defined in (3.2) yields

$$\begin{aligned} \eta &\leq \max_{z \in \mathbb{C}^n} \frac{z^* P^{-\frac{1}{2}} B Q^{-1} B^* P^{-\frac{1}{2}} z}{z^* z} \\ &= \lambda_{\max} \left((P^{-\frac{1}{2}} B Q^{-1} B^* P^{-\frac{1}{2}}) \right) \\ &= \lambda_{\max} \left((Q^{-\frac{1}{2}} B^* P^{-\frac{1}{2}}) (P^{-\frac{1}{2}} B Q^{-\frac{1}{2}}) \right) \\ &= \lambda_{\max} (Q^{-1} B^* P^{-1} B). \end{aligned} \quad (3.10)$$

In the above equality, we have used the conclusion that the nonzero eigenvalues of matrices EE^* and E^*E are the same for any square matrix E . Finally, we complete the proof by combining (3.10) with (3.9). \square

Using Theorem 3.1, the following convergence result for a special case of the UPSS iteration method can be easily obtained.

Corollary 3.1. *Let $A \in \mathbb{C}^{n \times n}$ be non-Hermitian positive definite and $B \in \mathbb{C}^{n \times m}$ be of full column rank. Matrices $P \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{m \times m}$ are Hermitian positive definite approximations of A and the Schur complement matrix $B^*A^{-1}B$, respectively. When $\alpha = \tau$, the UPSS iteration scheme (2.3) is convergent if*

$$\lambda_{\max}(Q^{-1}B^*P^{-1}B) < 2. \quad (3.11)$$

From the above corollary, we see that the convergence condition (3.11) is simple and easy to be satisfied. Particularly, when we choose $Q = B^*P^{-1}B$ and $\alpha = \tau$, the UPSS iteration method is convergent.

4. Preconditioning Properties

Since the UPSS method can be induced from the matrix splitting $\mathcal{A} = \mathcal{M}(\alpha, \tau) - \mathcal{N}(\alpha, \tau)$, the matrix $\mathcal{M}(\alpha, \tau)$ defined in (2.4) can be viewed as a preconditioner for the coefficient matrix \mathcal{A} . We name it as UPSS preconditioner. Hence, the UPSS preconditioner can be employed to accelerate the convergence rate of the Krylov subspace iteration methods used for solving saddle-point problems (1.1).

The following spectral properties of the preconditioned matrix $\mathcal{M}(\alpha, \tau)^{-1}\mathcal{A}$ partially reflect the preconditioning effect of the preconditioner $\mathcal{M}(\alpha, \tau)$.

Theorem 4.1. *Let A be non-Hermitian positive definite and B be of full column rank. Matrices $P \in \mathbb{C}^{n \times n}$ and $Q \in \mathbb{C}^{m \times m}$ are Hermitian positive definite approximations of A and the Schur complement $B^*A^{-1}B$, respectively. When the parameters $\alpha, \tau > 0$ close to zero, the eigenvalues of the preconditioned matrix $\mathcal{M}(\alpha, \tau)^{-1}\mathcal{A}$ cluster near the points $(0, 0)$ and $(2, 0)$.*

Proof. Let μ be an eigenvalue of the preconditioned matrix $\mathcal{M}(\alpha, \tau)^{-1}\mathcal{A}$. Then $\lambda = 1 - \mu$ must be an eigenvalue of the iteration matrix $\mathcal{T}(\alpha, \tau)$.

From (3.3), we have

$$\lambda_{\pm} = \frac{\alpha - \tau\eta \pm \sqrt{(\alpha - \tau\eta)^2 - (\alpha + \mu_1 + i\mu_2)(\alpha - \mu_1 - i\mu_2)}}{\alpha + \mu_1 + i\mu_2}, \quad (4.1)$$

which gives

$$\mu_{\pm} = 1 - \lambda_{\pm} = \frac{\tau\eta + \mu_1 + i\mu_2 \pm \sqrt{(\alpha - \tau\eta)^2 - (\alpha + \mu_1 + i\mu_2)(\alpha - \mu_1 - i\mu_2)}}{\alpha + \mu_1 + i\mu_2}. \quad (4.2)$$

Simple calculation gives

$$\lim_{\alpha \rightarrow 0, \tau \rightarrow 0} \mu_+ = 2, \quad \lim_{\alpha \rightarrow 0, \tau \rightarrow 0} \mu_- = 0.$$

Hence, the proof of this theorem is completed. \square

Remark 4.1. As is known, if the preconditioned matrix has only a few distinct eigenvalue clusters and the eigenvalues are away from the origin, the degree of the minimal polynomial of the preconditioned matrix is low. This makes the preconditioned Krylov subspace iteration method converges quickly, as the iteration steps of the preconditioned Krylov subspace methods will not exceed the degree of the minimal polynomial of the preconditioned matrix [2, 26]. Under this consideration, the smallest parameters are not the optimal ones although we can derive two separated eigenvalue clusters, because the leftmost eigenvalues are too close to zero when the parameters $\alpha, \tau > 0$ are very small. This can be seen from the numerical results presented in Section 5.

5. Numerical Experiments

In this section, we use an example to test the feasibility and efficiency of the UPSS iteration method proposed in this work. Moreover, the preconditioning effect of the UPSS preconditioner $\mathcal{M}(\alpha, \tau)$ used for accelerating the convergence rate of GMRES method is also tested.

Consider the non-Hermitian saddle-point matrix \mathcal{A} defined in (1.1) whose sub-blocks have the form of

$$A = \begin{pmatrix} I_l \otimes T_x + T_y \otimes I_l & 0 \\ 0 & I_l \otimes T_x + T_y \otimes I_l \end{pmatrix} \in \mathbb{R}^{2l^2 \times 2l^2}$$

and

$$B = \begin{pmatrix} I_l \otimes F \\ F \otimes I_l \end{pmatrix} \in \mathbb{R}^{2l^2 \times l^2},$$

where

$$T_x = T_y = \frac{1}{h^2} \text{tridiag}(-1 - r, 2, -1 + r) \in \mathbb{R}^{l \times l}, \quad F = \frac{1}{h} \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{l \times l}.$$

Here, \otimes denotes the Kronecker product and $r = qh/2$ is the mesh Reynolds number. The difference between this saddle-point matrix and the matrix described in Example 5.1 of Ref. [10] is that the sub-matrix A in this example is nonsymmetric. Here, T_x and T_y are obtained when the five-point centered finite difference discretization is applied to the two-dimensional convection-diffusion equation

$$-(u_{xx} + u_{yy}) + q(u_x + u_y) = f(x, y)$$

on the unit square $\Omega = [0, 1] \times [0, 1]$, with Dirichlet boundary condition and constant coefficient q . For the discretization, an equidistant step-size $h = 1/(l + 1)$ in each coordinate direction is used and the natural ordering was employed to the unknowns, which is analogue to the three-dimensional case presented in [8]. In this example, we have $n = 2l^2$ and $m = l^2$, so the total number of unknowns is $n + m = 3l^2$. In actual computations, we choose the right-hand-side vector $b \in \mathbb{R}^{n+m}$ such that its exact solution of linear system is $x_* = (1, 1, \dots, 1) \in \mathbb{R}^{n+m}$.

We compare the numerical results of the UPSS iteration scheme with those of the MLHSS [22], Uzawa–HSS [30] and Uzawa–PSS [18] methods. We should note that the PSS appeared in Uzawa–PSS is the abbreviation of *positive definite and skew-Hermitian splitting*, which is different with the abbreviation PSS appeared in the UPSS method. The iteration scheme of the MLHSS method is

$$\begin{cases} u_{k+1} = u_k + (Q_1 + H)^{-1}(f - Au_k - Bv_k), \\ v_{k+1} = v_k + Q_2^{-1}(B^*u_{k+1} - g), \end{cases} \tag{5.1}$$

where $H = (1/2)(A + A^*)$ is the Hermitian part of matrix A , $Q_1 \in \mathbb{R}^{n \times n}$ and $Q_2 \in \mathbb{R}^{m \times m}$ are Hermitian positive definite matrices. Now, splitting matrix A into its positive definite and skew-Hermitian parts as $A = A_p + A_s$, then the iteration scheme of the Uzawa–PSS method can be defined as

$$\begin{cases} u_{k+1} = u_k + 2\alpha(\alpha I + A_s)^{-1}(\alpha I + A_p)^{-1}(f - Au_k - Bv_k), \\ v_{k+1} = v_k + \tau Q^{-1}(B^*u_{k+1} - g), \end{cases} \tag{5.2}$$

where $A_p = D_H + 2L_H$, $A_s = L_H^* - L_H + S$, with D_H and L_H being the diagonal part and strictly lower triangular part of H .

In the implementation, we choose $Q_1 = \alpha I$ and $Q_2 = (1/\tau)Q$ in the MLHSS method. For the preconditioning matrices P and Q of the tested methods, we choose $P = H$ and $Q = B^*P^{-1}B$ or $Q = \text{diag}(B^*D^{-1}B)$, where $D = \text{diag}(A)$. In addition, all the involved

sub-linear systems are solved by Cholesky or LU factorization in combination with an approximate minimum degree (AMD) reordering, which can make a big reduction of CPU time. Particularly, when solving the sub-linear system with coefficient matrix $Q = B^*P^{-1}B$, we use the conjugate gradient (CG) method combined with the AMD reordering. The inner iteration is terminated when the relative residual satisfies

$$\text{res} = \frac{\|r_k\|}{\|r_0\|} < \text{intol} = 10^{-3}.$$

The detailed implementation of the CG method for $B^*P^{-1}Bx = b$ can be described as follows:

Pseudocode of CG(P, B, b, intol) for the linear system $B^*P^{-1}Bx = b$

1. compute $r_0 = b$, $p_0 = r_0$. Set $x = 0$.
 2. For $j = 0, 1, \dots$, until convergence, i.e., $\text{res} < \text{intol}$, Do:
 3. $r_1 = Bp_j$
 4. solve $Pq_1 = r_1$ by using Cholesky factorization with AMD reordering
 5. $q_j = B^T q_1$
 6. $\alpha_j = (r_j, r_j)/(q_j, p_j)$
 7. $x_{j+1} = x_j + \alpha_j p_j$
 8. $r_{j+1} = r_j - \alpha_j q_j$
 9. $\text{res} = \|r_{j+1}\|/\|r_0\|$
 10. $\beta_j = (r_{j+1}, r_{j+1})/(r_j, r_j)$
 11. $p_{j+1} = r_{j+1} + \beta_j p_j$
 12. EndDo
-

All the tested iteration methods and preconditioned GMRES methods are started from zero vector and terminated once the current iteration solution x_k satisfies

$$\text{RES} = \frac{\|b - \mathcal{A}x_k\|}{\|b\|} < 10^{-6}, \quad (5.3)$$

or the iteration steps exceed $k_{\max} = 1500$. In addition, all the computations are implemented in MATLAB [version 7.14.0.739 (R2012a)] in double precision on a personal computer with 3.20GHZ central processing unit [Intel(R) Core(TM) i5-3470] and 8.00GB memory.

Using experimentally found optimal parameters (denoted as α_{exp} and τ_{exp}) in the interval $(0, 2000]$, we present the numerical results including numbers of iteration count (denoted as IT), elapsed CPU times in seconds (denoted as CPU) and relative residuals (denoted as RES) of the iteration methods and the preconditioned GMRES methods.

In Tables 1–4, we list the numerical results of the four tested iteration methods with two choices of the preconditioning matrix Q . The sign “–” in Tables 3 and 4 is used to denote that the methods do not converge within $k_{\max} = 1500$ iteration steps. When we choose $P = H$ and appropriate parameter α , the preconditioner $(1/2)(\alpha H + A)$ seems to be more close to

Table 1: Numerical results of the iteration methods with $Q = B^*P^{-1}B$ for the case $q = 1$.

	Method	α_{exp}	τ_{exp}	IT	CPU	RES
$l = 16$	UPSS	0.89	0.89	10	0.0156	9.3177e-07
	MLHSS	0.11	0.79	13	0.0312	8.5462e-07
	Uzawa-HSS	188	0.10	122	0.5616	9.9853e-07
	Uzawa-PSS	184	0.10	180	0.2808	9.9785e-07
$l = 32$	UPSS	0.95	0.95	8	0.0468	9.9725e-07
	MLHSS	0.01	0.81	12	0.0468	9.7715e-07
	Uzawa-HSS	234	0.04	272	1.3884	9.9938e-07
	Uzawa-PSS	421	0.07	335	2.6676	9.9678e-07
$l = 64$	UPSS	0.94	0.94	8	0.2496	8.5329e-07
	MLHSS	0.05	0.74	12	0.3744	9.9332e-07
	Uzawa-HSS	309	0.02	708	21.5281	9.7968e-07
	Uzawa-PSS	1841	0.06	872	52.5723	9.9244e-07

Table 2: Numerical results of the iteration methods with $Q = B^*P^{-1}B$ for the case $q = 10$.

	Method	α_{exp}	τ_{exp}	IT	CPU	RES
$l = 16$	UPSS	1.51	1.42	35	0.0468	9.8463e-07
	MLHSS	63.79	0.66	93	0.1248	9.9983e-07
	Uzawa-HSS	299	0.38	181	0.2652	9.7256e-07
	Uzawa-PSS	185	0.23	255	0.4056	9.8738e-07
$l = 32$	UPSS	0.96	0.91	39	0.1872	8.2485e-07
	MLHSS	63.89	0.61	90	0.4212	9.9954e-07
	Uzawa-HSS	450	0.15	324	1.6848	9.9968e-07
	Uzawa-PSS	447	0.12	494	3.9624	9.9879e-07
$l = 64$	UPSS	1.51	1.51	30	0.9048	9.9337e-07
	MLHSS	63.56	0.59	84	2.5116	9.9958e-07
	Uzawa-HSS	303	0.02	1244	58.6563	9.9468e-07
	Uzawa-PSS	1501	0.10	867	54.1947	9.9742e-07

the matrix A than the other three preconditioners, i.e., $\alpha I + H$, $(1/2\alpha)(\alpha I + H)(\alpha I + S)$ and $(1/2\alpha)(\alpha I + A_p)(\alpha I + A_S)$. Now, solving the inner linear system by using Cholesky or LU factorization in combination with AMD reordering, the workload in each step of the UPSS method is almost the same as the MLHSS method and much more less than the Uzawa-HSS and the Uzawa-PSS method. Hence, we may expect that the IT and the CPU of the UPSS iteration method are less than those of the other three methods. Actually, from any of the four tables, we can see that the UPSS method is really the most efficient one, which always costs the least IT and CPU for different matrix Q and parameter q . Moreover, from the numerical results, we can also see that the IT of the UPSS iteration method is not sensitive

Table 3: Numerical results of the iteration methods with $Q = \text{diag}(B^*D^{-1}B)$ for the case $q = 1$.

	Method	α_{exp}	τ_{exp}	IT	CPU	RES
$l = 16$	UPSS	3.01	1.89	42	0.0156	9.9684e-07
	MLHSS	0.34	1.38	87	0.0312	9.9994e-07
	Uzawa-HSS	688	0.53	169	0.0312	9.8404e-07
	Uzawa-PSS	467	0.68	131	0.0156	9.9288e-07
$l = 32$	UPSS	3.53	2.91	50	0.0156	9.3390e-07
	MLHSS	157	1.52	126	0.0624	9.9306e-07
	Uzawa-HSS	845	0.21	878	0.5616	9.9933e-07
	Uzawa-PSS	1089	0.44	429	0.1872	9.9472e-07
$l = 64$	UPSS	4.17	4.59	60	0.1092	9.1818e-07
	MLHSS	154	1.48	251	0.7644	9.9941e-07
	Uzawa-HSS	-	-	-	-	-
	Uzawa-PSS	-	-	-	-	-

Table 4: Numerical results of the iteration methods with $Q = \text{diag}(B^*D^{-1}B)$ for the case $q = 10$.

	Method	α_{exp}	τ_{exp}	IT	CPU	RES
$l = 16$	UPSS	2.91	1.84	48	0.0156	9.8048e-07
	MLHSS	114	1.30	61	0.0312	9.9134e-07
	Uzawa-HSS	659	0.54	169	0.0312	9.9951e-07
	Uzawa-PSS	468	0.73	133	0.0156	9.6018e-07
$l = 32$	UPSS	3.69	2.77	54	0.0312	9.2546e-07
	MLHSS	155	1.54	126	0.0624	9.9835e-07
	Uzawa-HSS	790	0.31	614	0.3588	9.9991e-07
	Uzawa-PSS	1089	0.46	413	0.1716	9.9322e-07
$l = 64$	UPSS	4.21	4.53	64	0.1248	9.4761e-07
	MLHSS	151	1.49	252	0.6396	9.9842e-07
	Uzawa-HSS	-	-	-	-	-
	Uzawa-PSS	-	-	-	-	-

to the size of the coefficient matrices.

Comparing the numerical results of Tables 1 and 2 with those of Tables 3 and 4, we see that the ITs of the four methods for the case $Q = B^*P^{-1}B$ are obviously less than the case $Q = \text{diag}(B^*D^{-1}B)$. However, the costed CPU times are reversed. The reason is that $B^*P^{-1}B$ is a better approximation of the Schur complement matrix $B^*A^{-1}B$, but as the coefficient matrices of an inner linear system, it is more difficult to be solved than the matrix $\text{diag}(B^*D^{-1}B)$.

To test the preconditioning effect of the UPSS preconditioner, the numerical results of the UPSS preconditioned GMRES method, the MLHSS preconditioned GMRES method and

Table 5: Numerical results of the preconditioned GMRES methods with $Q = B^*P^{-1}B$.

		Preconditioner	α_{exp}	τ_{exp}	IT	CPU	RES
$q = 1$	$l = 16$	UPSS	0.55	1.10	7	0.0199	9.2289e-07
		MLHSS	0.19	0.37	10	0.0252	9.8162e-07
		Null	–	–	120	0.1214	9.8771e-07
	$l = 32$	UPSS	0.47	0.94	7	0.0704	9.4602e-07
		MLHSS	0.15	0.31	10	0.0876	9.8503e-07
		Null	–	–	264	2.1791	9.7325e-07
	$l = 64$	UPSS	0.37	0.74	7	0.3557	9.6021e-07
		MLHSS	0.01	0.91	9	0.4544	9.6198e-07
		Null	–	–	572	20.6641	9.9188e-07
$q = 10$	$l = 16$	UPSS	0.51	1.02	16	0.0370	8.7313e-07
		MLHSS	0.01	0.90	28	0.0638	9.6553e-07
		Null	–	–	197	0.3140	9.3602e-07
	$l = 32$	UPSS	0.52	1.04	16	0.1347	9.1791e-07
		MLHSS	0.19	0.90	28	0.2261	9.9927e-07
		Null	–	–	402	4.8989	9.1149e-07
	$l = 64$	UPSS	0.48	0.96	16	0.6861	9.2087e-07
		MLHSS	0.05	0.84	28	1.1914	9.9993e-07
		Null	–	–	804	39.8965	9.8169e-07

the GMRES method without preconditioning are compared in Tables 5 and 6. Here, the MLHSS preconditioner is of the form

$$\mathcal{P}_{\text{MLHSS}} = \begin{pmatrix} \alpha I + H & 0 \\ -B^* & \frac{1}{\tau} Q \end{pmatrix}.$$

In the tables, we denote the GMRES method without preconditioning by “Null”. From the results in Tables 5 and 6, we can observe that the UPSS preconditioned GMRES method outperforms both the GMRES and the MLHSS preconditioned GMRES methods for different choices of the preconditioning matrices Q . This is because when we choose $P = H$ and appropriate parameter α , the UPSS preconditioner $\mathcal{M}(\alpha, \tau)$ is more close to the coefficient matrix \mathcal{A} than the MLHSS preconditioner $\mathcal{P}_{\text{MLHSS}}$, but the computations of the two methods in each step are almost the same. Moreover, we can see that both the UPSS preconditioned and MLHSS preconditioned GMRES methods are not sensitive to the size of the saddle-point matrices, i.e., the two methods cost nearly the same ITs for different problem sizes.

In order to verify the theoretical results in Theorem 4.1, we figure out the eigenvalue distributions of the UPSS preconditioned matrices with $Q = \text{diag}(B^*D^{-1}B)$ and $\alpha = \tau$ in Fig. 1. We choose the parameters $\alpha = \tau = 1, 0.1, 0.01$ and 0.001 , respectively. In this figure, we see that, with the decreasing of iteration parameters α and τ , the eigenvalues of the UPSS preconditioned matrices cluster gradually to the points $(0, 0)$ and $(2, 0)$, which is

Table 6: Numerical results of the preconditioned GMRES methods with $Q = \text{diag}(B^*D^{-1}B)$.

		Preconditioner	α_{exp}	τ_{exp}	IT	CPU	RES
$q = 1$	$l = 16$	UPSS	1.00	0.75	12	0.0105	9.5648e-07
		MLHSS	0.32	0.68	15	0.0122	9.9929e-07
		Null	–	–	120	0.1214	9.8771e-07
	$l = 32$	UPSS	0.99	0.61	15	0.0374	9.9673e-07
		MLHSS	0.36	0.63	17	0.0412	9.9828e-07
		Null	–	–	264	2.1791	9.7325e-07
	$l = 64$	UPSS	1.01	0.77	15	0.1103	9.2157e-07
		MLHSS	0.06	0.62	17	0.1234	9.1759e-07
		Null	–	–	572	20.6641	9.9188e-07
$q = 10$	$l = 16$	UPSS	0.95	0.45	32	0.0228	9.9935e-07
		MLHSS	0.06	0.63	40	0.0303	9.6419e-07
		Null	–	–	197	0.3140	9.3602e-07
	$l = 32$	UPSS	0.96	0.46	34	0.0795	9.9500e-07
		MLHSS	0.08	0.62	43	0.1121	9.8589e-07
		Null	–	–	402	4.8989	9.1149e-07
	$l = 64$	UPSS	0.97	0.48	36	0.2241	9.9467e-07
		MLHSS	0.13	0.64	47	0.3432	9.9875e-07
		Null	–	–	804	39.8965	9.8169e-07

in accordance with the theoretical results in Theorem 4.1.

6. Conclusion

In this work, we proposed a UPSS iteration scheme to solve the non-Hermitian saddle-point problems (1.1). Compared with some similar Uzawa-type iteration methods, this new method has more simple convergence conditions, which are easy to be satisfied. In addition, using this method as a preconditioner to accelerate the convergence rate of the Krylov subspace method, we analyzed the spectral properties of the UPSS preconditioned saddle-point matrix. Numerical results were presented to verify the feasibility and the efficiency of the new iteration scheme and the corresponding preconditioner.

However, the efficiencies of the UPSS and many other existing iteration methods depend on the choices of the iteration parameters. How to find easy calculated parameters is a tough work for the implementation of the algorithm and should be further studied in the future.

Acknowledgments

The authors are very much indebted to the referees for their constructive and valuable comments and suggestions which greatly improved the original manuscript of this paper.

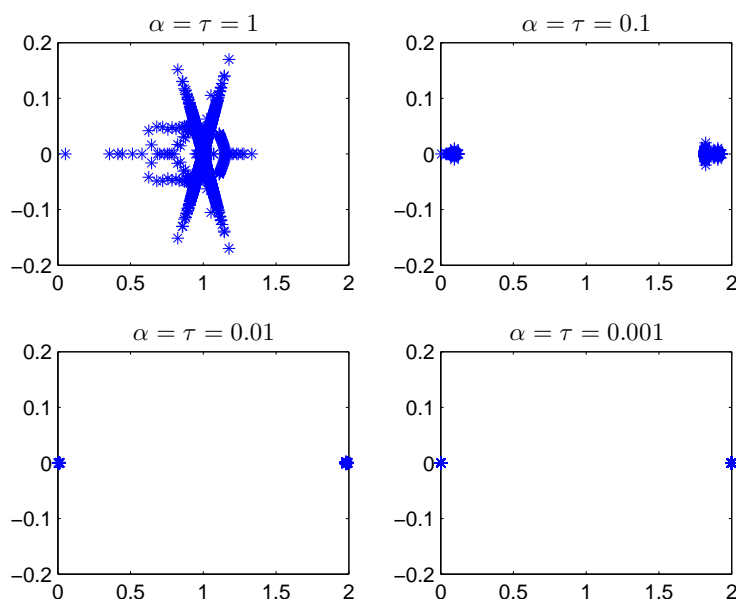


Figure 1: Eigenvalue distributions of the UPSS preconditioned matrices with $Q = \text{diag}(B^*D^{-1}B)$, $q = 1$ and $l = 32$.

This work is partially supported by the National Natural Science Foundation of China (Grant Nos. 11471150, 11401281, 11301290) and the Fundamental Research Fund for the Central Universities (Grant No. lzujbky-2016-97).

References

- [1] K. J. ARROW, L. HURWICZ, AND H. UZAWA, *Studies in Linear and Nonlinear Programming*, Stanford University Press, Stanford, 1958.
- [2] Z.-Z. BAI, *Structured preconditioners for nonsingular matrices of block two-by-two structures*, Math. Comput., 75 (2006), pp. 791–815.
- [3] Z.-Z. BAI, *Optimal parameters in the HSS-like methods for saddle-point problems*, Numer. Linear Algebra Appl., 16 (2009), pp. 447–479.
- [4] Z.-Z. BAI, *Block preconditioners for elliptic PDE-constrained optimization problems*, Computing, 91 (2011), pp. 379–395.
- [5] Z.-Z. BAI, *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] Z.-Z. BAI, AND G. H. GOLUB, *Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems*, IMA J. Numer. Anal., 27 (2007), pp. 1–23.
- [7] 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.
- [8] Z.-Z. BAI, AND A. HADJIDIMOS, *Optimization of extrapolated Cayley transform with non-Hermitian positive definite matrix*, Linear Algebra Appl., 463 (2014), pp. 322–339.

- [9] Z.-Z. BAI, M. K. NG, AND Z.-Q. WANG, *Constraint preconditioners for symmetric indefinite matrices*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 410–433.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
- [14] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASSILEV, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.
- [15] F. BREZZI, AND M. FORTIN, *Mixed and hybrid finite elements methods*, Springer Series in Computational Mathematics, Vol. 15, Springer-Verlag, New York, 2012.
- [16] Y. CAO, J. DU, AND Q. NIU, *Shift-splitting preconditioners for saddle point problems*, J. Comput. Appl. Math., 272 (2014), pp. 239–250.
- [17] Y. CAO, L.-Q. YAO, M.-Q. JIANG, AND Q. NIU, *A relaxed HSS preconditioner for saddle point problems from meshfree discretization*, J. Comput. Math., 31 (2013), pp. 398–421.
- [18] Y. CAO, AND S.-C. YI, *A class of Uzawa–PSS iteration methods for nonsingular and singular non-Hermitian saddle point problems*, Appl. Math. Comput., 275 (2016), pp. 41–49.
- [19] 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.
- [20] B. FISCHER, A. RAMAGE, D. J. SILVESTER, AND A. J. WATHEN, *Minimum residual methods for augmented systems*, BIT Numer. Math., 38 (1998), pp. 527–543.
- [21] G. H. GOLUB, AND A. J. WATHEN, *An iteration for indefinite systems and its application to the Navier-Stokes equations*, SIAM J. Sci. Comput., 19 (1998), pp. 530–539.
- [22] M.-Q. JIANG, AND Y. CAO, *On local Hermitian and skew-Hermitian splitting iteration methods for generalized saddle point problems*, J. Comput. Appl. Math., 231 (2009), pp. 973–982.
- [23] C. KELLER, N. I. M. GOULD, AND A. J. WATHEN, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.
- [24] K. H. LEEM, S. P. OLIVEIRA, AND D. E. STEWART, *Algebraic multigrid (AMG) for saddle point systems from meshfree discretizations*, Numer. Linear Algebra Appl., 11 (2004), pp. 293–308.
- [25] J. J. H. MILLER, *On the location of zeros of certain classes of polynomials with applications to numerical analysis*, J. Inst. Math. Appl., 8 (1971), pp. 397–406.
- [26] Y. SAAD, *Iterative Methods for Sparse Linear Systems, Second Edition*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.
- [27] C. H. SANTOS, B. P. B. SILVA, AND J.-Y. YUAN, *Block SOR methods for rank-deficient least-squares problems*, J. Comput. Appl. Math., 100 (1998), pp. 1–9.
- [28] X. WU, B. P. B. SILVA, AND J.-Y. YUAN, *Conjugate gradient method for rank deficient saddle point problems*, Numer. Algor., 35 (2004), pp. 139–154.
- [29] 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.
- [30] A.-L. YANG, AND Y.-J. WU, *The Uzawa–HSS method for saddle-point problems*, Appl. Math. Lett., 38 (2014), pp. 38–42.
- [31] J.-H. YUN, *Variants of the Uzawa method for saddle point problems*, Comput. Math. Appl., 65 (2013), pp. 1037–1046.
- [32] J.-J. ZHANG, AND J.-J. SHANG, *A class of Uzawa–SOR methods for saddle point problems*, Appl. Math. Comput., 216 (2010), pp. 2163–2168.