A Fast Shift-Splitting Iteration Method for Nonsymmetric Saddle Point Problems

Quan-Yu Dou, Jun-Feng Yin* and Ze-Yu Liao

School of Mathematical Sciences, Tongji University, Shanghai 200092, P.R. China.

Received 17 June 2016; Accepted (in revised version) 31 December 2016.

Abstract. Based on the shift-splitting technique and the idea of Hermitian and skew-Hermitian splitting, a fast shift-splitting iteration method is proposed for solving nonsingular and singular nonsymmetric saddle point problems in this paper. Convergence and semi-convergence of the proposed iteration method for nonsingular and singular cases are carefully studied, respectively. Numerical experiments are implemented to demonstrate the feasibility and effectiveness of the proposed method.

AMS subject classifications: 65M10, 78A48

Key words: Nonsymmetric saddle point problems, fast shift-splitting, convergence, semi-convergence.

1. Introduction

Consider the following nonsymmetric saddle point problems

\[
\begin{bmatrix}
A & B^T \\
-B & 0
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix} = \begin{bmatrix}
f \\
g
\end{bmatrix},
\]

(1.1)

where \(A \in \mathbb{R}^{n \times n}\) is a nonsymmetric positive definite matrix, \(B \in \mathbb{R}^{m \times n}\) is a rectangular matrix with \(m \leq n\), \(f \in \mathbb{R}^n\) and \(g \in \mathbb{R}^m\) are given vectors.

The saddle point problems (1.1) arise in a variety of scientific and engineering applications, such as computational fluid dynamics [13], mixed finite element approximation of elliptic partial differential equations [20] and Lagrange-type methods for constrained nonconvex optimization problems [27]. For a survey, we refer the readers to [13].

Since the matrices \(A\) and \(B\) are usually large and sparse, it may be more attractive to use iterative methods than direct methods for the solution of the saddle point problem (1.1). In the case that the matrix \(B\) has full row rank, many efficient iteration methods were proposed to solve the saddle point problems, for example, Uzawa-type methods [1, 2, 11, 15, 16], matrix splitting methods [5, 7–9, 23], residual algorithm [3], relaxation iterative methods [10, 22] and so on.
A Fast Shift-Splitting Iteration Method for Nonsymmetric Saddle Point Problems

When the matrix $B$ is rank deficient, the coefficient matrix of the equation (1.1) is singular, and the linear system (1.1) is called singular saddle point problems. There exist a lot of methods for solving singular saddle point problems in the literature. Generalized successive overrelaxation method was studied in [34], and the semi-convergence of this method was proved when it is applied to solve singular saddle point problems. Minimum residual and conjugate gradient methods were proposed for solving the rank-deficient saddle point problems in [21, 31], respectively. Inexact Uzawa method, which covers the Uzawa method, the preconditioned Uzawa method, and the parameterized method as special cases, was discussed for singular saddle point problems in [33], and the semi-convergence result under restrictions was proved by verifying two necessary and sufficient conditions. More numerical methods for singular saddle point problems could be found in [4, 19, 32] and the references therein.

In this paper, we construct a fast shift-splitting iteration method for nonsymmetric saddle point problems based on the ideas of the shift-splitting iteration method [12, 18] and the Hermitian and skew-Hermitian splitting technique [9, 23, 35]. The idea of shift-splitting iteration method was first proposed by Bai, Yin and Su in [12] for solving a class of non-Hermitian positive definite linear systems. Then, it was extended by Cao, Du and Niu in [17] to solve saddle point problems, and generalized by Salkuyeh for saddle point problems in [28]. After that, for nonsymmetric saddle point problems, Cao et al. in [18, 19] proposed the generalized shift-splitting (GSS) method

$$
\frac{1}{2} \begin{pmatrix} \alpha I + A & B^T \\ -B & \beta I \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \alpha I - A & -B^T \\ B & \beta I \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix},
$$

(1.2)

and Zhou et al. in [35] presented the modified shift-splitting (MSS) method

$$
\frac{1}{2} \begin{pmatrix} \alpha I + 2H & B^T \\ -B & \alpha I \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \alpha I - 2S & -B^T \\ B & \alpha I \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix},
$$

(1.3)

where $\alpha$ and $\beta$ are two given positive constants, $I$ is the identity matrix with appropriate dimension, and the matrices $H$ and $S$ are the symmetric (Hermitian) part and skew-symmetric (skew-Hermitian) part of the matrix $A$, respectively, i.e., $H = \frac{1}{2}(A + A^T)$, $S = \frac{1}{2}(A - A^T)$. Recently, Shen et al. applied the GSS iteration method to solve a broad class of nonsingular and singular generalized saddle point problems in [29]. In this paper, a fast shift-splitting iteration method is studied, which can be regarded as a special case of the BASI (block alternating splitting implicit) method proposed by Bai in [24]. Convergence and semi-convergence theories of this method for nonsingular and singular cases are carefully analyzed, respectively. Numerical experiments further show that the proposed method is efficient and feasible.

This paper is organized as follows. In Section 2, a fast shift-splitting iteration method for nonsymmetric saddle point problems is established. In Section 3, the convergence of the fast shift-splitting iteration method for nonsingular case is analyzed. In Section 4, semi-convergence of the fast shift-splitting iteration method for singular case is studied. In Section 5, numerical experiments are presented to illustrate the effectiveness and feasibility of the proposed method. Finally, a brief conclusion is given.
2. The Fast Shift-Splitting Iteration Method

Denote
\[ \mathcal{A} = \begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix}, \quad w = \begin{pmatrix} x \\ y \end{pmatrix}, \quad b = \begin{pmatrix} f \\ g \end{pmatrix}, \]
then the equation (1.1) can be rewritten as
\[ \mathcal{A} w = b. \tag{2.1} \]

The coefficient matrix \( \mathcal{A} \) can be split as follow
\[ \mathcal{A} = \mathcal{M} - \mathcal{N} = \begin{pmatrix} \alpha I + H B^T & 0 \\ -B & \alpha I \end{pmatrix} - \begin{pmatrix} \alpha I - S & 0 \\ 0 & \alpha I \end{pmatrix}, \]
where \( \alpha > 0 \) is a constant, the matrices \( H \) and \( S \) are the symmetric (Hermitian) part and skew-symmetric (skew-Hermitian) part of the matrix \( A \), respectively. By this special splitting, a fast shift-splitting iteration method can be defined as follow.

The fast shift-splitting (FSS) iteration method: Given an initial guess \( ((x^{(0)})^T, (y^{(0)})^T)^T \), for \( k = 0, 1, 2, \cdots \) until \( ((x^{(k)})^T, (y^{(k)})^T)^T \) converges, compute
\[ \begin{pmatrix} \alpha I + H B^T & 0 \\ -B & \alpha I \end{pmatrix} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha I - S & 0 \\ 0 & \alpha I \end{pmatrix} \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix}. \tag{2.2} \]

It is obvious that the matrix \( \mathcal{M} \) is invertible for nonsingular saddle point problems as long as \( \alpha > 0 \). For singular case, the matrix \( \mathcal{M} \) is also invertible because of
\[ \begin{pmatrix} I & 0 \\ B(\alpha I + H)^{-1} & I \end{pmatrix} \begin{pmatrix} \alpha I + H B^T & 0 \\ -B & \alpha I \end{pmatrix} \begin{pmatrix} I & -((\alpha I + H)^{-1} B^T) \\ 0 & I \end{pmatrix} \]
\[ = \begin{pmatrix} \alpha I + H & 0 \\ 0 & \alpha I + B(\alpha I + H)^{-1} B^T \end{pmatrix}. \]

Thus, for both nonsingular and singular saddle point problems, the iterative scheme (2.2) can be rewritten as
\[ \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \end{pmatrix} = \Gamma \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} + \mathcal{M}^{-1} \begin{pmatrix} f \\ g \end{pmatrix}, \tag{2.3} \]
where
\[ \Gamma = \begin{pmatrix} \alpha I + H B^T & 0 \\ -B & \alpha I \end{pmatrix}^{-1} \begin{pmatrix} \alpha I - S & 0 \\ 0 & \alpha I \end{pmatrix} \tag{2.4} \]
is the iteration matrix of FSS iteration method.

As a matter of fact, any matrix splitting can not only automatically lead to a splitting iteration method, but also naturally induce a preconditioner for Krylov subspace methods.
like GMRES, or its restarted version. The preconditioner corresponding to the FSS method (2.2) is given by
\[ M = \begin{pmatrix} \alpha I + H & B^T \\ -B & \alpha I \end{pmatrix}, \]
which is called the FSS preconditioner for the saddle point matrix $M$.

At each step of the FSS iteration (2.2) or applying preconditioner $M$ within a Krylov subspace methods, a linear system with $M$ as the coefficient matrix needs to be solved. That is to say, linear systems of the form $Mz = r$ needs to be solved for a given vector $r$ at each step. Since the matrix $M$ has the following matrix factorization
\[ M = \begin{pmatrix} I & \frac{1}{\alpha}B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} \alpha I + H + \frac{1}{\alpha}B^TB & 0 \\ 0 & \alpha I \end{pmatrix} \begin{pmatrix} I & 0 \\ -\frac{1}{\alpha}B & I \end{pmatrix}. \] (2.5)

Let $r = (r_1^T, r_2^T)^T$ and $z = (z_1^T, z_2^T)^T$, where $r_1, z_1 \in \mathbb{R}^n$ and $r_2, z_2 \in \mathbb{R}^m$. Then by (2.5), we have
\[ \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ \frac{1}{\alpha}B & I \end{pmatrix} \begin{pmatrix} \alpha I + H + \frac{1}{\alpha}B^TB & 0 \\ 0 & \alpha I \end{pmatrix}^{-1} \begin{pmatrix} I & -\frac{1}{\alpha}B^T \\ 0 & I \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}. \] (2.6)

Hence, the following algorithmic version of the FSS iteration method can be derived.

**Algorithms 2.1**

For a given vector $r = (r_1^T, r_2^T)^T$, the vector $z = (z_1^T, z_2^T)^T$ can be computed by (2.6) according to the following steps:
1. $t_1 = r_1 - \frac{1}{\alpha}B^T r_2$;
2. solve $(\alpha I + H + \frac{1}{\alpha}B^TB)z_1 = t_1$;
3. $z_2 = \frac{1}{\alpha}(r_2 + Bz_1)$.

In the Algorithm 2.1, a linear system with coefficient matrix $\alpha I + H + \frac{1}{\alpha}B^TB$ is required to be solved at each iteration. Since the coefficient matrix is symmetric positive definite for any $\alpha > 0$, the sub-linear system with the coefficient matrix $\alpha I + H + \frac{1}{\alpha}B^TB$ can be solved by the conjugate gradient (CG) method or some direct methods, such as, Cholesky or LU factorization in combination with AMD or column AMD reordering [17, 18].

In the following two sections, we will discuss the convergence and semi-convergence of the FSS method for nonsingular and singular saddle point problems, respectively.

### 3. Convergence analysis for nonsingular case

Let $\rho(\Gamma)$ be the spectral radius of the iteration matrix $\Gamma$. Then for nonsingular saddle point problems, the FSS iteration method is convergent if and only if $\rho(\Gamma) < 1$. To get a convergence condition, some lemmas are given initially.

**Lemma 3.1.** Let $A$ be nonsymmetric positive definite and $B$ be full row rank. Let $\lambda$ be an eigenvalue of the matrix $\Gamma$ and $[u^*, v^*]$ be the corresponding eigenvector, then $\lambda \neq 1$. 
Proof: From (2.4) we have
\[
\begin{bmatrix}
\alpha I - S & 0 \\
0 & \alpha I
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} = \begin{bmatrix}
\alpha I + H & B^T \\
-B & \alpha I
\end{bmatrix}
\begin{bmatrix}
u \\
\lambda u
\end{bmatrix},
\] (3.1)

If \( \lambda = 1 \), then from (3.1) we have
\[
\begin{bmatrix}
A & B^T \\
-B & 0
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix} = 0.
\]

Since the coefficient matrix is nonsingular, we have \( u = 0 \) and \( v = 0 \), which contradicts with the assumption that \([u^*, v^*]^*\) is an eigenvector of the iteration matrix \( \Gamma \). So \( \lambda \neq 1 \). \( \square \)

Lemma 3.2 ([23]). If \( S \) is a skew-Hermitian matrix, then \( iS \) (\( i \) is the imaginary unit) is a Hermitian matrix and \( u^*Su \) is a purely imaginary number or zero for all \( u \in \mathbb{C}^n \).

Lemma 3.3 ([26]). Both roots of the complex quadratic equation \( \lambda^2 + \phi \lambda + \varphi = 0 \) have modulus less than one if and only if \(|\phi - \bar{\phi}\varphi| + |\varphi|^2 < 1\), where \( \phi \) denotes the conjugate complex of \( \phi \).

Theorem 3.1. Let \( A \) be nonsymmetric positive definite and \( B \) be full row rank. Let \( \lambda \) be an eigenvalue of the matrix \( \Gamma \) and \([u^*, v^*]^*\) be the corresponding eigenvector. Denote
\[
a = \frac{u^*Hu}{u^*u}, \quad b = \frac{u^*B^T Bu}{u^*u}, \quad il = \frac{u^*Su}{u^*u},
\] (3.2)

where \( l \) is a real number. Then, the FSS iteration method is convergent if and only if the parameter \( \alpha \) satisfies the following conditions
\[
\left\{ \begin{array}{l}
\alpha^2l^2 < \alpha^2a^2 + b^2 + 2\alpha a^3 + 2b\alpha^2 + 2aba, \\
4a^2\alpha^4 + 2\alpha a^3(a^2 + 4b - l^2) + (5a^2b + 4b^2 - 3b^2l^2)d^2 + 4ab^2\alpha + b^3 > 0.
\end{array} \right.
\] (3.3)

Proof: From (3.1) we have
\[
\left\{ \begin{array}{l}
(\lambda - 1)au + \lambda B^Tv + \lambda Hu + Su = 0, \\
\lambda Bu + (1 - \lambda)av = 0.
\end{array} \right.
\] (3.4)

By Lemma 3.1, we know that \( \lambda \neq 1 \). In addition, we can get \( u \neq 0 \). Otherwise, by (3.4) we have \((1 - \lambda)a\alpha v = 0\). Then, it follows that \( v = 0 \), which contradicts with the assumption that \([u^*, v^*]^*\) is an eigenvector. Now, solving \( v \) from the second equation of (3.4) and substituting it into the first one, we have
\[
a^2(\lambda - 1)^2u + a(\lambda - 1)(\lambda H + S)u + \lambda^2B^Tu = 0.
\] (3.5)

Multiplying \( \frac{u^*}{\alpha} \) to both sides of the equation (3.5) from the left yields
\[
(a^2 + \alpha a + b)\lambda^2 + (-\alpha a - 2a^2 + il\alpha)\lambda + a^2 - il\alpha = 0.
\] (3.6)
Since $a > 0$, $b \geq 0$ and $\alpha > 0$, it follows that $a^2 + aa + b \neq 0$. Then the quadratic equation (3.6) can be written as $\lambda^2 + \phi \lambda + \varphi = 0$, where

$$
\phi = \frac{-a\alpha - 2a^2 + il\alpha}{a^2 + aa + b}, \quad \varphi = \frac{a^2 - il\alpha}{a^2 + aa + b}.
$$

By Lemma 3.3, we know that $|\lambda| < 1$ if and only if $|\phi - \bar{\phi}\varphi| + |\varphi|^2 < 1$. Define three auxiliary functions

$$
h_1(\alpha) = a^4 + a^2 l^2,
$$

$$
h_2(\alpha) = (a^2 + aa + b)^2,
$$

$$
h_3(\alpha) = (a^2 a^2 + ab\alpha + 2aa^3 + 2ba^2 - l^2 a^2)^2 + a^2 l^2 b^2,
$$

then we have

$$
|\phi - \bar{\phi}\varphi| + |\varphi|^2 = \frac{h_1(\alpha) + \sqrt{h_3(\alpha)}}{h_2(\alpha)},
$$

and $|\phi - \bar{\phi}\varphi| + |\varphi|^2 < 1$ if and only if

$$
h_1(\alpha) - h_2(\alpha) < 0, \quad \text{and} \quad (h_1(\alpha) - h_2(\alpha))^2 - h_3(\alpha) > 0.
$$

By careful calculation, we obtain that the FSS iteration method is convergent if and only if the parameter $\alpha$ satisfies the inequalities (3.3). Hence, the proof is completed.

Next, the special consequences of Theorem 3.1 is discussed.

From the inequalities (3.3), it can be seen that the FSS method is convergent when $a > l$ and $5l^2 + 4b > 3bl^2$. If $\alpha$ is located in the small neighborhood of zero, the inequalities (3.3) always hold true. Besides, when matrix $A$ is symmetric positive definite, the conditions (3.3) are reduced to $\alpha > 0$ since $H = A$ and $S = 0$.

### 4. Semi-Convergence Analysis for Singular Case

In this section, we discuss the semi-convergence property of the FSS iteration method (2.2) for solving singular saddle point problems. Firstly, some lemmas are introduced. Let $\sigma(A)$, $\rho(A)$, null($A$) and index($A$) be the spectral set, the spectral radius, the null subspace and the index of the matrix $A$, respectively.

The following lemma describes the semi-convergence property about the iteration scheme (2.3) when $\mathcal{A}$ is singular.

**Lemma 4.1** ([14]). *The iteration scheme (2.3) is semi-convergent if and only if the following two conditions are satisfied:*

1. *The elementary divisors of the iteration matrix $\Gamma$ associated with $\lambda = 1 \in \sigma(\Gamma)$ are linear,* i.e., $\text{rank}(I - \Gamma)^2 = \text{rank}(I - \Gamma)$, *or equivalently, index$(I - \Gamma) = 1$;*

2. *The pseudo-spectral radius satisfies* $\gamma(\Gamma) = \max\{|\lambda|, \lambda \in \sigma(\Gamma), \lambda \neq 1\} < 1$. 


4.1. The condition for $\text{rank}(I - \Gamma)^2 = \text{rank}(I - \Gamma)$

**Theorem 4.1.** Let $A$ be nonsymmetric positive definite and $B$ be rank deficient. Assume that $\alpha > 0$ and $\Gamma$ is the iteration matrix of the FSS iteration method, then $\text{rank}(I - \Gamma)^2 = \text{rank}(I - \Gamma)$ holds if

$$\text{null}((\mathcal{M}^{-1} \mathcal{A})^2) = \text{null}(\mathcal{M}^{-1} \mathcal{A}).$$

Proof. Since $\Gamma = \mathcal{M}^{-1} \mathcal{N} = I - \mathcal{M}^{-1} \mathcal{A}$, $\text{rank}(I - \Gamma)^2 = \text{rank}(I - \Gamma)$ holds if

$$\text{null}((\mathcal{M}^{-1} \mathcal{A})^2) = \text{null}(\mathcal{M}^{-1} \mathcal{A}).$$

It is obvious that $\text{null}((\mathcal{M}^{-1} \mathcal{A})^2) \supset \text{null}(\mathcal{M}^{-1} \mathcal{A})$. Now, we only need show

$$\text{null}((\mathcal{M}^{-1} \mathcal{A})^2) \subseteq \text{null}(\mathcal{M}^{-1} \mathcal{A}). \quad (4.1)$$

Let $p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \in \text{null}((\mathcal{M}^{-1} \mathcal{A})^2)$ with $p_1 \in \mathbb{R}^n$ and $p_2 \in \mathbb{R}^m$. It must be satisfied $(\mathcal{M}^{-1} \mathcal{A})^2 p = 0$. Denote by $q = \mathcal{M}^{-1} \mathcal{A} p$. Let $q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \in \mathbb{R}^{n+m}$. To prove (4.1), we only need to prove $q = 0$, i.e., $q_1 = 0$ and $q_2 = 0$. On one hand, we have

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{pmatrix} \alpha I + H & B^T \\ -B & \alpha I \end{pmatrix}^{-1} \begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix},$$

which can be rewritten as

$$\begin{cases} q_1 = (\alpha I + H + \frac{1}{\alpha} B^T B)^{-1} \left( A + \frac{1}{\alpha} B^T B \right) p_1 + (\alpha I + H + \frac{1}{\alpha} B^T B)^{-1} B^T p_2, \\ q_2 = \frac{1}{\alpha} \left( A + \frac{1}{\alpha} B^T B \right)^{-1} \left( A + \frac{1}{\alpha} B^T B - \frac{1}{\alpha} B \right) p_1 + \frac{1}{\alpha} B \left( A + H + \frac{1}{\alpha} B^T B \right)^{-1} B^T p_2. \end{cases} \quad (4.2)$$

On the other hand, note that $\text{null}(\mathcal{M}^{-1} \mathcal{A}) = \text{null}(\mathcal{A})$, then $(\mathcal{M}^{-1} \mathcal{A})^2 p = 0$ is equivalent to $\mathcal{A} q = 0$, i.e.,

$$\begin{cases} A q_1 + B^T q_2 = 0, \\ -B q_1 = 0. \end{cases} \quad (4.3)$$

Since the matrix $A$ is nonsingular, solving $q_1$ from the first equality of (4.3) and substituting it into the second equality of (4.3) gives $B A^{-1} B^T q_2 = 0$, which means $q_2^T B A^{-1} B^T q_2 = (B^T q_2)^T A^{-1} (B^T q_2) = 0$. Owing to the positive definiteness of the matrix $A^{-1}$, we obtain $B^T q_2 = 0$. Substituting it into the first equality of (4.3), we get $q_1 = 0$.

Since $q_1 = 0$, the first equality of (4.2) becomes

$$\left( \alpha I + H + \frac{1}{\alpha} B^T B \right)^{-1} \left( A + \frac{1}{\alpha} B^T B \right) p_1 + \left( \alpha I + H + \frac{1}{\alpha} B^T B \right)^{-1} B^T p_2 = 0.$$

Taking it into the second equality of (4.2) gives $q_2 = -\frac{1}{\alpha} B p_1$. Since $B^T q_2 = 0$, we have $p_1^T B^T B p_1 = 0$, which implies $B p_1 = 0$ i.e., $q_2 = 0$. Hence, we obtain $q = [q_1^T, q_2^T]^T = 0$. Thus, the proof is completed. \qed
4.2. The condition for $\gamma(\Gamma) < 1$

In order to discuss the condition for $\gamma(\Gamma) < 1$, without loss of generality, we assume that the rank of the rank deficient matrix $B \in \mathbb{R}^{m \times n}$ is $r( < m)$. Let $B = U(\begin{bmatrix} B_r \\ 0 \end{bmatrix})V^T$ be the singular value decomposition of the matrix $B$, where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, $B_r = [\Sigma_r \ 0] \in \mathbb{R}^{r \times n}$ with $\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_r) \in \mathbb{R}^{r \times r}$. Denote $\hat{A} = V^TAV$, $\hat{B} = V^THV$, $\hat{S} = V^TSV$ and

$$
\Omega = \begin{pmatrix} aI & 0 \\ 0 & aI \end{pmatrix}, \quad \mathcal{A}_1 = \begin{pmatrix} H & B^T \\ -B & 0 \end{pmatrix} \quad \text{and} \quad \mathcal{A}_2 = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix},
$$

then iteration matrix can be rewritten as $\Gamma = (\Omega + \mathcal{A}_1)^{-1}(\Omega - \mathcal{A}_2)$.

In addition, consider the block diagonal matrix

$$
P = \begin{pmatrix} V & 0 \\ 0 & U \end{pmatrix} \in \mathbb{R}^{(m+n) \times (m+n)},
$$

which is an orthogonal matrix. Then, the iteration matrix $\Gamma$ is similar to the following matrix

$$
\tilde{\Gamma} = P^T \Gamma P = P^T(\Omega + \mathcal{A}_1)^{-1}(\Omega - \mathcal{A}_2)P = (P^T(\Omega + \mathcal{A}_1)P)^{-1}(P^T(\Omega - \mathcal{A}_2)P)
$$

$$
= \begin{pmatrix} aI + V^THV & V^TB^TU \\ -U^TBV & aI \end{pmatrix}^{-1} \begin{pmatrix} aI - V^TSV & 0 \\ 0 & aI \end{pmatrix}
$$

$$
= \begin{pmatrix} aI + V^THV & B_r^T \\ -B_r & aI \end{pmatrix}^{-1} \begin{pmatrix} aI - V^TSV & 0 \\ 0 & aI \end{pmatrix}
$$

$$
= \begin{pmatrix} aI + V^THV & B_r^T \\ -B_r & aI \end{pmatrix}^{-1} \begin{pmatrix} aI - V^TSV & 0 \\ 0 & aI \end{pmatrix} 
$$

$$
= \left( (\Omega + \mathcal{A}_1)^{-1}(\Omega - \mathcal{A}_2) 0 \\ 0 I \right),
$$

where

$$
\mathcal{A}_1 = \begin{pmatrix} \hat{A} \\ -B_r \end{pmatrix}, \quad \text{and} \quad \mathcal{A}_2 = \begin{pmatrix} \hat{S} \\ 0 \end{pmatrix},
$$

are all $(n+r) \times (n+r)$ matrices. Since $\hat{A}$ is positive definite, and $B_r$ is full row rank, verifying the condition for $\gamma(\Gamma) < 1$ is equivalent to studying the convergence of the FSS iteration method which is used to solve the following nonsingular saddle point problems

$$
\begin{pmatrix} \hat{A} & B_r^T \\ -B_r & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ \hat{g} \end{pmatrix}.
$$
Let $[\hat{u}^*, \hat{v}^*]$ be an eigenvector corresponding to an eigenvalue of the iteration matrix $\bar{\Gamma} = (\Omega + \mathcal{A}_1)^{-1}(\Omega - \mathcal{A}_2)$. Denote 
\[ \hat{a} = \frac{\hat{u}^* \hat{H} \hat{u}}{\hat{u}^* \hat{u}}, \quad \hat{b} = \frac{\hat{u}^* B^T B \hat{u}}{\hat{u}^* \hat{u}}, \quad \hat{\ell} = \frac{\hat{u}^* \hat{S} \hat{u}}{\hat{u}^* \hat{u}}. \] (4.4)

Then, similar to the proof of Theorem 3.1, the condition for $\rho(\bar{\Gamma}) < 1$ can be easily obtained. Thus, the condition for $\gamma(\Gamma) < 1$ can be derived. We summarize the above discussion in the following theorem.

**Theorem 4.2.** Let $A \in \mathbb{R}^{n \times n}$ be nonsymmetric positive definite and $B \in \mathbb{R}^{m \times n}$ be of rank deficient. Suppose that $\alpha > 0$ is a given constant. Then the pseudo-spectral radius of the matrix $\Gamma$ is less than 1, i.e., $\gamma(\Gamma) < 1$ if and only if the parameter $\alpha$ satisfies the following conditions

\[ \frac{a^2}{a^2 + b^2 + 2 \hat{\ell}^2} < \frac{a^2 \hat{S} + 2 \hat{b} \hat{a}}{a^2 + 2 \hat{b} \hat{a} + \hat{b}^2}, \]
\[ 4 \hat{a}^2 \hat{S}^2 + 2 \hat{b} \hat{S} \hat{a} + \hat{b}^2 \hat{S} > 0. \] (4.5)

The following theorem readily follows from Lemma 4.1, Theorem 4.1 and Theorem 4.2.

**Theorem 4.3.** Let $A$ be nonsymmetric positive definite and $B$ be of rank deficient. Suppose that $\alpha > 0$ is a given constant. Then the FSS iteration method is semi-convergent for solving singular saddle point problems if and only if the parameter $\alpha$ satisfies the inequalities (4.5).

**5. Numerical Experiments**

In this section, some numerical experiments are presented to illustrate the feasibility and effectiveness of the FSS iteration method for nonsingular and singular saddle point problems (1.1). To show the advantages of the FSS iteration method over the GSS method in [18] and MSS method in [35], we compare the numerical results of these methods in the sense of the number of iteration steps (denoted by ‘IT’) and elapsed CPU time (denoted by ‘CPU’). Numerical results of the well-known GMRES method and the preconditioned GMRES methods are also given, which can further show that the induced FSS preconditioner is much better than the induced GSS and MSS preconditioners for solving nonsymmetric saddle point problems (1.1). The GSS and MSS preconditioners, which are induced by the iteration methods (1.2) and (1.3), are defined by

\[ \mathcal{P}_{GSS} = \frac{1}{2} \begin{pmatrix} \alpha I + A & B^T \\ -B & \beta I \end{pmatrix}, \quad \mathcal{P}_{MSS} = \frac{1}{2} \begin{pmatrix} \alpha I + 2H & B^T \\ -B & \alpha I \end{pmatrix}, \]

respectively. Correspondingly, we use $\mathcal{P}_{FSS}$ to denote the FSS preconditioner which is defined in Section 2.

In the following numerical experiments, the optimal parameters in the iteration methods are found by experiments, resulting in the least iteration number. The same parameters are used for the GSS, MSS and FSS preconditioners. It should be noted that ‘$\varphi$’ denotes the GMRES method without preconditioning in the following given tables. In all the test
problems discussed in this section, the sub-linear systems are solved by direct methods. In Matlab, this corresponds to compute the Cholesky or LU factorization in combination with AMD or column AMD reordering. The initial vector is set to be the zero vector and the iterations are terminated if the current iterations satisfy $\text{RES} := \|b - Aw_k\|_2/\|b\|_2 < 10^{-6}$. All tests are performed on a computer with Intel Core i5 CPU 2.50 GHz, 4.0GB memory.

5.1. Nonsingular case

Example 5.1. The nonsingular saddle point problem arising from a model Stokes equation has the following coefficient sub-matrices [17]:

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

and

$$T = \frac{\nu}{h^2} \cdot \text{tridiag}(-1, 2, -1) + \frac{1}{2h} \cdot \text{tridiag}(-1, 0, 1) \in \mathbb{R}^{q \times q},$$

$$F = \frac{1}{h} \cdot \text{tridiag}(-1, 0, 1) \in \mathbb{R}^{q \times q}.$$  

Here, $\otimes$ denotes the Kronecker product symbol, $\nu$ is the viscosity scalar and $h = \frac{1}{q+1}$ is the discretization mesh size.

In this example, we choose $\nu = 1$ and $\nu = 0.1$. For each $\nu$, four different $q$ are used, i.e., $q = 16, 32, 64, 128$.

In Table 1, numerical results of the GSS, MSS and FSS iteration methods with $\nu = 1$ are given. The optimal parameters of the three methods are also presented. From Table 1,

<table>
<thead>
<tr>
<th>Method</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\text{IT}$</th>
<th>$\text{CPU}$</th>
<th>RES</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS</td>
<td>255</td>
<td>1</td>
<td>57</td>
<td>0.17</td>
<td>9.84e-7</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>1</td>
<td>99</td>
<td>1.44</td>
<td>9.39e-7</td>
</tr>
<tr>
<td></td>
<td>920</td>
<td>1</td>
<td>159</td>
<td>9.35</td>
<td>9.33e-7</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>1</td>
<td>279</td>
<td>79.33</td>
<td>9.78e-7</td>
</tr>
<tr>
<td>MSS</td>
<td>0.6</td>
<td>0.01</td>
<td>0.09</td>
<td>0.09</td>
<td>8.71e-7</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.001</td>
<td>0.63</td>
<td>0.63</td>
<td>7.25e-7</td>
</tr>
<tr>
<td></td>
<td>0.3</td>
<td></td>
<td>3.33</td>
<td>3.33</td>
<td>9.69e-7</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td></td>
<td>18.60</td>
<td>18.60</td>
<td>7.43e-7</td>
</tr>
<tr>
<td>FSS</td>
<td>0.01</td>
<td>0.5</td>
<td>0.95</td>
<td>0.09</td>
<td>2.45e-7</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>4</td>
<td>0.06</td>
<td>4.06</td>
<td>3.88e-7</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>4</td>
<td>0.25</td>
<td>4.25</td>
<td>9.24e-8</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>3</td>
<td></td>
<td></td>
<td>4.49e-7</td>
</tr>
</tbody>
</table>
it is seen that the iteration steps and the elapsed CPU time of the proposed FSS iteration method are much less than those of the other two methods. These results show that the proposed FSS iteration method is the most efficient among the three methods.

In Table 2, numerical results of the GMRES and the preconditioned GMRES methods with the GSS, MSS and FSS preconditioners are given for $\nu = 1$. Numerical results show that the GSS, MSS and FSS preconditioners can accelerate the convergence rate of the GMRES greatly. Besides, both the iteration steps and the elapsed CPU time show that the proposed FSS preconditioner is much more efficient than the other two preconditioners.

In Table 3, numerical results and the optimal parameters of the GSS, MSS and FSS iteration methods for $\nu = 0.1$ are given. From Table 3, it is seen that FSS iteration method with the optimal iteration parameters succeeds to quickly produce approximate solutions. Moreover, the FSS method always outperforms the other methods considerably in terms of iteration steps and CPU time.

Numerical results for the GMRES and the three preconditioned GMRES methods with $\nu = 0.1$ are given in Table 4. It is seen that the FSS preconditioner outperforms the other two preconditioners considerably from the viewpoint of both iteration steps and CPU time.

In Table 5, numerical results of direct method and FSS method which employs conjugate gradient (CG) method as its inner processes for $\nu = 1$ are given. Note that the number in brackets is average iteration steps of CG method. It is seen that when the grid size is less than or equal to $256 \times 256$, direct method has obvious advantage in CPU time. But when the grid size is larger than or equal to $512 \times 512$, the FSS iteration method is superior to the direct method considerably in CPU time. It means that the FSS iteration method is feasible for large and sparse saddle point problems.
Table 3: Numerical results of iteration methods for Example 5.1 ($\nu = 0.1$).

<table>
<thead>
<tr>
<th>Method</th>
<th>$q$</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>20</td>
<td>40</td>
<td>90</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>9.993</td>
<td>9.992</td>
<td>9.991</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>GSS</td>
<td>IT</td>
<td>52</td>
<td>93</td>
<td>161</td>
<td>280</td>
</tr>
<tr>
<td>CPU</td>
<td>0.14</td>
<td>1.34</td>
<td>12.16</td>
<td>81.856</td>
<td></td>
</tr>
<tr>
<td>RES</td>
<td>7.67e-7</td>
<td>9.70e-7</td>
<td>9.40e-7</td>
<td>9.98e-7</td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>17</td>
<td>13.7</td>
<td>12</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>IT</td>
<td>82</td>
<td>121</td>
<td>174</td>
<td>269</td>
<td></td>
</tr>
<tr>
<td>MSS</td>
<td>CPU</td>
<td>0.19</td>
<td>1.75</td>
<td>13.23</td>
<td>78.29</td>
</tr>
<tr>
<td>RES</td>
<td>8.90e-7</td>
<td>9.75e-7</td>
<td>9.71e-7</td>
<td>9.94e-7</td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>2.7</td>
<td>2</td>
<td>1</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>IT</td>
<td>37</td>
<td>42</td>
<td>40</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>FSS</td>
<td>CPU</td>
<td>0.09</td>
<td>0.61</td>
<td>3.08</td>
<td>9.83</td>
</tr>
<tr>
<td>RES</td>
<td>9.47e-7</td>
<td>8.10e-7</td>
<td>8.96e-7</td>
<td>9.12e-7</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Numerical results of preconditioned GMRES methods for Example 5.1 ($\nu = 0.1$).

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>$q$</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{J}$</td>
<td>IT</td>
<td>105</td>
<td>208</td>
<td>421</td>
<td>825</td>
</tr>
<tr>
<td>CPU</td>
<td>0.24</td>
<td>3.01</td>
<td>39.83</td>
<td>415.65</td>
<td></td>
</tr>
<tr>
<td>RES</td>
<td>9.26e-7</td>
<td>9.00e-7</td>
<td>9.98e-7</td>
<td>9.86e-7</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P}_{GSS}$</td>
<td>IT</td>
<td>38</td>
<td>38</td>
<td>46</td>
<td>51</td>
</tr>
<tr>
<td>CPU</td>
<td>0.25</td>
<td>1.20</td>
<td>7.59</td>
<td>41.59</td>
<td></td>
</tr>
<tr>
<td>RES</td>
<td>9.66e-7</td>
<td>9.89e-7</td>
<td>9.50e-7</td>
<td>9.60e-7</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P}_{MSS}$</td>
<td>IT</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>CPU</td>
<td>0.13</td>
<td>0.67</td>
<td>3.41</td>
<td>14.81</td>
<td></td>
</tr>
<tr>
<td>RES</td>
<td>9.91e-7</td>
<td>9.47e-7</td>
<td>4.79e-7</td>
<td>8.45e-7</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P}_{FSS}$</td>
<td>IT</td>
<td>20</td>
<td>17</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>CPU</td>
<td>0.13</td>
<td>0.57</td>
<td>2.18</td>
<td>8.10</td>
<td></td>
</tr>
<tr>
<td>RES</td>
<td>6.89e-7</td>
<td>8.01e-7</td>
<td>8.35e-7</td>
<td>6.62e-7</td>
<td></td>
</tr>
</tbody>
</table>

5.2. Singular case

Example 5.2 ([32]). The singular saddle point problem has the following coefficient submatrices:

$$A = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2q^2 \times 2q^2}, \quad \tilde{B}^T = [\tilde{b}_1^T \ b_2^T] \in \mathbb{R}^{2q^2 \times (q^2+2)},$$
with
\[ \hat{B}^T = \left( \begin{array}{c} I \otimes F \\ F \otimes I \end{array} \right) \in \mathbb{R}^{2q^2 \times q^2}, \quad \mathbf{b}_1 = \hat{B}^T \left( \begin{array}{c} \mathbf{e} \\ 0 \end{array} \right), \quad \mathbf{b}_2 = \hat{B}^T \left( \begin{array}{c} 0 \\ \mathbf{e} \end{array} \right), \quad \mathbf{e} = (1, 1, \ldots, 1) \in \mathbb{R}^{q^2/2} \]

and
\[ T = \frac{\nu}{h^2} \cdot \text{tridiag}(-1, 2, -1) + \frac{1}{2h} \cdot \text{tridiag}(-1, 0, 1) \in \mathbb{R}^{q \times q}, \]
\[ F = \frac{1}{h} \cdot \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{q \times q}. \]

Here, \( \otimes \) denotes the Kronecker product symbol, \( \nu \) is the viscosity scalar and \( h = \frac{1}{q+1} \) is the discretization mesh size.

This singular problem is a technical modification of Example 5.1. Here, matrix \( B \) is an augmentation of the full rank matrix \( \hat{B} \) with two linearly independent vectors \( \mathbf{b}_1 \) and \( \mathbf{b}_2 \). As \( \mathbf{b}_1 \) and \( \mathbf{b}_2 \) are linear combinations of the columns of the matrix \( \hat{B} \); \( B \) is a rank-deficient
matrix. In this example, \( \nu \) and \( q \) are chosen to be the same as the ones in Example 5.1. Note that the optimal parameters of the GSS method are as the ones as in [19].

In Tables 6 and 8, the optimal parameters and numerical results of the GSS, MSS and FSS iteration methods for solving singular saddle point problem are presented for \( \nu = 1 \) and \( \nu = 0.1 \), respectively. From Table 6 and 8, it is seen that the proposed FSS iteration method converges very fast. These results show that proposed FSS iteration method is much efficient for solving singular saddle point problem.
Numerical results of GMRES and preconditioned GMRES methods for $\nu = 1$ and $\nu = 0.1$ are listed in Tables 7 and 9, respectively. It is observed that the three preconditioners can accelerate the convergence rate of the GMRES method largely, and the FSS preconditioners lead to the best numerical results.

**Example 5.3.** The second test singular nonsymmetric saddle point problem arises from the linearized version of the steady state Navier-Stokes equation with suitable boundary conditions.
conditions
\[-\nu \Delta \mathbf{u} + (\mathbf{w} \cdot \Delta) \mathbf{u} + \nabla p = f, \quad -\nabla \cdot \mathbf{u} = 0, \text{ in } \Omega, \tag{5.1}\]

which is obtained when the steady-state Navier-Stokes equation is linearized by the Picard iteration. Here \( \Omega \) is a bounded domain, \( \nu > 0 \) is the viscosity, \( \mathbf{u} \) represents velocity, and \( p \) represents pressure. Using the IFISS software package [20] to discretize the regularised-lid driven cavity problem on the unit square domain with \( Q_2 - Q_1 \) mixed finite element method on uniform grid. Here, we take three viscosity values \( \nu = 1, 0.1, 0.01 \) and four grids, i.e.,

Table 11: Numerical results of preconditioned GMRES methods for Example 5.3 (\( \nu = 1 \)).

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>Grids</th>
<th>16\times16</th>
<th>32\times32</th>
<th>64\times64</th>
<th>128\times128</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{I} )</td>
<td>IT</td>
<td>203</td>
<td>332</td>
<td>532</td>
<td>866</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.78</td>
<td>4.87</td>
<td>36.07</td>
<td>365.20</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>9.73e-7</td>
<td>9.92e-7</td>
<td>9.97e-7</td>
<td>9.95e-7</td>
</tr>
<tr>
<td>( \mathcal{P}_{\text{GSS}} )</td>
<td>IT</td>
<td>22</td>
<td>29</td>
<td>39</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.08</td>
<td>1.12</td>
<td>5.63</td>
<td>80.18</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>5.78e-7</td>
<td>8.02e-7</td>
<td>9.01e-7</td>
<td>8.09e-7</td>
</tr>
<tr>
<td>( \mathcal{P}_{\text{MSS}} )</td>
<td>IT</td>
<td>23</td>
<td>25</td>
<td>29</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.07</td>
<td>0.28</td>
<td>4.23</td>
<td>38.29</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>8.22e-7</td>
<td>9.73e-7</td>
<td>9.27e-7</td>
<td>9.79e-7</td>
</tr>
<tr>
<td>( \mathcal{P}_{\text{FSS}} )</td>
<td>IT</td>
<td>6</td>
<td>9</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.02</td>
<td>0.08</td>
<td>0.05</td>
<td>1.75</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>4.22e-7</td>
<td>9.66e-7</td>
<td>8.63e-8</td>
<td>4.61e-7</td>
</tr>
</tbody>
</table>

Table 12: Numerical results of iteration methods for Example 5.3 (\( \nu = 0.1 \)).

<table>
<thead>
<tr>
<th>Method</th>
<th>Grids</th>
<th>16\times16</th>
<th>32\times32</th>
<th>64\times64</th>
<th>128\times128</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS</td>
<td>( \alpha )</td>
<td>0.1</td>
<td>0.05</td>
<td>0.03</td>
<td>8e-3</td>
</tr>
<tr>
<td></td>
<td>( \beta )</td>
<td>0.055</td>
<td>0.015</td>
<td>0.004</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>IT</td>
<td>81</td>
<td>143</td>
<td>268</td>
<td>525</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.34</td>
<td>2.64</td>
<td>25.07</td>
<td>251.61</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>9.19e-7</td>
<td>8.52e-7</td>
<td>9.58e-7</td>
<td>9.68e-7</td>
</tr>
<tr>
<td>MSS</td>
<td>( \alpha )</td>
<td>0.08</td>
<td>0.02</td>
<td>0.008</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>IT</td>
<td>83</td>
<td>101</td>
<td>109</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.31</td>
<td>1.90</td>
<td>10.14</td>
<td>68.01</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>9.80e-7</td>
<td>9.73e-7</td>
<td>9.90e-7</td>
<td>9.73e-7</td>
</tr>
<tr>
<td>FSS</td>
<td>( \alpha )</td>
<td>0.001</td>
<td>0.001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>IT</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.02</td>
<td>0.11</td>
<td>0.27</td>
<td>2.46</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>5.22e-7</td>
<td>5.15e-7</td>
<td>9.84e-7</td>
<td>2.31e-7</td>
</tr>
</tbody>
</table>
Table 13: Numerical results of preconditioned GMRES methods for Example 5.3 ($\nu = 0.1$).

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>(16 \times 16)</th>
<th>(32 \times 32)</th>
<th>(64 \times 64)</th>
<th>(128 \times 128)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
<td>CPU</td>
<td>RES</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{I}$</td>
<td>127</td>
<td>0.31</td>
<td>9.95e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>260</td>
<td>2.89</td>
<td>9.83e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>528</td>
<td>35.43</td>
<td>9.93e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>927</td>
<td>425.26</td>
<td>9.99e-7</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P}_{GSS}$</td>
<td>25</td>
<td>0.06</td>
<td>5.77e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>1.23</td>
<td>8.07e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>7.75</td>
<td>5.24e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>92</td>
<td>83.78</td>
<td>8.09e-7</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P}_{MSS}$</td>
<td>29</td>
<td>0.08</td>
<td>6.79e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>1.19</td>
<td>8.06e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>5.76</td>
<td>7.60e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>48.20</td>
<td>9.79e-7</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P}_{FSS}$</td>
<td>5</td>
<td>0.02</td>
<td>2.58e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>0.06</td>
<td>2.82e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.76</td>
<td>2.31e-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7.82</td>
<td>4.61e-7</td>
<td></td>
</tr>
</tbody>
</table>

Table 14: Numerical results of iteration methods for Example 5.3 ($\nu = 0.01$).

<table>
<thead>
<tr>
<th>Method</th>
<th>(16 \times 16)</th>
<th>(32 \times 32)</th>
<th>(64 \times 64)</th>
<th>(128 \times 128)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IT</td>
<td>CPU</td>
<td>RES</td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.05</td>
<td>0.03</td>
<td>0.01</td>
<td>0.006</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.14</td>
<td>0.02</td>
<td>0.002</td>
<td>0.0008</td>
</tr>
<tr>
<td>GSS</td>
<td>142</td>
<td>245</td>
<td>691</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.64</td>
<td>4.66</td>
<td>66.78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.19e-7</td>
<td>8.52e-7</td>
<td>9.58e-7</td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.3</td>
<td>0.15</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>IT</td>
<td>477</td>
<td>1000</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>MSS</td>
<td>0.31</td>
<td>1.90</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>9.99e-7</td>
<td>8.52e-6</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.07</td>
<td>0.022</td>
<td>0.006</td>
<td>0.0014</td>
</tr>
<tr>
<td>IT</td>
<td>83</td>
<td>83</td>
<td>80</td>
<td>63</td>
</tr>
<tr>
<td>FSS</td>
<td>0.34</td>
<td>1.45</td>
<td>7.41</td>
<td>31.20</td>
</tr>
<tr>
<td>RES</td>
<td>9.73e-7</td>
<td>9.76e-7</td>
<td>9.87e-7</td>
<td>9.13e-7</td>
</tr>
</tbody>
</table>

16 × 16, 32 × 32, 64 × 64 and 128 × 128. Note that the rank of the matrix $B$ in all test nonsymmetric saddle point matrices is $m - 1$.

In Tables 10 and 12, numerical results of the three methods for singular saddle point problems with $\nu = 1$ and $\nu = 0.1$ are listed, respectively. From Tables 10 and 12, it is observed that both the iteration steps and the elapsed CPU time of the FSS iteration method are much less than the other two methods, which indicates the proposed FSS iteration method converges fast.
Table 15: Numerical results of preconditioned GMRES methods for Example 5.3 (\(\nu = 0.01\)).

<table>
<thead>
<tr>
<th>Preconditioner</th>
<th>Grids</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16×16</td>
<td>32×32</td>
<td>64×64</td>
<td>128×128</td>
<td></td>
</tr>
<tr>
<td>(\mathcal{P}_G)</td>
<td>IT</td>
<td>192</td>
<td>318</td>
<td>570</td>
<td>980</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.78</td>
<td>4.48</td>
<td>43.73</td>
<td>487.94</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>9.55e-7</td>
<td>9.92e-7</td>
<td>9.91e-7</td>
<td>9.97e-7</td>
</tr>
<tr>
<td>(\mathcal{P}_{GSS})</td>
<td>(\alpha)</td>
<td>0.05</td>
<td>0.03</td>
<td>0.01</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>(\beta)</td>
<td>0.14</td>
<td>0.02</td>
<td>0.002</td>
<td>0.0008</td>
</tr>
<tr>
<td></td>
<td>IT</td>
<td>42</td>
<td>61</td>
<td>52</td>
<td>62</td>
</tr>
<tr>
<td>(\mathcal{P}_M)</td>
<td>CPU</td>
<td>0.11</td>
<td>1.53</td>
<td>28.48</td>
<td>183.78</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>5.77e-7</td>
<td>8.07e-7</td>
<td>5.24e-7</td>
<td>8.09e-7</td>
</tr>
<tr>
<td>(\mathcal{P}_F)</td>
<td>(\alpha)</td>
<td>0.001</td>
<td>0.001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>IT</td>
<td>29</td>
<td>32</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.03</td>
<td>0.36</td>
<td>4.51</td>
<td>48.20</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>7.41e-7</td>
<td>6.90e-7</td>
<td>6.22e-7</td>
<td>7.79e-7</td>
</tr>
<tr>
<td>(\mathcal{P}_F)</td>
<td>(\alpha)</td>
<td>0.001</td>
<td>0.001</td>
<td>0.0001</td>
<td>0.0001</td>
</tr>
<tr>
<td></td>
<td>IT</td>
<td>28</td>
<td>25</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>0.03</td>
<td>0.27</td>
<td>3.23</td>
<td>33.92</td>
</tr>
<tr>
<td></td>
<td>RES</td>
<td>6.36e-7</td>
<td>7.60e-7</td>
<td>9.89e-7</td>
<td>9.61e-7</td>
</tr>
</tbody>
</table>

In Tables 11 and 13, numerical results of the GMRES and preconditioned GMRES are listed for \(\nu = 1\) and \(\nu = 0.1\), respectively. From Tables 11 and 13, it is seen that the GMRES with three preconditioners can be effective and feasible for solving singular saddle point problems. Moreover the FSS preconditioner is the most effective.

In Table 14, numerical results of GSS, MSS and FSS methods are listed for \(\nu = 0.01\). And the numerical results of corresponding preconditioned GMRES are listed for \(\nu = 0.01\) in Table 15. From Tables 14 and 15, it can be seen that FSS method and GMRES with FSS preconditioners for singular saddle point problems with \(\nu = 0.01\) are effective and feasible.

When \(\nu = 0.001\), GSS, MSS and FSS iteration methods fail to achieve the stopping rule at the maximal iteration steps. One of the possible reason is that the coefficient matrix has a strongly dominant skew-symmetric part. In this case, we refer the reader to the skew-Hermitian triangular splitting iteration methods in [24, 25, 30].

6. Conclusion

In this paper, a fast shift-splitting iteration method is proposed for solving nonsymmetric saddle point problems. The convergence and semi-convergence conditions of the fast shift-splitting iteration method for nonsingular and singular saddle point problems are presented, respectively. Numerical experiments show the effectiveness and feasibility of the fast shift-splitting iteration method for both nonsingular and singular saddle point problems.
Acknowledgments

The authors are very indebted to the anonymous referees for their helpful comments and valuable suggestions, which have improved the presentation of this paper significantly.

References

A Fast Shift-Splitting Iteration Method for Nonsymmetric Saddle Point Problems