Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(7): 13508–13536.
DOI: 10.3934/math.2022747
Received: 23 April 2022
Accepted: 05 May 2022
Published: 20 May 2022

Research article

A generalized Shift-HSS splitting method for nonsingular saddle point problems

Zhuo-Hong Huang[1,2,*]

[1] School of Science, Chongqing University of Technology, Chongqing, 400054, China
[2] School of Mathematics and Physics, Guangxi Minzu University, Nanning, 530006, China

* **Correspondence:** Email: zhuohonghuang@163.com.

Abstract: In this paper, we propose a generalized shift-HSS (denoted by SFHSS) iteration method for solving nonsingular saddle point systems with nonsymmetric positive definite (1,1)-block sub-matrix, and theoretically verify its convergence property. In addition, we discuss the algebraic properties of the resulted SFHSS preconditioner and estimate the sharp eigenvalue bounds of the related preconditioned matrix. Finally, numerical experiments are given to support our theoretical results and reveal that the new method is feasible and effective.

Keywords: generalized shift-HSS splitting; convergence property; nonsymmetric positive definite; nonsingular; Krylov subspace methods
Mathematics Subject Classification: 65F08, 65F10

1. Introduction

Consider the large and sparse saddle-point problem $\mathcal{A}x = b$ of form:

$$\mathcal{A}x = \begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsymmetric positive definite, in other words, its symmetric part $H = \frac{1}{2}(A + A^T)$ is positive definite, $B \in \mathbb{R}^{n\times m}$ $(n \gg m)$ is a full column rank matrix, $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^m$ are given vectors, and B^T denotes the transpose of B. Under these assumptions, the nonsingularity of $\mathcal{A}$ assures that the existence and uniqueness of the saddle problem (1.1). When the matrices A and B are large and sparse, this class of linear system of the form (1.1) arises in many problems of scientific computing and engineering applications, including computational fluid dynamics, optimization, constrained least squares problems, generalized least squares problems, incompressible flow problems, mixed finite element approximation of elliptic PDEs and so forth. See [1] and references therein for more examples and additional information. In addition, we also refer to [2] for periodic Sylvester matrix equations and generalized coupled Sylvester matrix equations.

In general, the coefficient matrix $\mathcal{A}$ of (1.1) is usually large, sparse and extremely ill-conditioned, it is a common belief that iteration methods become more attractive than direct methods in terms of storage requirements and computing time. As is well known, a clustered spectrum of preconditioned matrix often results in rapid convergence for Krylov subspace methods. Therefore, in light of the special structure of the saddle point problem (1.1), for the sake of achieving rapid convergence rate and improving computational efficiency, a lot of effective and practical preconditioning techniques have been investigated in the literatures, such as preconditioned Krylov subspace methods [3], SOR-type methods [4,5], Uzawa-type methods [6,7], HSS-type methods and its accelerated variants [8–10], and shift-splitting (denoted by SS) and generalized shift-splitting (denoted by GSS) iteration methods [11–13], and so forth.

In the past few years, many (GSS-)SS-type iteration methods have been proposed to greatly improve convergence rate by choosing the appropriate iteration parameters. Based on the theory of the shifting and clustered spectrum of the coefficient matrix, Bai and Zhang [12] presented a regularized conjugate gradient method for symmetric positive definite system. In [13], Bai et al. investigated a kind of shift-splitting iteration method for solving large and sparse non-Hermitian positive definite system. A modification of the SS preconditioner (i.e., the GSS preconditioner) has been established in [14, 15] with a symmetric positive definite (1,1)-block sub-matrix A for the positive iteration parameters α and β. Here, the GSS iteration method is defined as

$$\frac{1}{2}\begin{pmatrix} \alpha I_n + A & B \\ -B^T & \beta I_m \end{pmatrix} x^{k+1} = \frac{1}{2}\begin{pmatrix} \alpha I_n - A & -B \\ B^T & \beta I_m \end{pmatrix} x^k + \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.2}$$

where the related GSS preconditioner is given by

$$\mathcal{P}_{GSS} = \frac{1}{2}\begin{pmatrix} \alpha I_n + A & B \\ -B^T & \beta I_m \end{pmatrix}, \tag{1.3}$$

with I_q denoting the identity matrix of size q. Particularly, as $\alpha = \beta$, the GSS iteration method (1.2) reduces to the SS iteration method [16], where the SS preconditioner is given by the following form

$$\mathcal{P}_{SS} = \frac{1}{2}\begin{pmatrix} \alpha I_n + A & B \\ -B^T & \alpha I_m \end{pmatrix}. \tag{1.4}$$

For solving the saddle problem (1.1), Cao et al. [17] applied the GSS iteration method (1.2) to solve the saddle point system (1.1) and proposed a deteriorated shift-splitting (DSS) preconditioner by removing the shift parameter α in the $(1, 1)$-block of the GSS preconditioner. Based on the symmetric and skew-symmetric splitting $A = H + S$ with $S = \frac{1}{2}(A - A^T)$ of the $(1, 1)$-block sub-matrix A, Zhou et al. [18] proposed a modified shift-splitting (denoted by MSS) preconditioner of form

$$\mathcal{P}_{MSS} = \frac{1}{2}\begin{pmatrix} \alpha I_n + 2H & B \\ -B^T & \alpha I_m \end{pmatrix}, \tag{1.5}$$

and proposed a sufficient condition on the iteration parameter α for the convergence of the MSS iteration method. In [19], Huang et al. proposed a generalized modified shift-splitting iteration method (denoted by GMSS) by introducing two iteration parameters α and β instead of one parameter α, which further improve the convergence rate of the MSS iteration method, where the corresponding GMSS preconditioner is given by

$$\mathcal{P}_{GMSS} = \frac{1}{2}\begin{pmatrix} \alpha I_n + 2H & B \\ -B^T & \beta I_m \end{pmatrix}. \tag{1.6}$$

The aim of this paper is to propose an improvement for the (GSS-)SS-type iteration methods and investigate the SFHSS preconditioner for Krylov subspace methods, such as GMRES iteration method [20], which significantly accelerate the convergence speed of the related iterative method. According to theoretical analysis, when the iteration parameters α and β satisfy certain conditions, we prove that the SFHSS iteration method converges to the unique solution of the saddle point system (1.1). The rest of this paper is organized as follows. In Section 2, we review the SFHSS iteration method and its implementations. In Section 3, we analysis the convergent property of the SFHSS iteration method for nonsingular saddle point problems with nonsymmetric positive definite (1,1)-block. In Section 4, we discuss the algebraic properties of the resulted SFHSS preconditioned matrix. Finally, numerical experiments arise from finite element discretization of the Oseen problem are presented in Section 5 to show the correctness of the theoretical analysis and the feasibility of the SFHSS iteration method.

2. The SFHSS iteration method and its implementations

The main purpose of this section is to introduce the new GSS-type iteration method (i.e., the SFHSS iteration method) for the nonsingular saddle-point problems (1.1). Based on the local Hermitian and skew-Hermitian splitting scheme of the non-Hermitian positive definite (1,1)-block sub-matrix, Yang and Wu [6] presented the Uzawa-HSS iteration method. Let $A = H + S$ be the symmetric and skew-symmetric splitting of the (1,1)-block of the coefficient matrix $\mathcal{A}$ defined as in (1.1) with $S = \frac{1}{2}(A - A^T)$. Following the idea of the Uzawa-HSS iteration method, we make the following matrix splitting for the coefficient matrix $\mathcal{A}$ of the linear system (1.1):

$$\mathcal{A} = \mathcal{P}_{SFHSS} - \mathcal{N}_{SFHSS},$$

where the shift-HSS preconditioner $\mathcal{P}_{SFHSS}$ is given by

$$\mathcal{P}_{SFHSS} = \frac{1}{4}\begin{pmatrix} \frac{1}{\alpha}(\alpha I_n + 2H)(\alpha I_n + 2S) & 2B \\ -2B^T & \beta B^T B \end{pmatrix} \tag{2.1}$$

and the matrix $\mathcal{N}_{SFHSS}$ is defined as

$$\mathcal{N}_{SFHSS} = \frac{1}{4}\begin{pmatrix} \frac{1}{\alpha}(\alpha I_n - 2H)(\alpha I_n - 2S) & -2B \\ 2B^T & \beta B^T B \end{pmatrix}.$$

On the basis of the above splitting, we derive the following SFHSS iteration method

$$\mathcal{P}_{SFHSS} x^{k+1} = \mathcal{N}_{SFHSS} x^k + b, \tag{2.2}$$

and the generalized shift-HSS iteration matrix $\tau(\alpha,\beta)$ can be constructed as follows

$$\tau(\alpha,\beta) = \mathcal{P}_{SFHSS}^{-1}\mathcal{N}_{SFHSS}. \tag{2.3}$$

Denoted by

$$\Theta_1 = \frac{1}{\alpha}(\alpha I_n + 2H)(\alpha I_n + 2S),\ \ \Theta_2 = \frac{1}{\alpha}(\alpha I_n - 2H)(\alpha I_n - 2S),$$

$$\Phi_1 = \beta B^T B + 4B^T\Theta_1^{-1}B,\ \ \Phi_2 = \Theta_1 + 4B(\beta B^T B)^{-1}B^T.$$

More precisely, multiply both sides of (2.2) from the left by the matrices

$$\begin{pmatrix} I_n & -2B(\beta B^T B)^{-1} \\ 0 & I_m \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} I_n & 0 \\ 2B^T\Theta_1^{-1} & I_m \end{pmatrix},$$

respectively, then we can easily obtain the explicit formulations of $u^{(k+1)}$ and $v^{(k+1)}$, respectively. Therefore, the iteration scheme (2.2) naturally leads to the following algorithmic description of the SFHSS iteration method:

Method 2.1. (The SFHSS method) Given an initial guess $x^0 = [u^{(0)^T}, v^{(0)^T}]^T$, for $k = 0, 1, 2, \cdots$, until the iteration sequence $[u^{(k)^*}, v^{(k)^*}]^*$ converges,

(i) Compute $u^{(k)}$ from

$$u^{(k+1)} = u^{(k)} - 4\Phi_2^{-1}[A + 2B(\beta B^T B)^{-1}B^T]u^{(k)} - 4\Phi_2^{-1}Bv^{(k)} + 4\Phi_2^{-1}[f - 2B(\beta B^T B)^{-1}g],$$

(ii) Compute $v^{(k)}$ from

$$v^{(k+1)} = v^{(k)} + 2\Phi_1^{-1}B^T(I_n + \Theta_1^{-1}\Theta_2)u^{(k)} - 8\Phi_1^{-1}B^T\Theta_1^{-1}Bv^{(k)} + 4\Phi_1^{-1}(g + 2B^T\Theta_1^{-1}f).$$

Following the iteration scheme (2.2) again, it is not difficult to know that the SFHSS iteration method can be written as the following fixed point form

$$x^{k+1} = \tau(\alpha,\beta)x^k + 4\mathcal{P}_{SFHSS}^{-1}b. \tag{2.4}$$

It is easy to see that at each step of the SFHSS iteration method (2.4) or the preconditioned Krylov subspace method, such as GMRES iteration method, we need to solve a linear system $\mathcal{P}_{SFHSS}z = r$, i.e.,

$$\frac{1}{4}\begin{pmatrix} \frac{1}{\alpha}(\alpha I_n + 2H)(\alpha I_n + 2S) & 2B \\ -2B^T & \beta B^T B \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \tag{2.5}$$

where $z_1, r_1 \in \mathbb{R}^n$ and $z_2, r_2 \in \mathbb{R}^m$. A brisk calculation confirms that

$$\begin{aligned} \mathcal{P}_{SFHSS} = &\frac{1}{4}\begin{pmatrix} I_n & 2B(\beta B^T B)^{-1} \\ 0 & I_m \end{pmatrix} \\ &\times\begin{pmatrix} \Phi_2 & 0 \\ 0 & \beta B^T B \end{pmatrix}\begin{pmatrix} I_n & 0 \\ -2(\beta B^T B)^{-1}B^T & I_m \end{pmatrix}. \end{aligned} \tag{2.6}$$

From the decomposition of $\mathcal{P}_{SFHSS}$ in (4), it is obvious that

$$\begin{aligned}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} &= 4\begin{pmatrix} I_n & 0 \\ 2(\beta B^T B)^{-1}B^T & I_m \end{pmatrix}\begin{pmatrix} \Phi_2^{-1} & 0 \\ 0 & (\beta B^T B)^{-1} \end{pmatrix} \\ &\times \begin{pmatrix} I_n & -2B(\beta B^T B)^{-1} \\ 0 & I_m \end{pmatrix}\begin{pmatrix} r_1 \\ r_2 \end{pmatrix}.\end{aligned} \tag{2.7}$$

Then we have the following algorithmic implementation of the SFHSS iteration method.

Algorithm 2.1. For a given vector $(r_1^T, r_2^T)^T$, the vector $(z_1^T, z_2^T)^T$ can be derived from (2.7) by the following steps:

(1) solve $B^T Bt = \frac{1}{\beta} r_2$.

(2) solve $\Phi_2 z_1 = 4(r_1 - 2Bt)$;

(3) solve $B^T Bp = \frac{2}{\beta} B^T z_1$.

(4) compute $z_2 = p + 4t$.

Since $B^T B$ is symmetric positive definite and $(\alpha I_n + 2H)(\alpha I_n + 2S) + 4B(\beta B^T B)^{-1}B^T$ is nonsymmetric positive definite, then in practical implementations, the sub-linear systems $B^T Bz = r$ can be solved by the Cholesky factorization and $[\frac{1}{\alpha}(\alpha I_n + 2H)(\alpha I_n + 2S) + 4B(\beta B^T B)^{-1}B^T]z = r$ contained in the SFHSS iteration method can be solved inexactly by the sparse ILU factorization or gmres iteration method.

3. Convergence analysis of the SFHSS method

Consider the solution of a linear system of form $\mathcal{A}x = b$. To a matrix $\mathcal{A} \in \mathbb{R}^{(n+m)\times(n+m)}$, if $\mathcal{M}$ is nonsingular, then the representation $\mathcal{A} = \mathcal{M} - \mathcal{N}$ is called as a splitting. Assume that the iteration matrix $\mathcal{T} = \mathcal{M}^{-1}\mathcal{N}$ and $c = \mathcal{M}^{-1}b$, then the stationary iteration scheme of the linear system $\mathcal{A}x = b$ is defined as

$$x^{k+1} = \mathcal{T}x^k + c. \tag{3.1}$$

As is well known, if the spectral radius $\rho(\mathcal{T})$ of the iteration matrix $\mathcal{T}$ is less than one, then the stationary iteration scheme (3.1) converges to the unique solution of $\mathcal{A}x = b$, for any choice of the initial vector x^0.

Let λ be an eigenvalue of the iteration matrix $\tau(\alpha, \beta)$ in (2.3) and $x^T = (u^*, v^*)^*$ be the corresponding eigenvector. To study the convergence property of the generalized shift-HSS iteration method (2.4), we need to consider the following generalized eigenvalue problem:

$$\begin{aligned}&\begin{pmatrix} \frac{1}{\alpha}(\alpha I_n - 2H)(\alpha I_n - 2S) & -2B \\ 2B^T & \beta B^T B \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \\ &= \lambda \begin{pmatrix} \frac{1}{\alpha}(\alpha I_n + 2H)(\alpha I_n + 2S) & 2B \\ -2B^T & \beta B^T B \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix},\end{aligned} \tag{3.2}$$

By straightforward computation, the generalized eigenvalue problem (3.2) is equivalent to the following form

$$\begin{cases} \frac{1}{\alpha}(\alpha I_n - 2H)(\alpha I_n - 2S)u - 2Bv = \frac{1}{\alpha}\lambda(\alpha I_n + 2H)(\alpha I_n + 2S)u + 2\lambda Bv, \\ 2B^T u + \beta B^T Bv = -2\lambda B^T u + \beta\lambda B^T Bv. \end{cases} \tag{3.3}$$

For convenience, we denote

$$\frac{u^*Au}{u^*u} = a + bi, \frac{u^*HSu}{u^*u} = c + di \text{ and } \frac{u^*B(B^TB)^{-1}B^Tu}{u^*u} = e, \tag{3.4}$$

where $a > 0$, $e \geq 0$,

$$c = \frac{1}{2}\frac{u^*(HS - SH)u}{u^*u} \text{ and } d = \frac{1}{2i}\frac{u^*(HS + SH)u}{u^*u}.$$

In what follows, we give the following lemmas to verify the convergence of the SFHSS iteration method (2.4).

Lemma 3.1. ([21]). If S is a skew-Hermitian matrix, then iS is a Hermitian matrix and u^*Su is a purely imaginary number or zero for all $u \in C_n$.

Lemma 3.2. ([22]). Both roots of the complex quadratic equation $\lambda^2 - \Phi\lambda + \Psi = 0$ have modulus less than one if and only if $|\Phi - \overline{\Phi}\Psi| + |\Psi|^2 < 1$, where $\overline{\Phi}$ denotes the conjugate complex of Φ.

Lemma 3.3. Let A be nonsymmetric positive definite and B have full column rank. Assume λ is an eigenvalue of the iteration matrix $\tau(\alpha,\beta)$ defined as in (2.3) with $\beta > 0$, if $\alpha > 0$, then $\lambda \neq 1$, and if $\alpha^2 > 4|c|_{max}$ with $|c|_{max}$ denoting the maximum of c, then $\lambda \neq -1$.

Proof. Following the spirit of the proof of [17]. If $\lambda = 1$, then it is straightforward to show that the Eq (3.2) yield the following result

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Since A is nonsymmetric positive definite and B has full column rank, then we can easily know that $u = 0$ and $v = 0$. This is a contradiction as $(u^*, v^*)^*$ is an eigenvector.

If $\lambda = -1$, then the Eq (3.2) reduce to the following forms

$$\begin{cases} (\alpha^2 I_n + 4HS)u = 0, \\ \beta B^T Bv = 0. \end{cases}$$

If $\alpha^2 > 4|c|_{max}$, we can easily know that $\alpha^2 I_n + 4HS$ is nonsingular. Therefore, it is easy to see that $u = 0$ and $v = 0$. The result seems to contradict with $(u^*, v^*)^*$ being an eigenvector.

Thus, we complete the proof. □

Lemma 3.4. Let the conditions of Lemma 3.3 be satisfied. Assume that λ is an eigenvalue of the iteration matrix $\tau(\alpha,\beta)$ defined as in (2.3) and $(u^*, v^*)^*$ is the corresponding eigenvector with $u \in \mathbb{C}^n$ and $v \in \mathbb{C}^m$, if $0 \neq u \in \aleph(B^T)$, then $|\lambda| < 1$.

Proof. We demonstrate the verification of $u \neq 0$. Unless, if $u = 0$, then it follows from the second of the Eq (3.2) that $\alpha(\lambda - 1)B^T Bv = 0$. According to Lemma 3.3, since $\lambda \neq 1$, then $B^T Bv = 0$. As B has full column rank, then $B^T B$ is nonsingular. Therefore, we further conclude $v = 0$. This is a contradiction since $(u^*, v^*)^*$ is an eigenvector, so $u \neq 0$.
We now turn to verify $|\lambda| < 1$. Assume $u \in \aleph(B^T)$ with $\| u \|_2 = 1$, from the second of the Eq (3.2), we get $v = 0$. Following [8, Theorem 2.2] and multiplying the first of the Eq (3.2) from the left-hand side by u^* , it is obvious that

$$\begin{aligned}|\lambda| &= |u^*(\alpha I_n + 2S)^{-1}(\alpha I_n + 2H)^{-1}(\alpha I_n - 2H)(\alpha I_n - 2S)u| \\ &\leq \| (\alpha I_n + 2S)^{-1}(\alpha I_n + 2H)^{-1}(\alpha I_n - 2H)(\alpha I_n - 2S) \|_2 \\ &\leq \| (\alpha I_n - 2H)(\alpha I_n + 2H)^{-1} \|_2 \\ &= \max_i \left|\frac{\alpha - 2\lambda_i(H)}{\alpha + 2\lambda_i(H)}\right| \\ &< 1,\end{aligned}$$

where $\lambda_i(H)$ denotes the ith eigenvalue of the symmetric positive definite matrix H.
Therefore, the proof is completed. □

Theorem 3.1. Let the conditions of Lemma 3.3 be satisfied. Assume that λ is an eigenvalue of the iteration matrix $\tau(\alpha,\beta)$ defined as in (2.3), if the positive iteration parameters α and β satisfy the following conditions

If $ac + bd \geq 0$, then

$$\alpha^2 > 4|c|_{max} \text{ and } \beta > \frac{16d^2 e}{\alpha^3 a^2},$$

and if $ac + bd < 0$, then

$$\alpha^2 > max\left\{\frac{4|ac + bd|_{max}}{a}, 4|c|_{max}\right\} \text{ and } \beta > \frac{16d^2 e}{\alpha a[\alpha^2 a - |4ac + 4bd|]}.$$

then the iteration method (2.4) converges to the unique solution of the nonsymmetric saddle point problem (1.1), i.e.,

$$|\lambda| < 1.$$

Proof. Combine Lemmas 3.3 and 3.4, in order to complete the proof, we need only to verify the case $B^T u \neq 0$. Suppose $u \notin \aleph(B^T)$, then the second of the Eq (3.2) yields the following result

$$v = \frac{2(\lambda + 1)}{\beta(\lambda - 1)}(B^T B)^{-1} B^T u. \tag{3.5}$$

By substituting the relationship (3.5) into the first of the Eq (3.2), we have

$$(\alpha I_n - 2H)(\alpha I_n - 2S)u = \lambda(\alpha I_n + 2H)(\alpha I_n + 2S)u + \frac{4\alpha(\lambda + 1)^2}{\beta(\lambda - 1)} B(B^T B)^{-1} B^T u. \tag{3.6}$$

Multiplying the Eq (3.6) from the left-hand side by u^*, after straightforward calculations, then the Eq (3.6) yields the following form

$$\begin{aligned}\alpha^2\beta(\lambda - 1)^2 \quad &+ 2\alpha\beta(\lambda^2 - 1)\frac{u^* A u}{u^* u} + 4\beta(\lambda - 1)^2 \frac{u^* HS u}{u^* u} \\ &+ 4\alpha(\lambda + 1)^2 \frac{u^* B(B^T B)^{-1} B^T u}{u^* u} = 0.\end{aligned} \tag{3.7}$$

Following (3.4), a quadratic equation of λ is derived from the Eq (3.7). After some algebra, it is straightforward to show that

$$\begin{aligned}&[(\alpha^2+2\alpha a+4c)\beta+4\alpha e+2(\alpha b+2d)\beta i]\lambda^2+2(4\alpha e-\alpha^2\beta-4\beta c-4\beta di)\lambda\\&\quad+\alpha^2\beta-2\alpha\beta a+4\beta c+4\alpha e+2\beta(2d-\alpha b)i=0.\end{aligned}\tag{3.8}$$

If $(\alpha^2+2\alpha a+4c)\beta+4\alpha e+2(\alpha b+2d)\beta i=0$, then it is easy to see that $(\alpha^2+2\alpha a+4c)\beta+4\alpha e=0$ and $\alpha b+2d=0$. Therefore, the Eq (3.8) yields

$$\begin{aligned}\lambda&=-\frac{\alpha^2\beta-2\alpha\beta a+4\beta c+4\alpha e+2\beta(2d-\alpha b)i}{2(4\alpha e-\alpha^2\beta-4\beta c-4\beta di)}\\&=\frac{\beta a+\beta bi}{4e+\beta a+\beta bi}.\end{aligned}$$

By Lemma 3.3, we get $\lambda\neq\pm1$ as $\alpha^2>4|c|_{max}$. Note that $a>0$ and $e\geq 0$, we have

$$|\lambda|=\sqrt{\frac{(\beta a)^2+(\beta b)^2}{(4e+\beta a)^2+(\beta b)^2}}<1.$$

In what follows, we consider the case $(\alpha^2+2\alpha a+4c)\beta+4\alpha e+2(\alpha b+2d)\beta i\neq 0$. From lemma 3.2, we know that $|\lambda|<1$ if and only if $|\Phi-\overline{\Phi}\Psi|+|\Psi|^2<1$. For convenience, we denote Φ and Ψ by

$$\Phi=\frac{2(4\alpha e-\alpha^2\beta-4\beta c-4\beta di)}{(\alpha^2+2\alpha a+4c)\beta+4\alpha e+2(\alpha b+2d)\beta i}$$

and

$$\Psi=\frac{\alpha^2\beta-2\alpha\beta a+4\beta c+4\alpha e+2\beta(2d-\alpha b)i}{(\alpha^2+2\alpha a+4c)\beta+4\alpha e+2(\alpha b+2d)\beta i}.$$

After straightforward computation, we have

$$|\Phi-\overline{\Phi}\Psi|+|\Psi|^2=\frac{8\alpha\beta\sqrt{\Gamma+(16de)^2}+\Upsilon+4\beta^2(2d-\alpha b)^2}{(\alpha^2\beta+2\alpha\beta a+4\beta c+4\alpha e)^2+4\beta^2(2d+\alpha b)^2},$$

where $\Gamma=(4\alpha ae-\alpha^2\beta a-4\beta ac-4\beta bd)^2$ and $\Upsilon=(\alpha^2\beta-2\alpha\beta a+4\beta c+4\alpha e)^2$.

The following inequality

$$\begin{aligned}&|\Phi-\overline{\Phi}\Psi|+|\Psi|^2\\&<\frac{8\alpha\beta\sqrt{\Gamma+16\alpha ae(\alpha^2\beta a+4\beta ac+4\beta bd)}+\Upsilon+4\beta^2(2d-\alpha b)^2}{(\alpha^2\beta+2\alpha\beta a+4\beta c+4\alpha e)^2+4\beta^2(2d+\alpha b)^2}\\&=\frac{8\alpha\beta(4\alpha ae+\alpha^2\beta a+4\beta ac+4\beta bd)+\Upsilon+4\beta^2(2d-\alpha b)^2}{(\alpha^2\beta+2\alpha\beta a+4\beta c+4\alpha e)^2+4\beta^2(2d+\alpha b)^2}\\&=1\end{aligned}$$

holds true for this case

$$16\alpha ae(\alpha^2\beta a + 4\beta ac + 4\beta bd) > (16de)^2. \tag{3.9}$$

This implies

$$\alpha^2 a + 4ac + 4bd > 0. \tag{3.10}$$

Following the inequalities (3.9) and (3.10), if $ac + bd \geq 0$, we have

$$\alpha^2 > 0 > -\frac{4(ac + bd)}{a},$$

and

$$\beta > \frac{16d^2e}{\alpha^3a^2} \geq \frac{16d^2e}{\alpha a[\alpha^2 a + (4ac + 4bd)]}.$$

If $ac + bd < 0$, then we get

$$\alpha^2 > \frac{4|ac + bd|_{max}}{a} \geq -\frac{4(ac + bd)}{a} > 0,$$

and

$$\beta > \frac{16d^2e}{\alpha a[\alpha^2 a - |4ac + 4bd|]} \geq \frac{16d^2e}{\alpha a[\alpha^2 a + (4ac + 4bd)]}.$$

By making use of $\alpha^2 > 4|c|_{max}$, we complete the proof. □

For the sake of convenience, we denote the maxima of $|bd|$, $|d|$ and e by $|bd|_{max}$, $|d|_{max}$ and e_{max}, respectively, and denote the minimums of a, b, c and d by a_{min}, b_{min}, c_{min}, and d_{min}, respectively. According to Theorem 3.1, we give the following sufficient conditions for the convergence of the SFHSS iteration method (2.4).

Corollary 3.1. Let the conditions of Theorem 3.1 be satisfied. If the positive iteration parameters α and β satisfy:

If $ac + bd \geq 0$, then

$$\alpha > 2\sqrt{|c|_{max}}, \text{ and } \beta > \frac{16|d|^2_{max}e_{max}}{\alpha^3 a^2_{min}}.$$

If $ac + bd < 0$, then

$$\alpha^2 > 2\sqrt{|c|_{max} + \frac{|bd|_{max}}{a_{min}}},$$

and

$$\beta > \frac{16d^2e}{\alpha a_{min}[\alpha^2 a_{min} - 4|ac + bd|_{max}]}.$$

Then the generalized shift-HSS iteration method (2.4) converges to the unique solution of the nonsymmetric saddle point problem (1.1).

Proof. According to Theorem 3.1, if $ac + bd \geq 0$, then

$$\alpha^2 > 4|c|_{max}, \ \beta > \frac{16|d|^2_{max}e_{max}}{\alpha^3 a^2_{min}} \geq \frac{16d^2e}{\alpha^3a^2},$$

and if $ac + bd < 0$, then

$$\alpha^2 > 4|c|_{max} + \frac{|bd|_{max}}{a} \geq max\left\{\frac{4|ac + bd|_{max}}{a}, 4|c|_{max}\right\},$$

and

$$\beta > \frac{16d^2 e}{\alpha a_{min}[\alpha^2 a_{min} - |4ac + 4bd|_{max}]} \geq \frac{16d^2 e}{\alpha a[\alpha^2 a - (4ac + 4bd)]}.$$

Therefore, we complete the proof. □

4. The spectral properties of the preconditioned matrix

The main objective of this section is to introduce some elegant inclusion regions for the spectrum of $\mathcal{P}_{SFHSS}^{-1}\mathcal{A}$ for the saddle point problem (1.1).

In the following, to derive some related bounds of the eigenvalues of the preconditioned saddle point matrix $\mathcal{P}_{SFHSS}^{-1}\mathcal{A}$, we study the eigenvalue problem $\mathcal{P}_{SFHSS}^{-1}\mathcal{A}x = \eta x$, that is to say

$$\mathcal{A}x = \eta \mathcal{P}_{SFHSS}x, \tag{4.1}$$

where η denotes an any eigenvalue of the preconditioned matrix $\mathcal{P}_{SFHSS}^{-1}\mathcal{A}$ with the corresponding eigenvector $x = (u^*, v^*)^*$.

For simplicity, we denote $v^* B^T Bv$ by σ^2 and the null space of B^T by $\aleph(B^T)$, at the same time, the matrix $\mathcal{R}_{SFHSS}$ is defined by

$$\mathcal{R}_{SFHSS} = \frac{1}{4}\begin{pmatrix} \alpha I_n + \frac{4}{\alpha}HS & 0 \\ 0 & \beta B^T B \end{pmatrix},$$

then it is easy to see that

$$\mathcal{P}_{SFHSS} = \mathcal{R}_{SFHSS} + \frac{1}{2}\mathcal{A}. \tag{4.2}$$

After some algebra, we can rewrite the generalized eigenvalue problem (4.1) as

$$(1 - \frac{\eta}{2})\mathcal{A}x = \eta \mathcal{R}_{SFHSS}x. \tag{4.3}$$

Following Lemma 3.3, since the eigenvalue $\lambda = 1 - \eta$ of $\tau(\alpha,\beta)$ satisfies $\lambda \neq -1$ with $\alpha^2 > 4|c|_{max}$, then $\eta = 1 - \lambda \neq 2$. So, $1 - \frac{1}{2}\eta \neq 0$, we set

$$\theta = \frac{2\eta}{2-\eta}, \quad \text{for which} \quad \eta = \frac{2\theta}{\theta + 2} = 2 - \frac{4}{\theta + 2}.$$

For convenience, we use $\Re(\theta)$ and $\Im(\theta)$ to denote the real part and image part of the eigenvalue θ, respectively.

We can explicitly write the equivalent eigenproblem $\mathcal{A}x = \theta \mathcal{R}_{SFHSS}x$ as

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \frac{\theta}{4}\begin{pmatrix} \alpha I_n + \frac{4}{\alpha}HS & 0 \\ 0 & \beta B^T B \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}. \tag{4.4}$$

The equivalent results to Eq (4.4) are given by

$$\begin{cases} Au + Bv = \theta(\frac{\alpha}{4}I_n + \frac{1}{\alpha}HS)u, \\ -B^T u = \frac{1}{4}\beta\theta B^T Bv. \end{cases} \tag{4.5}$$

It is obvious that $u \neq 0$, otherwise the second equation of (4.5) would implies $\theta = 0$ or $v = 0$. However, from Lemma 3.3, neither of them can be satisfied. So, $u \neq 0$. If $v = 0$ and $\alpha^2 > 4|c|_{max}$, then

Theorem 4.1. Let the conditions of Lemma 3.3 be satisfied. Assume (η, x) is an eigenpair of (4.1) with $x = (u^*, v^*)^*$ and $\| u \|_2 = 1$. Then for $\forall\alpha, \beta > 0$, the eigenvalue η can be written as $\eta = \frac{2\theta}{\theta + 2}$, where θ satisfies the following:

(i) If $v = 0$, then $u \in \aleph(B^T)$ and

$$\frac{4\alpha|\lambda(A)|_{min}}{\sqrt{\alpha^2 + 8\alpha c_{max} + 16\rho^2(HS)}} \leq |\theta| \leq \frac{4\alpha\rho(A)}{\sqrt{\alpha^2 + 8\alpha c_{min} + 16|\lambda(HS)|^2_{min}}}, \tag{4.6}$$

with

$$\Re(\theta) = \frac{4\alpha(\alpha^2 a + 4ac + 4bd)}{(\alpha^2 + 4c)^2 + 16d^2} \quad \text{and} \quad \Im(\theta) = \frac{4\alpha(\alpha^2 b + 4bc - 4ad)}{(\alpha^2 + 4c)^2 + 16d^2}. \tag{4.7}$$

(ii) If $v \neq 0$, then $u \notin \aleph(B^T)$ and

$$\frac{4\alpha(\alpha^2 + 4c_{min})|\lambda(A)|_{min}}{\Theta(\alpha,\beta)} < |\theta| \leq \frac{4\alpha\rho(A)\Pi(\alpha,\beta)}{\Upsilon(\alpha,\beta)}. \tag{4.8}$$

with

$$\Re(\theta) = \frac{4\alpha(\alpha^2 a + 4ac + 4bd - \alpha\beta a\sigma^2)}{(\alpha^2 + 4c)^2 - (\alpha\beta\sigma^2)^2 + 16d^2}, \; \Im(\theta) = \frac{4\alpha(\alpha^2 b + 4bc + \alpha\beta b\sigma^2 - 4ad)}{(\alpha^2 + 4c)^2 - (\alpha\beta\sigma^2)^2 + 16d^2}, \tag{4.9}$$

and

$$\begin{aligned} \Pi(\alpha,\beta) &= \alpha^2(\alpha^2 + 8c_{max}) + 16\rho^2(HS) + 2(\alpha^2 + 4c_{max})\alpha\beta\sigma^2_{max} + (\alpha\beta\sigma^2_{max})^2, \\ \omega_1(\alpha,\beta) &= \alpha^2(\alpha^2 + 8c_{min}) + 16|\lambda(HS)|^2_{min} - (\alpha\beta\sigma^2_{max})^2, \end{aligned}$$

$$\begin{aligned} \omega_2(\alpha,\beta) &= (\alpha\beta\sigma^2_{min})^2 - \alpha^2(\alpha^2 + 8c_{max}) - 16\rho^2(HS), \\ \Upsilon(\alpha,\beta) &= min\{\omega_1(\alpha,\beta), \omega_2(\alpha,\beta)\}, \\ \Theta(\alpha,\beta) &= \alpha^2(\alpha^2 + 8c_{min}) + 16\rho^2(HS) + (\alpha\beta\sigma^2_{max})^2. \end{aligned}$$

Proof. In order to obtain the inequalities (4.6) and (4.8), we need to consider two cases: (i) $v = 0$, (ii) $v \neq 0$.

We now turn to verify (i). If $v = 0$, from (4.5), it is easy to see that

$$Au = \theta(\frac{\alpha}{4}I_n + \frac{1}{\alpha}HS)u. \tag{4.10}$$

Multiplying u^* to the two sides of (4.10) from left, it then from (3.4) that

$$\begin{aligned}\theta &= \frac{4\alpha(a+bi)}{(\alpha^2+4c)+4di} \\ &= \frac{4\alpha(\alpha^2 a+4ac+4bd)+4\alpha(\alpha^2 b+4bc-4ad)i}{(\alpha^2+4c)^2+16d^2}.\end{aligned} \tag{4.11}$$

Consequently, we obtain (4.7). After some algebra, it is straightforward to show that

$$|\theta| = \sqrt{\Re^2(\theta)+\Im^2(\theta)} = \frac{4\alpha\sqrt{a^2+b^2}}{\sqrt{(\alpha^2+4c)^2+16d^2}}. \tag{4.12}$$

By straightforward calculation, we can get the inequality (4.6).

We demonstrate the validity of (ii). If $v \neq 0$, multiplying by u^* from left, then the first of the Eq (4.5) yields

$$u^*Au + u^*Bv = \theta u^*(\frac{\alpha}{4}I_n + \frac{1}{\alpha}HS)u. \tag{4.13}$$

Multiplying the transposed conjugate of the second of Eq (4.5) by v^*, we get

$$u^*Bv = -\frac{1}{4}\beta\bar{\theta}v^*B^TBv. \tag{4.14}$$

Substituting (4.14) into (4.13), we obtain

$$4\alpha u^*Au = \alpha^2\theta + 4\theta u^*HS\,u + \alpha\beta\bar{\theta}v^*B^TBv. \tag{4.15}$$

Following the above notes, it will be shown that

$$4\alpha(a+bi) = \alpha^2\theta + 4\theta(c+di) + \alpha\beta\bar{\theta}\sigma^2. \tag{4.16}$$

It is obvious to obtain that

$$(\alpha^2+4c+\alpha\beta\sigma^2)\Re(\theta) - 4d\Im(\theta) = 4\alpha a,$$

and

$$(\alpha^2+4c-\alpha\beta\sigma^2)\Im(\theta) + 4d\Re(\theta) = 4\alpha b.$$

Through direct calculations, we get (4.9) and

$$|\theta| = \sqrt{\Re^2(\theta)+\Im^2(\theta)} = \frac{4\alpha\sqrt{\varphi(\alpha,\beta)+\psi(\alpha,\beta)}}{(\alpha^2+4c)^2-(\alpha\beta\sigma^2)^2+(4d)^2}, \tag{4.17}$$

where

$$\varphi(\alpha,\beta) = (a^2+b^2)[(\alpha^2+4c)^2+(4d)^2+(\alpha\beta\sigma^2)^2],$$

and

$$\psi(\alpha,\beta) = 2[(b^2 - a^2)(\alpha^2 + 4c) - 8abd]\alpha\beta\sigma^2.$$

Since $\alpha^2 a + 4ac + 4bd > 0$, then we have

$$\begin{aligned}
&\varphi(\alpha,\beta) + \psi(\alpha,\beta) \\
&\quad < \varphi(\alpha,\beta) + 2(b^2 - a^2)(\alpha^2 + 4c)\alpha\beta\sigma^2 + 4\alpha\beta\sigma^2 a^2(\alpha^2 + 4c) \\
&\quad = \varphi(\alpha,\beta) + 2(b^2 + a^2)(\alpha^2 + 4c)\alpha\beta\sigma^2 \\
&\quad = (a^2 + b^2)[(\alpha^2 + 4c + \alpha\beta\sigma^2)^2 + (4d)^2].
\end{aligned} \tag{4.18}$$

Consider

$$(\alpha^2 + 4c)^2 + (4d)^2 > (\alpha\beta\sigma^2)^2,$$

or

$$(\alpha^2 + 4c)^2 + (4d)^2 < (\alpha\beta\sigma^2)^2.$$

By straightforward computation, we can find that

$$\begin{aligned}
|\theta| &< \frac{\sqrt{(a^2 + b^2)[(\alpha^2 + 4c + \alpha\beta\sigma^2)^2 + (4d)^2]}}{|(\alpha^2 + 4c)^2 - (\alpha\beta\sigma^2)^2 + 16d^2|} \\
&< \frac{4\alpha\sqrt{(a^2 + b^2)\Pi(\alpha,\beta)}}{\Upsilon(\alpha,\beta)}.
\end{aligned} \tag{4.19}$$

Additionally, as $a \geq b$, use $\alpha^2 a + 4ac + 4bd > 0$ again, we have

$$\begin{aligned}
&\varphi(\alpha,\beta) + \psi(\alpha,\beta) \\
&\quad > \varphi(\alpha,\beta) + 2\left[(b^2 - a^2)\left(-\frac{4bd}{a}\right) - 8abd\right]\alpha\beta\sigma^2 \\
&\quad = (a^2 + b^2)\left[(\alpha^2 + 4c)^2 + (4d)^2 + (\alpha\beta\sigma^2)^2 - 8\left(\frac{b}{a}\right)d\alpha\beta\sigma^2\right] \\
&\quad \geq (a^2 + b^2)\left[(\alpha^2 + 4c)^2 + (4d)^2 + (\alpha\beta\sigma^2)^2 - 8d\alpha\beta\sigma^2\right] \\
&\quad \geq (a^2 + b^2)(\alpha^2 + 4c)^2.
\end{aligned} \tag{4.20}$$

As $a < b$, then we obtain

$$\begin{aligned}
&\varphi(\alpha,\beta) + \psi(\alpha,\beta) \\
&\quad > \varphi(\alpha,\beta) - 16a^2 d\alpha\beta\sigma^2 \\
&\quad > (a^2 + b^2)(\alpha^2 + 4c)^2 + 2a^2[(4d)^2 + (\alpha\beta\sigma^2)^2 - 8d\alpha\beta\sigma^2] \\
&\quad > (a^2 + b^2)(\alpha^2 + 4c)^2.
\end{aligned} \tag{4.21}$$

Hence, combine (4.20) and (4.21), we have

$$\begin{aligned}
|\theta| &> \frac{\sqrt{(a^2 + b^2)}(\alpha^2 + 4c)}{(\alpha^2 + 4c)^2 + (\alpha\beta\sigma^2)^2 + (4d)^2} \\
&\geq \frac{4\alpha|\lambda(A)|_{min}(\alpha^2 + 4c_{min})}{\Theta(\alpha,\beta)}.
\end{aligned} \tag{4.22}$$

Hence, we complete the proof. □

Remark 4.1. Following Theorem 4.1, the eigenvalue θ satisfies two cases:

(i) If $v = 0$ and $\alpha^2 b + 4bc = 4ad$, then $u \in \aleph(B^T)$ and the real eigenvalue

$$\theta = \frac{a}{\alpha^2 + 4c} > 0$$

is bounded by

$$\frac{a_{min}}{\alpha^2 + 4c_{max}} < \theta < \frac{a_{max}}{\alpha^2 + 4c_{min}}.$$

(ii) If $v \neq 0$ and $\alpha^2 b + 4bc + \alpha\beta b\sigma^2 = 4ad$, then $u \notin \aleph(B^T)$ and the real eigenvalue

$$\theta = \frac{a}{\alpha^2 + 4c + \alpha\beta\sigma^2} > 0$$

is bounded as

$$\frac{a_{min}}{\alpha^2 + 4c_{max} + \alpha\beta\sigma^2_{max}} < \theta < \frac{a_{max}}{\alpha^2 + 4c_{min} + \alpha\beta\sigma^2_{min}}. \quad \square$$

Theorem 4.2. Let the conditions of Theorem 4.1 be satisfied. For any iteration parameters $\alpha, \beta > 0$, then the eigenvalue η of the SFHSS preconditioned matrix $\mathcal{P}^{-1}_{SFHSS}\mathcal{A}$ satisfies

$$\frac{2|\theta|_{min}}{|\theta|_{min} + 2} \leq |\eta| < \frac{2|\theta|_{max}}{\sqrt{|\theta|^2_{max} + 4}}. \tag{4.23}$$

Proof. For any iteration parameters $\alpha, \beta > 0$, since $|\theta| = \sqrt{\Re^2(\theta) + \Im^2(\theta)}$, then we get $0 < \Re(\theta) \leq |\theta|$. From $\eta = \frac{2\theta}{\theta + 2}$, it is easy to see that

$$|\eta| = \frac{2|\theta|}{\sqrt{|\theta|^2 + 4\Re(\theta) + 4}}.$$

Hence, we have

$$\frac{2|\theta|}{|\theta| + 2} \leq |\eta| < \frac{2|\theta|}{\sqrt{|\theta|^2 + 4}}.$$

Together the monotone properties of $\frac{2|\theta|}{|\theta| + 2}$ and $\frac{2|\theta|}{\sqrt{|\theta|^2 + 4}}$ with respect to $|\theta|$, we complete the proof of Theorem 4.2. □

Combine Theorems 4.1 and 4.2, we can find the following result.

Remark 4.2. Combine Theorems 4.1 and 4.2, for $\forall \alpha, \beta > 0$, some refined bounds for the eigenvalue η of the SFHSS preconditioned matrix $\mathcal{P}^{-1}_{SFHSS}\mathcal{A}$ are given by

(i) If $v = 0$, then $u \in \aleph(B^T)$ and

$$\frac{4\alpha|\lambda(A)|_{min}}{2\alpha|\lambda(A)|_{min} + \sqrt{\alpha^2 + 8\alpha c_{max} + 16\rho^2(HS)}} \leq |\eta| \leq \frac{4\alpha\rho(A)}{\mathcal{F}(\alpha)},$$

where

$$\mathcal{F}(\alpha) = \sqrt{4\alpha^2\rho^2(A) + \alpha^2 + 8\alpha c_{min} + 16|\lambda(HS)|^2_{min}}.$$

(ii) If $v \neq 0$, then

$$\frac{4\alpha|\lambda(A)|_{min}(\alpha^2 + 4c_{min})}{2\alpha|\lambda(A)|_{min}(\alpha^2 + 4c_{min}) + \Theta(\alpha,\beta)} < |\theta| \leq \frac{4\alpha\rho(A)\Pi(\alpha,\beta)}{\sqrt{4[\alpha\rho(A)\Pi(\alpha,\beta)]^2 + \Upsilon^2(\alpha,\beta)}}. \quad \square$$

Remark 4.3. Since $\eta = \dfrac{2\theta}{\theta + 2}$, by simple algebra, it is easy to see that η is real if and only if $\Im(\theta) = 0$. Following Remark 4.1, the real eigenvalue η satisfies two cases of forms:

(i) If $v = 0$ and $\alpha^2 b + 4bc = 4ad$, then $u \in \aleph(B^T)$ and the real eigenvalue

$$\eta = \frac{2a}{a + 2\alpha^2 + 8c} > 0$$

meets the following inequality

$$\frac{2a_{min}}{a_{min} + 2\alpha^2 + 8c_{max}} < \eta < \frac{2a_{max}}{a_{max} + 2\alpha^2 + 8c_{min}}.$$

(ii) If $v \neq 0$ and $\alpha^2 b + 4bc + \alpha\beta b\sigma^2 = 4ad$, then $u \notin \aleph(B^T)$ and the real eigenvalue

$$\eta = \frac{2a}{a + 2\alpha^2 + 8c + 2\alpha\beta\sigma^2} > 0$$

is bounded by

$$\frac{2a_{min}}{a_{min} + 2\alpha^2 + 8c_{max} + 2\alpha\beta\sigma^2_{max}} < \eta < \frac{2a_{max}}{a_{max} + 2\alpha^2 + 8c_{min} + 2\alpha\beta\sigma^2_{min}}. \quad \square$$

In the following, based on the above descriptions, I will further discuss the algebraic properties of the preconditioned matrix $\mathcal{P}^{-1}_{SFHSS}\mathcal{A}$, where the preconditioner $\mathcal{P}_{\alpha,0}$ is a reduced form of (2.1) with $\beta = 0$. For simplicity of description, we denote $\mathcal{P}_{SFHSS}$ and $\mathcal{N}_{SFHSS}$ with $\alpha^2 > 4|c|_{max}$ and $\beta = 0$ by $\mathcal{P}_{\alpha,0}$ and $\mathcal{N}_{\alpha,0}$, respectively. For more details on the the algebraic properties of the preconditioned matrix, we refer to [23–25].

Theorem 4.3. Let A be nonsymmetric positive definite and B have full column rank. Then the preconditioned saddle point matrix $\mathcal{P}^{-1}_{\alpha,0}\mathcal{A}$ has an eigenvalue $\eta = 2$ with algebraic multiplicity at least m and the remaining eigenvalues are $\eta_j = \dfrac{4\alpha[\lambda_j(H) + \lambda_j(S)]}{[\alpha + 2\lambda_j(H)][\alpha + 2\lambda_j(S)]}(j = 1, 2, \ldots, n)$, where $\lambda_j(H)$ and $\lambda_j(S)$ denote the jth eigenvalue of H and S, respectively.

Proof. If $\beta = 0$, according to the second of the Eq (3.2), we can obtain

$$(\lambda + 1)B^T u = 0,$$

then we further get either $\lambda + 1 = 0$ or $B^T u = 0$. If $\lambda = -1$, i.e. $\eta = 1 - \lambda = 2$, by the first of the Eq (3.2), we have

$$(\alpha^2 I_n + 4HS)u = 0.$$

Since $\alpha^2 I_n + 4HS$ is nonsingular with $\alpha^2 > 4|c|_{max}$, then it is easy to obtain the related eigenvectors have the form of $\begin{bmatrix} 0 \\ v_l^T \end{bmatrix}$ $(l = 1, 2, \ldots, m)$. If $B^T u = 0$, use the first of the Eq (3.2) again, we obtain

$$\begin{aligned} \lambda &= \frac{u * (\alpha I_n - 2H)(\alpha I_n - 2S)u}{u * (\alpha I_n + 2H)(\alpha I_n + 2S)u} \\ &= \frac{[\alpha - 2\lambda(H)][\alpha - 2\lambda(S)]}{[\alpha + 2\lambda(H)][\alpha I_n + 2\lambda(S)]}, \end{aligned}$$

therefore, we further get

$$\eta = 1 - \lambda = \frac{4\alpha[\lambda(H) + \lambda(S)]}{[\alpha + 2\lambda(H)][\alpha + 2\lambda(S)]}.$$

Hence, we complete the proof of Theorem 4.3. □

Remark 4.4. Following Theorem 4.3, it is easy to know that the preconditioned matrix $\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}$ has $m + j(1 \le j \le n)$ linearly independent eigenvectors, where

(i) m linearly independent eigenvectors related to the eigenvalue 2 have the form of $\begin{bmatrix} 0 \\ v_l^T \end{bmatrix}$ $(l = 1, 2, \ldots, m)$.

(ii) $j(j = 1, 2, \ldots, n)$ linearly independent eigenvectors associated with eigenvalues unequal to 2 have the form $\begin{bmatrix} u_s^T \\ v_s^T \end{bmatrix}$ $(s = 1, 2, \ldots, j)$ with $B^T u = 0$. □

In what follows, we devote to study the properties of the minimal polynomial for the preconditioned matrix $\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}$. which are beneficial to the Krylov subspace acceleration. To derive an expression for the corresponding characteristic polynomial of $\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}$, we decompose once again the preconditioner $\mathcal{P}_{\alpha,0}$ as

$$\mathcal{P}_{\alpha,0} = \frac{1}{4}\begin{pmatrix} I_n & 0 \\ -2B^T\mathcal{F}^{-1} & I_m \end{pmatrix}\begin{pmatrix} \mathcal{F} & 0 \\ 0 & 4B^T\mathcal{F}^{-1}B \end{pmatrix}\begin{pmatrix} I_n & 2\mathcal{F}^{-1}B \\ 0 & I_m \end{pmatrix},$$

where

$$\mathcal{F} = \frac{1}{\alpha}(\alpha I_n + 2H)(\alpha I_n + 2S).$$

It is obvious that

$$\mathcal{P}_{\alpha,0}^{-1} = 4\begin{pmatrix} I_n & -2\mathcal{F}^{-1}B \\ 0 & I_m \end{pmatrix}\begin{pmatrix} \mathcal{F}^{-1} & 0 \\ 0 & (4B^T\mathcal{F}^{-1}B)^{-1} \end{pmatrix}\begin{pmatrix} I_n & 0 \\ 2B^T\mathcal{F}^{-1} & I_m \end{pmatrix}.$$

A simple computation reveals that

$$\begin{aligned} \mathcal{P}_{\alpha,0}^{-1}\mathcal{A} &= I_{n+m} - \mathcal{P}_{\alpha,0}^{-1}\mathcal{N}_{\alpha,0} \\ &= I_{n+m} - \begin{pmatrix} I_n & -2\mathcal{F}^{-1}B \\ 0 & I_m \end{pmatrix}\begin{pmatrix} \mathcal{F}^{-1} & 0 \\ 0 & (4B^T\mathcal{F}^{-1}B)^{-1} \end{pmatrix} \\ &\times \begin{pmatrix} I_n & 0 \\ 2B^T\mathcal{F}^{-1} & I_m \end{pmatrix}\begin{pmatrix} \mathcal{G} & -2B \\ 2B^T & 0 \end{pmatrix} \\ &= I_{n+m} - \begin{pmatrix} \mathcal{F}^{-1}\mathcal{G} - 2\mathcal{F}^{-1}B\mathcal{D} & 0 \\ \mathcal{D} & -I_m \end{pmatrix} \\ &= \begin{pmatrix} I_n - \mathcal{F}^{-1}\mathcal{G} + \mathcal{F}^{-1}B\mathcal{D} & 0 \\ -\mathcal{D} & 2I_m \end{pmatrix}, \end{aligned}$$

where

$$\mathcal{G} = \frac{1}{\alpha}(\alpha I_n - 2H)(\alpha I_n - 2S),$$

and

$$D = (B^T\mathcal{F}^{-1}B)^{-1}(B^T\mathcal{F}^{-1}\mathcal{G} + B^T).$$

Since $B^T u = 0$, then for $j = 1, 2, \ldots, n$, we can get

$$\eta_j(I_n - \mathcal{F}^{-1}\mathcal{G} + 2\mathcal{F}^{-1}B\mathcal{D}) = \eta_j(I_n - \mathcal{F}^{-1}\mathcal{G}) = \frac{4\alpha[\lambda_j(H) + \lambda_j(S)]}{[\alpha + 2\lambda_j(H)][\alpha + 2\lambda_j(S)]}.$$

Then the characteristic polynomial of $\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}$ is described as follows

$$\Phi_{\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}} = det(\eta I_{n+m} - \mathcal{P}_{\alpha,0}^{-1}\mathcal{A}) = (\eta - 2)^m \prod_{j=1}^{n}(\eta - \eta_j).$$

Denote by

$$\Psi(\eta) = (\eta - 2)^m \prod_{j=1}^{n}(\eta - \eta_j).$$

It is straightforward to show that $\Psi(\eta)$ is a polynomial related to η of degree $n + 1$. Then a simple computation reveals that

$$\begin{aligned}\Psi(\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}) &= (\mathcal{P}_{\alpha,0}^{-1}\mathcal{A} - 2I_{n+m})^m \prod_{j=1}^{n}(\mathcal{P}_{\alpha,0}^{-1}\mathcal{A} - \eta_j I_{n+m}) \\ &= \begin{pmatrix} (\Theta - I_n)\prod_{j=1}^{n}(\Theta - \eta_j I_n) & 0 \\ -\mathcal{D}\prod_{j=1}^{n}(\Theta - \eta_j I_n) & 0 \end{pmatrix},\end{aligned}$$

where $\Theta = I_n - \mathcal{F}^{-1}\mathcal{G} + \mathcal{F}^{-1}B\mathcal{D}$.

Since $\eta_j (j = 1, 2, \ldots, n)$ are the eigenvalues of the matrix Θ, following the spirit of the Hamilton-Cayley theorem, it is easy to see that $\prod_{j=1}^{n}(\Theta - \eta_j I_n)$, this leads to $\Psi(\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}) = 0$.

The following conclusion is direct consequence of the above statements and therefore its proof is omitted.

Theorem 4.4. Under the assumptions of Theorem 4.3, if $\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}$ has $k(1 \le k \le n)$ distinct eigenvalues $\eta_j(1 \le j \le k)$ related to algebraic multiplicity γ_j with $\sum_{j=1}^{k} = n$, respectively, then the degree of the minimal polynomial of the preconditioned matrix $\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}$ is at most $k + 1(1 \le k \le n)$. Thus, the dimension of the Krylov subspace $K(\mathcal{P}_{\alpha,0}^{-1}\mathcal{A}, b)$ is at most $k + 1(1 \le k \le n)$. □

Next, we restrict our attentions to the determination of the optimal parameters problem. It is easy to see that the performance of the $\mathcal{P}_{SFHSS}$ preconditioner largely depends on the choices of parameters α and β. However, in our experience this is a difficult task to select the optimal parameters, therefore, we usually need to investigate the estimation method in practical implementations.

By taking similar steps to those taken in [26], we are ready to consider the choice of the parameters α and β in the SFHSS iteration methods. In order to obtain the fast convergence rate of the SFHSS iteration method (2.2) and clustered eigenvalue distribution of the SFHSS-preconditioned matrix, we usually think it is very important to choice a suitable preconditioner $2\mathcal{P}_{SFHSS}$ in (4.2) to approximate infinitely $\mathcal{A}$, hence, we may expect $\mathcal{R}_{SFHSS} \approx 0$ defined as in (4.2) to compute the quasi-optimal iteration parameters α_{exp} and β_{exp}.

We begin our analysis by minimizing the following Frobenius norm of $\mathcal{R}_{SFHSS}$

$$\begin{aligned}\Theta(\alpha,\beta) &\triangleq \|4\mathcal{R}_{SFHSS}\|_F \\ &= \left\| \begin{pmatrix} \alpha I_n + \frac{4}{\alpha} HS & 0 \\ 0 & \beta B^T B \end{pmatrix} \right\|_F \\ &= n\alpha^2 + 4tr(HS - SH) - \frac{16}{\alpha^2} tr(HS^2H) + \beta^2 tr(B^T B)^2.\end{aligned}$$

where $tr(E)$ denotes the trace of the matrix E.

By taking partial derivative for $\Theta(\alpha,\beta)$, we can obtain

$$\frac{\partial\Theta(\alpha,\beta)}{\partial\alpha} = 2n\alpha + \frac{32}{\alpha^3} tr(HS^2H) \quad \text{and} \quad \frac{\partial\Theta(\alpha,\beta)}{\partial\beta} = 2\beta tr(B^T B)^2.$$

It is obvious that $\Theta(\alpha,\beta)$ has a minimum if

$$\alpha_{exp} = 2\sqrt[4]{-\frac{tr(HS^2H)}{n}}.$$

The proof of $tr(HS^2H) < 0$ refer to [26, Lemma 1]. In addition, in practical implementations, it seems a good idea to try some values as close to 0 as possible for the iteration parameter β_{exp}.

5. Numerical examples

In this section, we present some numerical experiments to test the feasibility and robustness of the generalized shift-HSS iteration method for solving the saddle point problem (1.1) arising from Oseen models of incompressible flow. In order to evaluate the performance of the proposed generalized shift-HSS preconditioner over some existing matrix splitting preconditioners, we compare the numerical results of the generalized shift-HSS preconditioner $\mathcal{P}_{SFHSS}$ (2.1) with the GSS preconditioner $\mathcal{P}_{GSS}$ (1.3), the SS preconditioner $\mathcal{P}_{SS}$ (1.4), the MSS preconditioner $\mathcal{P}_{MSS}$ (1.5), the GMSS preconditioner $\mathcal{P}_{MSS}$ (1.5), the DPSS preconditioner $\mathcal{P}_{\mathcal{DPSS}}$ presented in [27, 28] and the preconditioner $\mathcal{P}_{\mathcal{IDPSS}}$ established by [29], where the preconditioner $\mathcal{P}_{\mathcal{DPSS}}$ is defined by

$$\mathcal{P}_{\mathcal{DPSS}} = \frac{1}{2\alpha} \begin{pmatrix} \alpha I_n + A & \\ & \alpha I_m \end{pmatrix} \begin{pmatrix} \alpha I_n & B \\ -B^T & \alpha I_m \end{pmatrix}. \tag{5.1}$$

Here, the optimal iteration parameter is given by $\alpha_{exp} = \frac{\|A\|_F + 2\|B\|_F}{2(n+m)}$ ([30]). In addition, the preconditioner $\mathcal{P}_{\mathcal{IDPSS}}$ is given by

$$\mathcal{P}_{\mathcal{IDPSS}} = \frac{1}{2\alpha} \begin{pmatrix} \alpha I_n + A & \\ & 2\alpha I_m \end{pmatrix} \begin{pmatrix} \alpha I_n & B \\ -B^T & \end{pmatrix}, \tag{5.2}$$

the optimal iteration parameter is given by $\alpha_{exp} = \frac{||A||_F + ||B||_F}{2\sqrt{n}}$ ([24]).

We use the above preconditioners to accelerate GMRES iteration method and compare these different preconditioned GMRES iteration methods in terms of both the number of iteration steps (denoted by IT) and elapsed CPU times in seconds (denoted by CPU). In our implementations, we choose the zero vector $x^{(0)} = 0$ as the initial guess, take the right-hand-side vector b so that the exact solutions u and v are the unity vectors with all entries equal to one, and set the stopping criterion to be the residual norm

$$\text{RES} = \frac{||b - Ax^{(k)}||_2}{||b - Ax^{(0)}||_2} < 10^{-6}$$

or the prescribed iteration number $k_{max} = n$, where $x^{(k)}$ is the solution at the kth iteration. In actual applications, the iteration parameters α and β for the preconditioner $\mathcal{P}_{SFHSS}$ are chosen to be the experimentally found optimal value, which leads to the least numbers of iterations of the preconditioned GMRES method for each choice of the spatial mesh-sizes [18].

Example 5.1. [31]. Consider the Oseen equation of form

$$\begin{cases} -\nu \triangle u + (v \cdot \nabla)u + \nabla p = f, \\ \nabla \cdot u = 0, \end{cases} \text{in } \Omega. \tag{5.3}$$

where Ω is a bounded domain with suitable boundary conditions, the parameter $\nu > 0$ denotes the viscosity, u represents the vector field and stands for the velocity, v is the approximation of u from the previous Picard iteration, and p denotes the pressure. The test problem is the classical two-dimensional leaky-lid driven cavity problem. Here, we usually employ the “IFISS” software package proposed in [32] to discretize the Oseen problem (5.3) with the $Q_2 - Q_1$ mixed finite element method on uniform grid. The generated saddle point system of type (1.1) has nonsymmetric positive definite sub-matrix B which corresponds to a discretization of the convection diffusion operator $L[u] := -\nu \triangle u + (v \cdot \nabla)u$. In actual implementation, four values of the viscosity parameters are used, such as $\nu = 1, 0.1, 0.01, 0.001$, and four increasing grids are selected, i.e., 16×16, 32×32, 64×64 and 128×128 grids.

In Tables 1–7, we use GMRES iteration method in conjunction with the corresponding preconditioners and present IT and CPU with respect to different sizes of the discretization grids for different values of α, β and ν. As these tables show, we can easily know that the generalized shift-HSS method behaves much better than the MSS, GSS, SS, DPSS and IDPSS iteration methods, especially when problem size increases, the convergence rate of GMRES iteration method with the generalized shift-HSS preconditioner are much faster than that of GMRES iteration method with the GMSS, MSS, GSS, SS, DPSS and IDPSS preconditioners. Therefore, the generalized shift-HSS preconditioner is more efficient and stable to accelerate the convergence rate of the GMRES iteration method.

Table 1. IT and CPU with $\nu = 1$, $\alpha = 0.01$ and $\beta = 0.005$.

Grid		64 × 64	128 × 128	256 × 256	512 × 512
$\mathcal{P}_{SFHSS}$	CPU	2.2468	19.5751	159.8347	978.9446
	IT	4	4	4	4
$\mathcal{P}_{SS}$	CPU	7.4697	127.3370	1.6111e+03	2.0024e+04
	IT	47	169	487	998
$\mathcal{P}_{GSS}$	CPU	6.1840	88.7871	1.2920e+03	8.1583e+04
	IT	36	116	392	4550
$\mathcal{P}_{MSS}$	CPU	6.2798	78.7841	660.9457	1.4831e+04
	IT	37	104	200	817
$\mathcal{P}_{GMSS}$	CPU	7.0076	86.0697	807.2316	1.1505e+04
	IT	42	113	244	631
$\mathcal{P}_{DPSS}$	CPU	81.6477	993.7621	7.5473e+03	1.0011e+05
	IT	260	573	627	1121
$\mathcal{P}_{IDPSS}$	CPU	50.0780	369.2814	2.9020e+03	1.8826e+04
	IT	164	216	243	200

Table 2. IT and CPU with $\nu = 0.1$, $\alpha = 0.01$ and $\beta = 0.005$.

Grid		16 × 16	32 × 32	64 × 64	128 × 128
$\mathcal{P}_{SFHSS}$	CPU	0.0560	0.3080	2.4352	20.0278
	IT	3	3	4	4
$\mathcal{P}_{SS}$	CPU	0.0755	0.6873	7.9002	128.0045
	IT	11	20	47	169
$\mathcal{P}_{GSS}$	CPU	0.0590	0.4890	6.1251	89.8709
	IT	7	15	36	116
$\mathcal{P}_{MSS}$	CPU	0.0807	0.6550	6.6287	80.3747
	IT	13	17	37	104
$\mathcal{P}_{GMSS}$	CPU	0.0841	0.8567	7.4521	86.2577
	IT	14	24	42	113
$\mathcal{P}_{DPSS}$	CPU	0.9915	8.5094	83.2322	998.5375
	IT	87	180	255	573
$\mathcal{P}_{IDPSS}$	CPU	0.8671	8.6078	51.8677	373.8926
	IT	77	140	164	216

Table 3. IT and CPU with $\nu = 0.01$, $\alpha = 0.01$ and $\beta = 0.005$.

Grid		16 × 16	32 × 32	64 × 64	128 × 128
$\mathcal{P}_{SFHSS}$	CPU	0.0690	0.3106	2.5461	19.4539
	IT	3	3	4	4
$\mathcal{P}_{SS}$	CPU	0.1031	0.7981	7.7896	130.0777
	IT	11	21	47	169
$\mathcal{P}_{GSS}$	CPU	0.0499	0.5213	5.9485	90.4792
	IT	8	15	36	116
$\mathcal{P}_{MSS}$	CPU	0.0950	0.5529	5.9219	83.7678
	IT	13	17	37	104
$\mathcal{P}_{GMSS}$	CPU	0.0903	0.7730	6.6637	89.1844
	IT	14	24	42	113
$\mathcal{P}_{DPSS}$	CPU	1.0676	9.0995	79.1151	1.0232e+03
	IT	87	150	260	573
$\mathcal{P}_{IDPSS}$	CPU	0.9469	8.0854	49.9778	372.0878
	IT	77	140	164	216

Table 4. IT and CPU with $\nu = 1$, $\alpha = \alpha_{exp}$ and $\beta = 0.05$.

Grid		16 × 16	32 × 32	64 × 64	128 × 128
$\mathcal{P}_{SFHSS}$	CPU	0.1059	0.4267	2.5201	20.0278
	IT	4	4	4	4
$\mathcal{P}_{DPSS}$	CPU	0.6867	11.1614	52.6255	913.3530
	IT	52	152	163	515
$\mathcal{P}_{IDPSS}$	CPU	0.2370	1.1227	4.5158	21.7143
	IT	18	15	13	13

Table 5. IT and CPU with $\nu = 0.001$, $\alpha = 0.01$ and $\beta = 0.005$.

Grid		16×16	32×32	64×64	128×128
$\mathcal{P}_{SFHSS}$	CPU	0.0794	0.3281	2.4645	19.4732
	IT	3	3	4	4
$\mathcal{P}_{SS}$	CPU	0.0876	0.7553	7.6986	128.4442
	IT	11	21	47	169
$\mathcal{P}_{GSS}$	CPU	0.0620	0.5892	6.1956	89.6522
	IT	8	15	36	116
$\mathcal{P}_{MSS}$	CPU	0.0923	0.5876	5.9810	79.4758
	IT	15	17	37	104
$\mathcal{P}_{GMSS}$	CPU	0.0989	0.8472	7.0856	87.2737
	IT	14	24	42	113
$\mathcal{P}_{DPSS}$	CPU	1.0748	9.1363	81.7333	1.0026e+03
	IT	87	150	260	573
$\mathcal{P}_{IDPSS}$	CPU	0.9799	8.5531	51.4426	367.3563
	IT	77	140	164	216

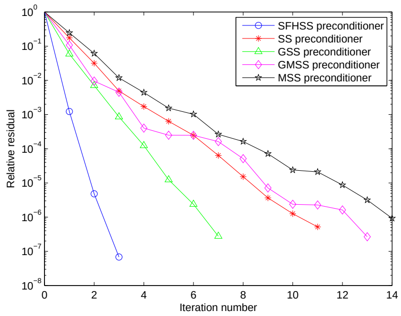
Table 6. IT and CPU with $\nu = 0.1$, $\alpha = \alpha_{exp}$ and $\beta = 0.05$.

Grid		16×16	32×32	64×64	128×128
$\mathcal{P}_{SFHSS}$	CPU	0.0975	0.4061	2.4989	19.6638
	IT	4	4	4	4
$\mathcal{P}_{DPSS}$	CPU	0.6554	10.7887	52.3130	940.0833
	IT	52	152	163	510
$\mathcal{P}_{IDPSS}$	CPU	0.2531	1.1453	4.1881	22.5455
	IT	18	15	13	13

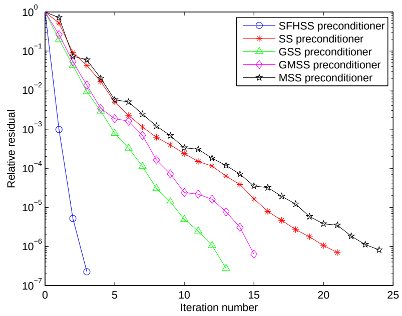
Table 7. IT and CPU with $\nu = 0.01$, $\alpha = \alpha_{exp}$ and $\beta = 0.001$.

Grid		16×16	32×32	64×64	128×128
$\mathcal{P}_{SFHSS}$	CPU	0.0595	0.2582	1.5179	12.3510
	IT	2	2	2	4
$\mathcal{P}_{DPSS}$	CPU	0.6741	11.4814	53.4351	939.0087
	IT	52	152	163	525
$\mathcal{P}_{IDPSS}$	CPU	0.2454	1.1735	4.6387	22.7694
	IT	18	15	13	13

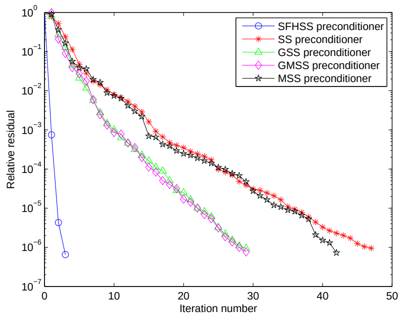
From Figures 1–4, we give the convergence history of the corresponding iteration methods to compare effects of the corresponding preconditioners with respect to the iteration parameters α and β. It is easily seen that the generalized shift-HSS iteration method has more smooth convergence curves than the GMSS, MSS, GSS and SS iteration methods.



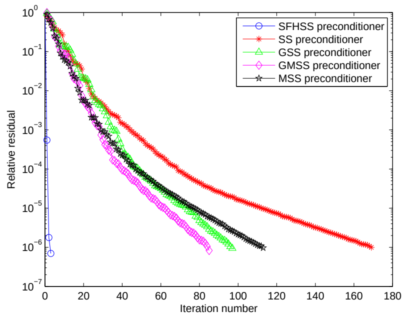
(a) Grid = 16 × 16



(b) Grid = 32 × 32

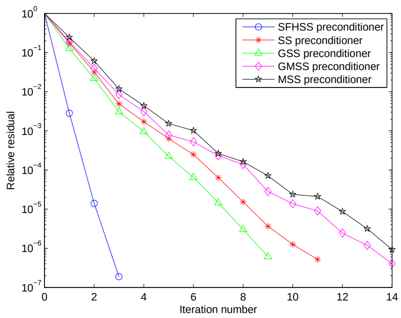


(c) Grid = 64 × 64

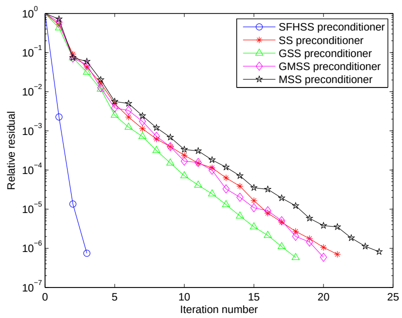


(d) Grid = 128 × 128

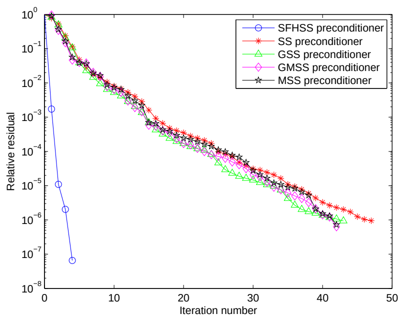
Figure 1. Convergence curves of GMRES iteration methods.



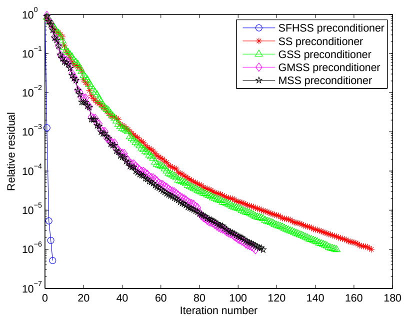
(a) Grid = 16 × 16



(b) Grid = 32 × 32

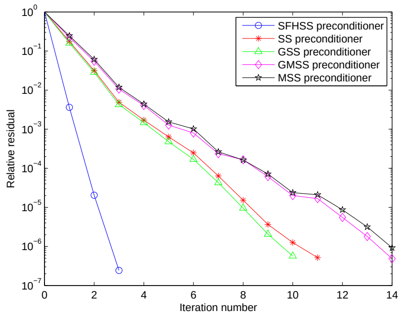


(c) Grid = 64 × 64

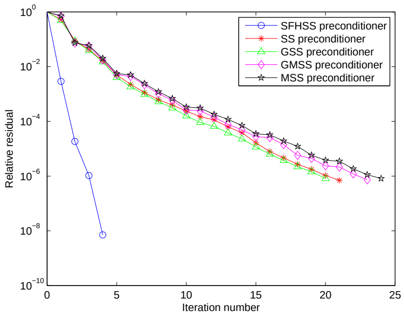


(d) Grid = 128 × 128

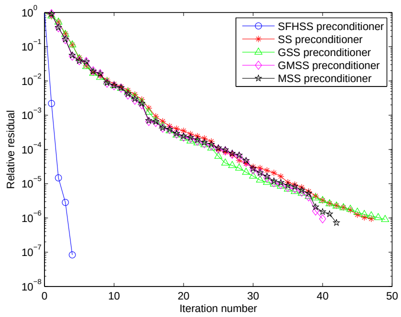
Figure 2. Convergence curves of GMRES iteration methods.



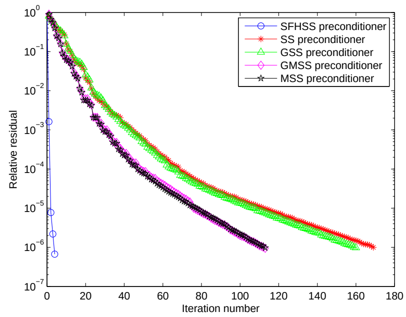
(a) Grid = 16×16



(b) Grid = 32×32

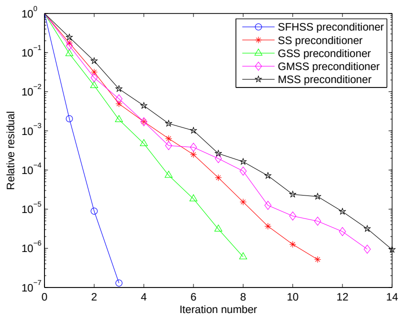


(c) Grid = 64×64

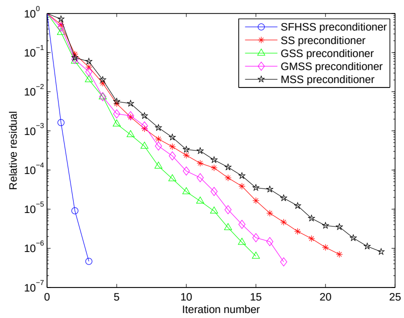


(d) Grid = 128×128

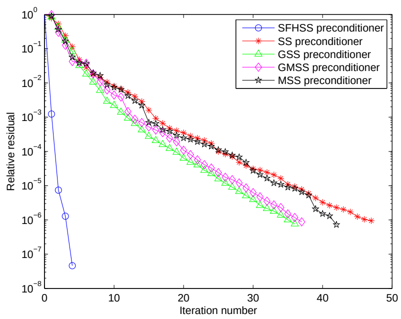
Figure 3. Convergence curves of GMRES iteration methods.



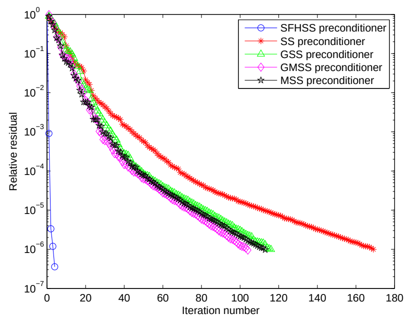
(a) Grid = 16×16



(b) Grid = 32×32



(c) Grid = 64×64



(d) Grid = 128×128

Figure 4. Convergence curves of GMRES iteration methods.

Example 5.2. [33, 34]. Consider the two-dimensional convection-diffusion equation

$$-\nabla^2 u + q \nabla u = f(x,y) \quad \text{in } \Omega = [0,1] \times [0,1], \tag{5.4}$$

with Dirichlet boundary condition and the constant coefficient q. Similar to the three-dimensional case proposed in [35], the five-point centered finite difference discretization is used for the above equation, the nonsymmetric saddle point system (1.1) can be easily obtained, where

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

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

and

$$T_r = \frac{1}{h^2} tridiag(-1-r, 2, -1+r) \in R^{l \times l}, \qquad F = \frac{1}{h} tridiag(-1, 1, 0) \in R^{l \times l}.$$

Here, $h = \frac{1}{l+1}$ represents an equidistant step-size in each coordinate direction, $\otimes$ denotes the Kronecker product and $r = qh/2$ indicates the mesh Reynolds number.

In Tables 8–11, from two aspects of IT and CPU, we use SFHSS, DPSS and IDPSS preconditioners to accelerate GMRES iteration method associated with different sizes of the discretization grids for different values of q with α_{exp} and $\beta = 0.00001$. As these tables show, we can easily know that the SFHSS method outperforms the DPSS and IDPSS methods, especially when problem size increases, the convergence rate of GMRES iteration method with the generalized shift-HSS preconditioner are much faster than that of GMRES iteration method with the DPSS and IDPSS preconditioners. Therefore, the generalized SFHSS preconditioner is more efficient and stable.

Table 8. IT and CPU with $q = 0.01$, $\alpha = \alpha_{exp}$ and $\beta = 0.00001$.

l		16	32	64	128
$\mathcal{P}_{SFHSS}$	CPU	0.0430	0.1943	1.1330	6.9972
	IT	3	4	4	5
$\mathcal{P}_{DPSS}$	CPU	0.5416	6.8962	142.8853	743.6954
	IT	68	127	650	680
$\mathcal{P}_{IDPSS}$	CPU	0.0949	0.4838	2.1202	10.0982
	IT	10	10	9	8

Table 9. IT and CPU with $q = 0.1$, $\alpha = \alpha_{exp}$ and $\beta = 0.00001$.

l		16	32	64	128
$\mathcal{P}_{SFHSS}$	CPU	0.0405	0.1926	1.1506	6.8099
	IT	4	4	4	5
$\mathcal{P}_{DPSS}$	CPU	0.5003	5.7164	62.4237	807.8969
	IT	66	152	341	670
$\mathcal{P}_{IDPSS}$	CPU	0.0804	0.3724	1.8443	8.7051
	IT	10	10	9	8

Table 10. IT and CPU with $q = 1$, $\alpha = \alpha_{exp}$ and $\beta = 0.00001$.

l		16	32	64	128
$\mathcal{P}_{SFHSS}$	CPU	0.0405	0.2352	1.3521	7.8656
	IT	4	4	4	5
$\mathcal{P}_{DPSS}$	CPU	0.7747	12.3715	151.9526	988.8506
	IT	86	258	694	840
$\mathcal{P}_{IDPSS}$	CPU	0.0899	0.5204	2.2006	9.8894
	IT	10	10	9	8

Table 11. IT and CPU with $q = 10$, $\alpha = \alpha_{exp}$ and $\beta = 0.00001$.

l		16	32	64	128
$\mathcal{P}_{SFHSS}$	CPU	0.0650	0.2451	1.3483	7.9823
	IT	4	4	4	5
$\mathcal{P}_{DPSS}$	CPU	0.6067	11.2988	165.5783	2.1922e+03
	IT	74	273	761	1844
$\mathcal{P}_{IDPSS}$	CPU	0.0964	0.4138	2.1823	17.0936
	IT	12	10	9	8

6. Conclusions

The novelty of this present paper is the construction and analysis of the generalized shift-HSS iteration method for nonsingular saddle point systems with nonsymmetric positive definite (1,1)-block. We investigate the convergence property of the SFHSS iteration method and further illustrate the robustness and efficiency of the generalized shift-HSS preconditioner by a numerical example. Future work should focus on developing the modified forms of the GSS iteration method and generalized shift GSOR-like method for complex symmetric linear system, and study the effects of iteration parameters on eigenvalue-clustering of the corresponding preconditioned matrices.

Acknowledgments

This research is supported by Scientific and Technological Research Program of Chongqing Research Program of Basic Research and Frontier Technology (cstc2018jcyjAX0685), Xiangsi Lake Young Scholars Innovation Team of Guangxi Minzu University (No.2021RSCXSHQN05) and Natural Science Foundation of Guangxi Minzu University (2021KJQD01).

Conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

1. M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, *Acta Numer.*, **14** (2005), 1–137. https://doi.org/10.1017/S0962492904000212
2. C. Q. Lv, C. F. Ma, BCR method for solving generalized coupled Sylvester equations over centrosymmetric or anti-centrosymmetric matrix, *Comput. Math. Appl.*, **75** (2018), 70–88. https://doi.org/10.1016/j.camwa.2017.08.041
3. H. C. Elman, Preconditioning for the steady-state Navier-Stokes equations with low viscosity, *SIAM J. Sci. Comput.*, **20** (1999), 1299–1316. https://doi.org/10.1137/S1064827596312547
4. Z. Z. Bai, B. N. Parlett, Z. Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, *Numer. Math.*, **102** (2005), 1–38. https://doi.org/10.1007/s00211-005-0643-0
5. S. L. Wu, T. Z. Huang, X. L. Zhao, A modified SSOR iterative method for augmented systems, *J. Comput. Appl. Math.*, **228** (2009), 424–433. https://doi.org/10.1016/j.cam.2008.10.006
6. A. L. Yang, Y. J. Wu, The Uzawa-HSS method for saddle-point problems, *Appl. Math. Lett.*, **38** (2014), 38–42. https://doi.org/10.1016/j.aml.2014.06.018
7. P. Y. Chen, J. G. Huang, H. S. Sheng, Some Uzawa methods for steady incompressible Navier-Stokes equations discretized by mixed element methods, *J. Comput. Appl. Math.*, **273** (2015), 313–325. https://doi.org/10.1016/j.cam.2014.06.019
8. Z. Z. Bai, G. H. Golub, M. K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM J. Matrix. Anal. Appl.*, **24** (2002), 603–626. https://doi.org/10.1137/S0895479801395458
9. Z. Z. Bai, G. H. Golub, Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems, *IMA J. Numer. Anal.*, **27** (2007), 1–23. https://doi.org/10.1093/imanum/drl017
10. Z. H. Huang, T. Z. Huang, Spectral properties of the preconditioned AHSS iteration method for generalized saddle point problems, *Comput. Appl. Math.*, **29** (2010), 269–295.
11. Z. H. Huang, H. Su, A modified shift-splitting method for nonsymmetric saddle point problems, *J. Comput. Appl. Math.*, **317** (2017), 535–546. https://doi.org/10.1016/j.cam.2016.11.032
12. Z. Z. Bai, S. L. Zhang, A regularized conjugate gradient method for symmetric positive definite system of linear equations, *J. Comput. Math.*, **20** (2002), 437–448.
13. Z. Z. Bai, J. F. Yin, Y. F. Su, A shift-splitting preconditioner for non-Hermitian positive definite matrices, *J. Comput. Math.*, **24** (2006), 539–552.
14. Y. Cao, H. R. Tao, M. Q. Jiang, Generalized shift splitting preconditioners for saddle point problems, *Math. Numer. Sinca*, **36** (2014), 16–26.
15. C. R. Chen, C. F. Ma, A generalized shift-splitting preconditioner for saddle point problems, *Appl. Math. Lett.*, **43** (2015), 49–55. https://doi.org/10.1016/j.aml.2014.12.001
16. Y. Cao, J. Du, Q. Niu, Shift-splitting preconditioners for saddle point problems, *J. Comput. Appl. Math.*, **272** (2014), 239–250. https://doi.org/10.1016/j.cam.2014.05.017

17. Y. Cao, S. Li, L. Q. Yao, A class of generalized shift-splitting preconditioners for nonsymmetric saddle point problems, *Appl. Math. Lett.*, **49** (2015), 20–27. https://doi.org/10.1016/j.aml.2015.04.001

18. S. W. Zhou, A. L. Yang, Y. Dou, Y. J. Wu. The modified shift-splitting preconditioners for nonsymmetric saddle-point problems, *Appl. Math. Lett.*, **59** (2016), 109–114. https://doi.org/10.1016/j.aml.2016.03.011

19. Z. G. Huang, L. G. Wang, Z. Xu, J. J. Cui, The generalized modified shift-splitting preconditioners for nonsymmetric saddle point problems, *Appl. Math. Comput.*, **299** (2017), 95–118. https://doi.org/10.1016/j.amc.2016.11.038

20. Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.*, **7** (1986), 856–869. https://doi.org/10.1137/0907058

21. M. Q. Jiang, Y. Cao, On local Hermitian and skew-Hermitian splitting iteration methods for generalized saddle point problems, *J. Comput. Appl. Math.*, **231** (2009), 973–982. https://doi.org/10.1016/j.cam.2009.05.012

22. 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), 397–406. https://doi.org/10.1093/imamat/8.3.397

23. S. T. Ling, Q. B. Liu, New local generalized shift-splitting preconditioners for saddle point problems, *Appl. Math. Comput.*, **302** (2017), 58–67. https://doi.org/10.1016/j.amc.2017.01.014

24. Z. G. Huang, L. G. Wang, Z. Xu, J. J. Cui, An efficient preconditioned variant of the PSS preconditioner for generalized saddle point problems, *Appl. Math. Comput.*, **376** (2020), 125110. https://doi.org/10.1016/j.amc.2020.125110

25. J. L. Zhu, A. L. Yang, Y. J. Wu, A parameterized deteriorated PSS preconditioner and its optimization for nonsymmetric saddle point problems, *Comput. Math. Appl.*, **79** (2020), 1420–1434. https://doi.org/10.1016/j.camwa.2019.09.004

26. Y. M. Huang, A practical formula for computing optimal parameters in the HSS iteration methods, *J. Comput. Appl. Math.*, **255** (2014), 142–149. https://doi.org/10.1016/j.cam.2013.01.023

27. S. Q. Shen, A note on PSS preconditioners for generalized saddle point problems, *Appl. Math. Comput.*, **237** (2014), 723–729. https://doi.org/10.1016/j.amc.2014.03.151

28. Y. Cao, J. L. Dong, Y. M. Wang, A relaxed deteriorated PSS preconditioner for nonsymmetric saddle point problems from the steady Navier-Stokes equation, *J. Comput. Appl. Math.*, **273** (2015), 41–60. https://doi.org/10.1016/j.cam.2014.06.001

29. Q. Q. Shen, Y. Cao, L. Wang, Two improvements of the deteriorated PSS preconditioner for generalized saddle point problems, *Numer. Algor.*, **75** (2017), 33–54. https://doi.org/10.1007/s11075-016-0195-7

30. Y. Cao, A block positive-semidefinite splitting preconditioner for generalized saddle point linear systems, *J. Comput. Appl. Math.*, **374** (2020), 112787. https://doi.org/10.1016/j.cam.2020.112787

31. M. Benzi, G. H. Golub, A preconditioner for generalized saddle point problems, *SIAM J. Matrix Anal. Appl.*, **26** (2004), 20–41. https://doi.org/10.1137/S0895479802417106

32. H. C. Elman, A. Ramage, D. J. Silvester, Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow, *ACM T. Math. Software*, **33** (2007), 14. https://doi.org/10.1145/1236463.1236469

33. A. Hadjidimos, The saddle point problem and the Manteuffel algorithm, *BIT. Numer. Math.*, **56** (2016), 1281–1302. https://doi.org/10.1007/s10543-016-0617-x

34. Y. Dou, A. L. Yang, Y. J. Wu, Z. Z. Liang, Convergence analysis of modified PGSS methods for singular saddle-point problems, *Comput. Math. Appl.*, **77** (2019), 93–104. https://doi.org/10.1016/j.camwa.2018.09.016

35. Z. Z. Bai, A. Hadjidimos, Optimization of extrapolated Cayley transform with non-Hermitian positive definite matrix, *Linear Algebra Appl.*, **463** (2014), 322–339. https://doi.org/10.1016/j.laa.2014.08.021