Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(10): 7583–7600.
DOI: 10.3934/era.2026327
Received: 18 April 2026
Revised: 11 August 2026
Accepted: 17 August 2026
Published: 03 September 2026

Research article

A block diagonal preconditioner with two preconditioning matrices for saddle point problems

Hong-Yu Wu[1,2,*]

[1] School of Mathematics and Statistics, Hunan First Normal University, Changsha 410205, China

[2] Hunan Provincial University Key Laboratory for Big Data Analysis and Application, Hunan First Normal University, Changsha 410205, China

* **Correspondence:** Email: styhywu@hnfnu.edu.cn.

Abstract: This paper introduced a block diagonal preconditioner that incorporated two distinct preconditioned matrices for solving saddle point problems. Then we established some convergence conditions for the resulting iterative method under appropriate conditions. Moreover, a spectral property of the preconditioned matrix was also provided. Numerical experiments demonstrated that the proposed preconditioner exhibited superior performance compared with several recently developed preconditioners.

Keywords: block diagonal preconditioner; saddle point problem; convergence; iterative method

1. Introduction

We are concerned with the following saddle point problem numerically:

$$\mathcal{A}u \equiv \begin{bmatrix} A & B^T \\ -B & O \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ q \end{bmatrix} \equiv f, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, and $B \in \mathbb{R}^{m\times n}$ has full row rank ($m \leq n$). Moreover, we have $b \in \mathbb{R}^n$ and $q \in \mathbb{R}^m$, and B^T represents the transpose of B. Thus, the saddle point problem (1.1) admits a unique solution, which is attributed to the invertibility of its coefficient matrix $\mathcal{A}$. The linear system (1.1) with a saddle point structure arises in numerous areas of scientific computing and engineering, including computational fluid dynamics, Navier-Stokes equations, constraint optimization, piezoelectric equations, least-squares problems, and others. We refer to [1–3] for further details.

Nowadays, matrix splitting techniques serve as the foundation of developing various iterative methods and preconditioners when they are employed to solve linear systems [2, 4]. Consider

$$\widetilde{\mathcal{A}}\widetilde{u} = \widetilde{f},$$

where $\widetilde{\mathcal{A}}$ is a non-Hermitian matrix with a positive definite Hermitian part, the following iterative method, known as the Hermitian and skew-Hermitian splitting (HSS) iterative method, was originally introduced by Bai et al. [5]:

$$\begin{cases} (\alpha I + \widetilde{\mathcal{H}})\widetilde{u}^{k+\frac{1}{2}} = (\alpha I - \widetilde{\mathcal{S}})\widetilde{u}^{(k)} + \widetilde{f}, \\ (\alpha I + \widetilde{\mathcal{S}})\widetilde{u}^{(k+1)} = (\alpha I - \widetilde{\mathcal{H}})\widetilde{u}^{k+\frac{1}{2}} + \widetilde{f}, \ k = 0, 1, ..., \end{cases} \tag{1.2}$$

where the Hermitian and skew-Hermitian parts of $\widetilde{\mathcal{A}}$ are given by $\widetilde{\mathcal{H}} = 1/2(\widetilde{\mathcal{A}} + \widetilde{\mathcal{A}}^*)$ and $\widetilde{\mathcal{S}} = 1/2(\widetilde{\mathcal{A}} - \widetilde{\mathcal{A}}^*)$, respectively. The original HSS iterative method has subsequently inspired a significant body of work. To improve upon the HSS iterative method (1.2), the more effective single-step HSS (SHSS) iterative method [6] was developed:

$$(\alpha I + \widetilde{\mathcal{H}})\widetilde{u}^{(k+1)} = (\alpha I - \widetilde{\mathcal{S}})\widetilde{u}^{(k)} + \widetilde{f}, \ k = 0, 1, ... \tag{1.3}$$

A key advantage of the SHSS iterative method (1.3) is that it avoids solving the computationally expensive shifted skew-Hermitian linear subsystem. Subsequently, by incorporating a preconditioned matrix, Wang et al. [7] established a new iterative method:

$$(\widetilde{P} + \widetilde{\mathcal{H}})\widetilde{u}^{(k+1)} = (\widetilde{P} - \widetilde{\mathcal{S}})\widetilde{u}^{(k)} + \widetilde{f}, \ k = 0, 1, ..., \tag{1.4}$$

where $\widetilde{P}$ is a Hermitian positive definite matrix. A key observation is that the iterative method (1.4) is more general and flexible than the SHSS method (1.3), which it contains by noting that the former reduces to the latter when $\widetilde{P} = \alpha I$. Additionally, since $\widetilde{P} + \widetilde{\mathcal{H}}$ is Hermitian positive definite, it admits the use of effective solvers like the conjugate gradient method or Cholesky factorization.

During the past two decades, iterative methods and preconditioning techniques have become vital tools for solving (1.1) [8]. Among numerous numerical approaches, Krylov subspace methods with high-quality preconditioners are one of the most effective categories. Many contributions in this direction include block diagonal preconditioners [9–11], block triangular preconditioners [12–14], constraint preconditioners [15–17], and preconditioners inspired by matrix splitting schemes [18–20]. Additionally, iterative methods and preconditioners inspired by matrix splitting techniques have also played an important role in solving (1.1). For instance, motivated by the HSS iterative scheme, an HSS preconditioner for a generalized version of (1.1) was proposed by Benzi and Golub [21]. Moreover, several improvements of HSS preconditioners have been presented [22–24]. These studies demonstrate that preconditioners derived from matrix splitting strategies can significantly enhance the convergence of Krylov solvers. The effectiveness of a preconditioner is determined by the specific structure of the problem and the actual choice of splitting schemes [3]. Therefore, the design of a good preconditioner remains a central task in developing effective solution strategies for (1.1).

This paper presents a solution strategy for the saddle point problem (1.1), which is inspired by the iterative method (1.4). Sufficient convergence conditions of the resulting iterative scheme are established through rigorous analysis. Furthermore, a novel block diagonal preconditioner

incorporating two preconditioning matrices is designed to accelerate the convergence of Krylov subspace methods, which is structured as

$$\mathcal{P} = \begin{pmatrix} P + A & O \\ O & Q \end{pmatrix}.$$

Here, P and Q are symmetric positive definite matrices. Our contributions are summarized in three aspects. First, unlike standard block diagonal preconditioners that simply take the (1,1)-block as A, we introduce P to regularize the (1,1)-block. Meanwhile, this proposed preconditioner avoids the full Schur complement. As a result, this preconditioner effectively improves the spectral properties of the original matrix, thereby enhancing the computational efficiency. Second, it should be pointed out that the proposed preconditioner is more flexible than the SHSS preconditioner, as it not only reduces to the SHSS preconditioner but also yields new preconditioners. Third, suitable choices of the preconditioning matrices P and Q can achieve fast convergence.

This work is organized as follows: Section 2 reviews existing block preconditioners inspired by the HSS scheme. Section 3 presents our proposed iterative method, along with a detailed analysis of its convergence conditions. Section 4 introduces a block diagonal preconditioner and investigates the spectral properties of the preconditioned matrix. Section 5 presents numerical experiments to validate the effectiveness of the proposed strategy. Finally, concluding remarks are provided in Section 6.

2. A review of the HSS-type preconditioners

We begin with a concise overview of the well-known HSS preconditioner and its improved variants. Following the HSS iterative framework, the coefficient matrix $\mathcal{A}$ admits the splitting

$$\mathcal{A} = \mathcal{H} + \mathcal{S},$$

where

$$\mathcal{H} = \frac{1}{2}(\mathcal{A} + \mathcal{A}^T) = \begin{bmatrix} A & O \\ O & O \end{bmatrix}, \ \mathcal{S} = \frac{1}{2}(\mathcal{A} - \mathcal{A}^T) = \begin{bmatrix} O & B^T \\ -B & O \end{bmatrix}.$$

Benzi and Golub [21] demonstrated that the HSS iterative method is unconditionally convergent. Furthermore, by employing the special matrix splitting scheme, we can derive the HSS preconditioner

$$\begin{aligned} \mathcal{P}_{HSS} &= \frac{1}{2\alpha}(\alpha I + \mathcal{H})(\alpha I + \mathcal{S}) \\ &= \frac{1}{2\alpha}\begin{bmatrix} \alpha I + A & O \\ O & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} \alpha I + A & (I + \frac{1}{\alpha}A)B^T \\ -B & \alpha I \end{bmatrix}. \end{aligned} \tag{2.1}$$

Since the pre-factor $\frac{1}{2}$ does not influence the preconditioned system, the HSS preconditioner (2.1) is typically simplified to the form:

$$\widetilde{\mathcal{P}}_{HSS} = \begin{bmatrix} \alpha I + A & (I + \frac{1}{\alpha}A)B^T \\ -B & \alpha I \end{bmatrix}. \tag{2.2}$$

Utilizing the splitting of the sub-matrix A in (1.1), we obtain the following splitting scheme

$$\widehat{\mathcal{H}} = \begin{bmatrix} H & B^T \\ O & O \end{bmatrix}, \widehat{\mathcal{S}} = \begin{bmatrix} S & O \\ -B & O \end{bmatrix}.$$

Here, $H = 1/2(A + A^T)$ and $S = 1/2(A - A^T)$ denote, respectively, the symmetric and skew-symmetric parts of A. Following the classical alternating direction implicit (ADI) method, Liu [25] subsequently developed a new alternating local HSS (ALHSS) preconditioner with the form of

$$\begin{aligned} \mathcal{P}_{ALHSS} &= \frac{1}{2\alpha}\begin{bmatrix} \alpha I + H & B^T \\ O & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I + S & O \\ -B & \alpha I \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} \alpha I + A + \frac{1}{\alpha}HS - \frac{1}{\alpha}B^T B & B^T \\ -B & \alpha I \end{bmatrix}. \end{aligned}$$

Theoretical findings establish that the eigenvalues of the corresponding preconditioned matrix cluster around the points $(0, 0)$ and $(2, 0)$. Motivated by relaxed techniques and the preconditioner (2.2), Cao et al. [22] consequently developed a relaxed HSS (RHSS) preconditioner

$$\begin{aligned} \mathcal{P}_{RHSS} &= \frac{1}{\alpha}\begin{bmatrix} A & O \\ O & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & O \end{bmatrix} \\ &= \begin{bmatrix} A & \frac{1}{\alpha}AB^T \\ -B & O \end{bmatrix}. \end{aligned} \tag{2.3}$$

To address the limitations of the HSS and RHSS preconditioners, Salkuyeh and Masoudi [23] thus presented a new RHSS (REHSS) preconditioner

$$\mathcal{P}_{REHSS} = \begin{bmatrix} A & AB^T \\ -B & \alpha I \end{bmatrix}. \tag{2.4}$$

Numerical results demonstrated the superior performance of the REHSS preconditioner. By incorporating a regularization matrix, Bai and Benzi [24] presented a class of regularized HSS preconditioners. Numerical results demonstrated the effectiveness of the proposed regularized HSS preconditioner compared with the original HSS preconditioner. Very recently, inspired by the iterative scheme (1.3), Li and Wu [26] developed a SHSS preconditioner for solving (1.1), which can be given by

$$\mathcal{P}_{SHSS} = \begin{bmatrix} \alpha I + A & O \\ O & \alpha I \end{bmatrix}.$$

3. The new iterative method

Drawing inspiration from the iterative method (1.4), we proceed to propose the following splitting scheme:

$$\mathcal{A} = (\Omega + \mathcal{H}) - (\Omega - \mathcal{S}), \tag{3.1}$$

where

$$\Omega = \begin{bmatrix} P & O \\ O & Q \end{bmatrix}.$$

Here, $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric positive definite matrices. According to the matrix splitting (3.1), we now propose a new iterative method to solve (1.1), as outlined below:
The new iterative method: Let $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ be defined as symmetric positive definite matrices. Starting from an initial guess $x^0 \in \mathbb{R}^n$ and $y^0 \in \mathbb{R}^m$, the iterative proceeds for $k = 0, 1, 2, ...$ until $\{x^{(k)}, y^{(k)}\}_{k=0}^{+\infty}$ converges, computing

$$\begin{bmatrix} P+A & O \\ O & Q \end{bmatrix}\begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} = \begin{bmatrix} P & -B^T \\ B & Q \end{bmatrix}\begin{bmatrix} x^{(k)} \\ y^{(k)} \end{bmatrix} + \begin{bmatrix} b \\ q \end{bmatrix}. \tag{3.2}$$

Typically, the matrix $\Omega + \mathcal{H}$ can serve as a preconditioner for the coefficient matrix $\mathcal{A}$. Furthermore, the iterative matrix Γ respect to the new method (3.2) is

$$\Gamma = \begin{bmatrix} P+A & O \\ O & Q \end{bmatrix}^{-1}\begin{bmatrix} P & -B^T \\ B & Q \end{bmatrix}. \tag{3.3}$$

Let $\rho(\Gamma)$ denote the spectral radius of the iterative matrix Γ, then it is clear to observe that the new iterative method (3.2) converges if and only if $\rho(\Gamma) < 1$ [2, 27].

Indeed, the proposed iterative method (3.2) represents a generalized version of the SHSS method [26], which it reduces to the latter when $P = \alpha I$ and $Q = \alpha I$. In actual computations, we can select suitable preconditioning matrices P and Q to minimize the implementation cost of the proposed iterative method (3.2). This flexibility implies that the proposed iterative scheme is more adaptable for solving (1.1).

Next, we proceed to investigate the convergence of the new iterative method (3.2). Specifically, let λ denote an eigenvalue of the iterative matrix Γ defined as in (3.3). Also, the eigenvector associated with λ is denoted by $[\widetilde{x}^T, \widetilde{y}^T]^T$. The associated generalized eigenvalue problem can then be derived:

$$\begin{bmatrix} P & -B^T \\ B & Q \end{bmatrix}\begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix} = \lambda\begin{bmatrix} P+A & O \\ O & Q \end{bmatrix}\begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix},$$

or equivalently,

$$P\widetilde{x} - B^T\widetilde{y} = \lambda(P+A)\widetilde{x}, \tag{3.4}$$

$$B\widetilde{x} + Q\widetilde{y} = \lambda Q\widetilde{y}. \tag{3.5}$$

Immediately, we can derive the following results.

Lemma 3.1. *Assume that $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, and $B \in \mathbb{R}^{m\times n}$ has full row rank. Let Γ be defined as in (3.3). For any eigenvalue λ of Γ that is associated with the eigenvector $[\widetilde{x}^T, \widetilde{y}^T]^T$, it can be concluded that $\lambda \neq 1$.*

Proof. Assuming $\lambda = 1$ and substituting it into (3.4) and (3.5) yields

$$\begin{cases} A\widetilde{x} + B^T\widetilde{y} = 0, \\ -B\widetilde{x} = 0. \end{cases}$$

Given that $\mathcal{A}$ is nonsingular, we obtain $\widetilde{x} = \widetilde{y} = 0$. This result leads to a contradiction with the assumption. Thereby, we have $\lambda \neq 1$. □

Lemma 3.2. *Suppose that the hypotheses of Lemma 3.1 are satisfied. Assume that $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric positive definite matrices, then we have $\widetilde{x} \neq 0$. Furthermore, if $\widetilde{y} = 0$, it follows that $0 < \lambda < 1$.*

Proof. If $\widetilde{x} = 0$, Eq (3.4) thus yields $B^T\widetilde{y} = 0$. Moreover, the assumption of B implies $\widetilde{y} = 0$. This conclusion is inconsistent with the fact that $[\widetilde{x}^T, \widetilde{y}^T]^T$ is the corresponding eigenvector with respect to λ. Therefore, we have $\widetilde{x} \neq 0$.

If $\widetilde{y} = 0$, we obtain from (3.4) that

$$P\widetilde{x} = \lambda P\widetilde{x} + \lambda A\widetilde{x}. \tag{3.6}$$

Let

$$a = \frac{\widetilde{x}^T P\widetilde{x}}{\widetilde{x}^T\widetilde{x}} > 0, \text{ and } b = \frac{\widetilde{x}^T A\widetilde{x}}{\widetilde{x}^T\widetilde{x}} > 0. \tag{3.7}$$

Multiplying both sides of Eq (3.6) by $\frac{\widetilde{x}^T}{\widetilde{x}^T\widetilde{x}}$, we derive

$$\lambda = \frac{\frac{\widetilde{x}^T P\widetilde{x}}{\widetilde{x}^T\widetilde{x}}}{\frac{\widetilde{x}^T P\widetilde{x}}{\widetilde{x}^T\widetilde{x}} + \frac{\widetilde{x}^T A\widetilde{x}}{\widetilde{x}^T\widetilde{x}}}.$$

Then based on (3.7), we have

$$\lambda = \frac{a}{a+b}. \tag{3.8}$$

Under the assumptions of A and P, from (3.8) we conclude that $0 < \lambda < 1$. □

The following lemma provides a crucial tool that is vital to the following results.

Lemma 3.3. *[27] A necessary and sufficient condition for the two roots of real quadratic equation $x^2 - px + q = 0$ to have modulus less than 1 is that $|q| < 1$ and $|p| < 1 + q$.*

Theorem 3.1. *Assume that $A, P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric positive definite matrices, and $B \in \mathbb{R}^{m\times n}$ is a matrix of full row rank. Let Γ be defined as in (3.3). Define*

$$a = \frac{\widetilde{x}^T P\widetilde{x}}{\widetilde{x}^T\widetilde{x}} > 0,\ b = \frac{\widetilde{x}^T A\widetilde{x}}{\widetilde{x}^T\widetilde{x}} > 0,\ c = \frac{\widetilde{x}^T B^T Q^{-1} B\widetilde{x}}{\widetilde{x}^T\widetilde{x}} \geq 0, \text{ and } \|\widetilde{x}\|_2 = 1. \tag{3.9}$$

If $c < b$, then

$$\rho(\Gamma) < 1.$$

Thus, the iterative method (3.2) is convergent.

Proof. It follows from Lemma 3.1 and (3.5) that

$$\widetilde{y} = \frac{Q^{-1}B\widetilde{x}}{\lambda - 1}. \tag{3.10}$$

Substituting (3.10) into (3.4) results in

$$(\lambda - 1)P\widetilde{x} + \lambda A\widetilde{x} + \frac{B^T Q^{-1} B\widetilde{x}}{\lambda - 1} = 0. \tag{3.11}$$

Lemma 3.2 implies that $\widetilde{x} \neq 0$. Multiplying Eq (3.11) by $\frac{\widetilde{x}^T}{\widetilde{x}^T\widetilde{x}}$ on both sides, we obtain

$$(\lambda - 1)\frac{\widetilde{x}^T P\widetilde{x}}{\widetilde{x}^T\widetilde{x}} + \lambda\frac{\widetilde{x}^T A\widetilde{x}}{\widetilde{x}^T\widetilde{x}} + \frac{\frac{\widetilde{x}^T B^T Q^{-1} B\widetilde{x}}{\widetilde{x}^T\widetilde{x}}}{\lambda - 1} = 0. \tag{3.12}$$

Together with (3.9) and (3.12), we have

$$\lambda^2 - \frac{2a+b}{a+b}\lambda + \frac{a+c}{a+b} = 0. \tag{3.13}$$

Here, $a > 0$, $b > 0$, and $c \geq 0$. By Lemma 3.3, we know $|\lambda| < 1$ in (3.13) is that

$$\left|\frac{a+c}{a+b}\right| < 1 \tag{3.14}$$

and

$$\left|\frac{2a+b}{a+b}\right| < 1 + \left|\frac{a+c}{a+b}\right|. \tag{3.15}$$

Thus, it is easy to confirm that (3.14) and (3.15) are valid for $c < b$ when $c > 0$. When $c = 0$, since Q is symmetric positive definite, there exists a nonzero vector $\widetilde{x}$ with the property $B\widetilde{x} = 0$. From (3.10), it follows that $\widetilde{y} = 0$, and Lemma 3.2 then implies $0 < \lambda < 1$. Consequently, we have $\rho(\Gamma) < 1$, which completes the proof. □

Furthermore, since Q is symmetric positive definite, denoting by $\delta_{\min}$ the minimum eigenvalue of A and by $\mu_{\max}$ the maximum singular value of $Q^{-1/2}B$, we obtain

$$\frac{\widetilde{x}^T B^T Q^{-1} B\widetilde{x}}{\widetilde{x}^T A\widetilde{x}} \leq \frac{\mu_{\max}^2}{\delta_{\min}}. \tag{3.16}$$

Consequently, the following results follow from Eq (3.16).

Corollary 3.1. *Suppose that the hypotheses of Theorem 3.1 are satisfied. Assume that δ_{min} and μ_{max} are the minimum eigenvalue of A and the maximum singular value of $Q^{-1/2}B$, respectively. If $\frac{\mu_{max}^2}{\delta_{min}} < 1$, then*

$$\rho(\Gamma) < 1.$$

That is, the new iterative method (3.2) is convergent.

Remark 3.1. *The condition $\frac{\mu_{max}^2}{\delta_{min}} < 1$ can be achieved by a special selection of the matrix Q, owing to the fact that μ_{max} is defined as the maximum singular value of $Q^{-1/2}B$.*

Furthermore, since A is symmetric positive definite, we let $\widetilde{y} = A^{1/2}\widetilde{x}$ and $\|\widetilde{y}\|_2 = 1$. Thus, we have

$$\frac{\widetilde{x}^T B^T Q^{-1} B\widetilde{x}}{\widetilde{x}^T A\widetilde{x}} = \widetilde{y}^T A^{-1/2} B^T Q^{-1} B A^{-1/2}\widetilde{y} = \widetilde{y}^T (Q^{-1/2} B A^{-1/2})^T Q^{-1/2} B A^{-1/2}\widetilde{y}.$$

Consequently, we establish the following results.

Corollary 3.2. *Suppose that the hypotheses of Theorem 3.1 are satisfied and the maximum singular value θ_{max} of $Q^{-1/2}BA^{-1/2}$ satisfies $\theta_{max} < 1$, then*

$$\rho(\Gamma) < 1.$$

This is sufficient to ensure that the new iterative method (3.2) is convergent.

In summary, the new iterative method (3.2) is convergent under the conditions stated in Theorem 3.1, Corollary 3.1, and Corollary 3.2. When used directly as a stationary iterative method (i.e., a fixed-point iterative), the splitting scheme $\mathcal{A} = \mathcal{P} - \mathcal{R}$ commonly yields a slow convergence rate. However, the splitting matrix $\mathcal{P}$ itself can serve as an effective preconditioner for the coefficient matrix $\mathcal{A}$ within a Krylov subspace method. In other words, although the fixed-point iterative converges slowly, it provides a preconditioner that significantly accelerates the convergence of Krylov solvers.

4. The block diagonal preconditioner

We now analyze the following block diagonal preconditioner (denote to "PHSS") for (1.1), which originates from the special matrix splitting scheme [7]:

$$\mathcal{P}_{PHSS} = \begin{pmatrix} P + A & O \\ O & Q \end{pmatrix}. \tag{4.1}$$

As is known that the saddle point problem (1.1) can be reformulated as the left-preconditioned linear system

$$\mathcal{P}_{PHSS}^{-1}\mathcal{A}u = \mathcal{P}_{PHSS}^{-1}f.$$

It is a well-established fact that eigenvalue clustering of the preconditioned matrix typically results in a fast convergence rate. Given this, we now examine the spectral properties of $\mathcal{P}_{PHSS}^{-1}\mathcal{A}$, as they directly inform the convergence behavior.

Remark 4.1. *The proposed block diagonal preconditioner defined as in (4.1) can be viewed as a block diagonal version of works introduced in [28, 29].*

Remark 4.2. *The proposed PHSS preconditioner is particularly advantageous, as it not only includes the SHSS preconditioner as a special case, but also yields new block diagonal preconditioners. As a result, the PHSS preconditioner offers greater flexibility, and therefore enables significantly faster convergence than alternative approaches. Moreover, we can choose suitable Q such that the proposed PHSS preconditioner reduces the overall computational cost. We emphasize that the addition of the symmetric positive definite matrices P and Q is not arbitrary modification; instead, it is inherited from the regularization concept of [7]. Although for general linear systems, additive correction on the (1,1)-block often fails to accelerate iteration; for saddle point matrices with a zero (2,2)-block, this structured regularization can effectively adjust the spectrum of the preconditioned matrix.*

Remark 4.3. *As is known, the optimal block diagonal preconditioner takes the form*

$$\mathcal{P}_{opt} = \begin{pmatrix} A & O \\ O & \widehat{S} \end{pmatrix},$$

where $\widehat{S} = BA^{-1}B^T$. It is well known that the preconditioned system with the preconditioner $\mathcal{P}_{opt}$ possesses only three distinct eigenvalues, and achieves the convergence of generalized minimal residual (GMRES) in four iterations. Nevertheless, the preconditioner $\mathcal{P}_{opt}$ relies on the exact inverse of A and the full Schur complement, which is computationally unaffordable for large systems. In this

paper, from a computational perspective, the extra matrices P and Q provide flexible approximation choices to reduce the computational cost. Moreover, the proposed block diagonal preconditioner is simper and more feasible to implement in comparison with several existing preconditioners. It is therefore expected to outperform others in terms of the computational cost.

Theorem 4.1. *Suppose $A, P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric positive definite, and $B \in \mathbb{R}^{m\times n}$ has full row rank. Let η denote an eigenvalue of $\mathcal{P}_{PHSS}^{-1}\mathcal{A}$, with $[u^T, v^T]^T$ being its associated eigenvector. If $\overline{c} < \overline{b}$, then we have $|\eta| < 1$. Here,*

$$\overline{a} = \frac{u^T P u}{u^T u} > 0,\ \overline{b} = \frac{u^T A u}{u^T u} > 0 \text{ and } \overline{c} = \frac{u^T B^T Q^{-1} B u}{u^T u} \geq 0. \tag{4.2}$$

Proof. First we derive

$$\begin{bmatrix} A & B^T \\ -B & O \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \eta \begin{bmatrix} P+A & O \\ O & Q \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix},$$

or equivalently,

$$Au + B^T v = \eta(P + A)u, \tag{4.3}$$

$$-Bu = \eta Q v. \tag{4.4}$$

Lemma 3.1 implies that $\eta \neq 0$. Then, Eq (4.4) leads to

$$v = \frac{-Q^{-1}Bu}{\eta}. \tag{4.5}$$

Thus, substituting (4.5) into (4.3) results

$$Au - \frac{B^T Q^{-1} B u}{\eta} = \eta(P + A)u. \tag{4.6}$$

In the following part, we set $\|u\|_2 = 1$. Multiplying $\frac{u^T}{u^T u}$ on both sides of Eq (4.6), based on (4.2), we derive

$$\eta^2 - \frac{\overline{b}}{\overline{a} + \overline{b}}\eta + \frac{\overline{c}}{\overline{a} + \overline{b}} = 0. \tag{4.7}$$

Based on Lemma 3.3 and (4.7), it is easy to derive that $|\eta| < 1$. □

Remark 4.4. *As described in Theorems 3.1 and 4.1, the validity of the theoretical results relies on certain assumptions regarding the preconditioned matrix Q. In practice, Q is chosen to be symmetric positive definite. Moreover, when A is symmetric positive definite, the required condition $Q > BA^{-1}B^T$ is equivalent to the positive definiteness of the Schur complement $A - B^T Q^{-1} B$. Therefore, verifying $A - B^T Q^{-1} B > 0$ provides a criterion such that the assumptions holds. This highlights that the choice of Q plays a critical role in the performance of the PHSS preconditioner. In the numerical example, we consider different types of the matrix Q to highlight its importance of the proposed preconditioner.*

Given that the preconditioned matrices P and Q are unknown, a general analysis of the eigenvalue distribution for $\mathcal{P}_{PHSS}^{-1}\mathcal{A}$ is inherently difficult. This very freedom, however, allows for the selection of P and Q to purposefully cluster the eigenvalues, thereby improving the spectral properties of the preconditioned system. Optimizing this eigenvalue distribution is fundamental to achieving faster convergence in iterative methods.

5. Numerical results

In this part, numerical tests are conducted to evaluate the performance of the proposed block diagonal preconditioner. To highlight its efficiency, we compare numerical results with those from several existing preconditioners. The comparison is based on the following criteria: one is the number of iterations ("IT"), and the other is the elapsed CPU time in seconds ("CPU"). All involved preconditioners are employed to preconditioned GMRES method. We begin with a zero initial vector and the iterative stops if

$$\frac{\|r^k\|_2}{\|r^0\|_2} \leq 10^{-7}$$

or $\kappa_{\max} = 1000$ is exceeded. Here, r^k represents the kth iteration residual.

We compare with the following existing preconditioners:

- The relaxed HSS preconditioner [22];
- The new relaxed HSS preconditioner [23];
- The modified generalized shift-splitting (MGSS) preconditioner [29].

Here, the involved RHSS and REHSS preconditioners are given in (2.3) and (2.4). Additionally, the MGSS preconditioner has the form of

$$\mathcal{P}_{MGSS} = \frac{1}{2}\begin{bmatrix} P+A & B^T \\ -B & Q \end{bmatrix}.$$

We performed the computations in MATLAB (R2024a) equipped with an Intel(R) Core(TM) i7-14700 (2.10 GHz) with 32GB RAM. Moreover, the vector f in (1.1) is set so that it corresponds to an exact solution of $(1, 1, ..., 1)^T$.

Example 5.1. *[30, 31] Consider the Stokes problem*

$$\begin{cases} -\Delta \boldsymbol{u} + \nabla \boldsymbol{p} = \boldsymbol{f}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{cases} \tag{5.1}$$

in $\Omega = [-1, 1] \times [-1, 1]$. *Here,* Δ, ∇, $\boldsymbol{u}$, *and* $\boldsymbol{p}$ *denote, respectively, the Laplace operator, the gradient operator, velocity, and pressure of the fluid, coupled with suitable boundary condition on* $\partial\Omega$. *As is known, many discretization schemes for (5.1) lead to the saddle point structure (1.1). Here, we employ* $Q2 - P1$ *finite element to discretize on uniform grids on the unit square of the two standard model problems:*

- *Case I: The leaky lid-driven cavity problem;*
- *Case II: The channel domain problem.*

For more details, we refer to [30, 31].

In this example, the linear system was generated through the Incompressible Flow & Iterative Solver Software (IFISS) package developed by Elman et al. [30]. Four different sizes are chosen as 16×16, 32×32, 64×64, and 128×128. Here, the IFISS software provides the involved matrices *Ast* and *Bst* for A and B of (1.1), respectively. In particular, the matrix *Bst* is of full rank for the Case II, while is

rank deficient for the Case I. Therefore, in Case I, we obtain a full rank matrix by dropping the two first rows of Bst.

The preconditioning matrices of the PHSS and MGSS preconditioner are assigned as follows:

- PHSS preconditioner: $P = \alpha H$ and $Q = \alpha I$, where $H = 1/2(A + A^T)$;
- MGSS preconditioner: $P = \alpha H$ and $Q = \alpha I + \alpha BB^T$.

It should be pointed out that the optimal parameter is difficult for us now. Instead, in order to ensure a fair and consistent comparison, numerical results are compared for parameters $\alpha = 0.01, 0.1, 1$ and different grids.

Table 1. The size of the matrices A and B for different cases.

	Case I				Case II			
Grid	n	m	$nnz(A)$	$nnz(B)$	n	m	$nnz(A)$	$nnz(B)$
16×16	578	190	6178	1967	578	192	6698	2084
32×32	2178	766	28418	9868	2178	768	29546	10142
64×64	8450	3070	122206	44516	8450	3072	124550	45062
128×128	33282	12286	506376	191084	33282	12288	511152	192174

Table 1 presents some basic information of (5.1), including n, m, $nnz(A)$, and $nnz(B)$. Here, $nnz(\cdot)$ denotes the number of the nonzero entries of a given matrix. Numerical results for Cases I and II are presented in Tables 2 and 3. As shown in Tables 2 and 3, the proposed preconditioner $\mathcal{P}_{PHSS}$ consistently requires less computational time than those of others. These observations indicate the computational superiority of $\mathcal{P}_{PHSS}$ in accelerating the convergence of iterative solvers.

Table 2. Numerical results for Case I on $2^l \times 2^l$ grid.

		$\alpha = 0.01$		$\alpha = 0.1$		$\alpha = 1$	
l	Preconditioner	IT	CPU	IT	CPU	IT	CPU
	$\mathcal{P}_{RHSS}$	25	0.1117	27	0.1246	27	0.1233
4	$\mathcal{P}_{REHSS}$	27	0.1139	24	0.1037	22	0.0965
	$\mathcal{P}_{MGSS}$	11	0.0300	23	0.0626	34	0.0906
	$\mathcal{P}_{PHSS}$	45	0.0328	45	0.0332	41	0.0301
	$\mathcal{P}_{RHSS}$	37	0.7237	41	0.7777	44	0.8314
5	$\mathcal{P}_{REHSS}$	31	0.6192	25	0.4807	22	0.4387
	$\mathcal{P}_{MGSS}$	17	0.2189	28	0.3616	35	0.4608
	$\mathcal{P}_{PHSS}$	47	0.2040	43	0.1899	40	0.1731
	$\mathcal{P}_{RHSS}$	58	7.6520	62	8.3970	68	8.3944
6	$\mathcal{P}_{REHSS}$	28	3.4705	24	2.9625	21	2.8392
	$\mathcal{P}_{MGSS}$	24	1.6958	31	2.2311	34	2.6570
	$\mathcal{P}_{PHSS}$	47	0.8888	43	0.8148	38	0.7208
	$\mathcal{P}_{RHSS}$	87	62.9047	98	74.4238	106	93.9496
7	$\mathcal{P}_{REHSS}$	26	19.2951	23	16.6009	8	6.9093
	$\mathcal{P}_{MGSS}$	28	9.8991	31	10.1461	14	5.5942
	$\mathcal{P}_{PHSS}$	47	4.1319	41	3.6522	16	1.5359

Table 3. Numerical results for Case II on $2^l \times 2^l$ grid.

		$\alpha = 0.01$		$\alpha = 0.1$		$\alpha = 1$	
l	Preconditioner	IT	CPU	IT	CPU	IT	CPU
4	$\mathcal{P}_{RHSS}$	26	0.1316	27	0.1354	27	0.1361
	$\mathcal{P}_{REHSS}$	25	0.1305	20	0.1051	19	0.0958
	$\mathcal{P}_{MGSS}$	9	0.0244	19	0.0556	30	0.0736
	$\mathcal{P}_{PHSS}$	43	0.0351	41	0.0335	37	0.0298
5	$\mathcal{P}_{RHSS}$	39	0.8252	41	0.8672	40	0.8439
	$\mathcal{P}_{REHSS}$	26	0.5676	20	0.4290	19	0.4052
	$\mathcal{P}_{MGSS}$	14	0.1906	25	0.3315	30	0.3433
	$\mathcal{P}_{PHSS}$	43	0.1869	39	0.1717	35	0.1490
6	$\mathcal{P}_{RHSS}$	60	8.9139	57	7.4187	57	7.1830
	$\mathcal{P}_{REHSS}$	22	2.8906	20	2.6750	17	2.4816
	$\mathcal{P}_{MGSS}$	20	1.4302	26	1.8767	28	2.1367
	$\mathcal{P}_{PHSS}$	41	0.8329	37	0.7366	31	0.6377
7	$\mathcal{P}_{RHSS}$	83	61.4040	80	61.2539	80	59.4119
	$\mathcal{P}_{REHSS}$	22	15.8820	18	13.1340	14	10.9738
	$\mathcal{P}_{MGSS}$	23	8.6719	24	9.4720	23	9.1592
	$\mathcal{P}_{PHSS}$	39	3.4066	32	2.7854	26	2.3738

Figures 1 and 2 present the spectral distribution of $\mathcal{A}$ and the preconditioned matrices for Case I on 32×32 grid, using different parameter values $\alpha = 0.1, 1$. From Figures 1 and 2, it can be readily observed that the preconditioned matrices exhibit significantly greater eigenvalue clustering than $\mathcal{A}$. In particular, all eigenvalues of $\mathcal{P}_{PHSS}^{-1}\mathcal{A}$ lie within $(0, 1)$.

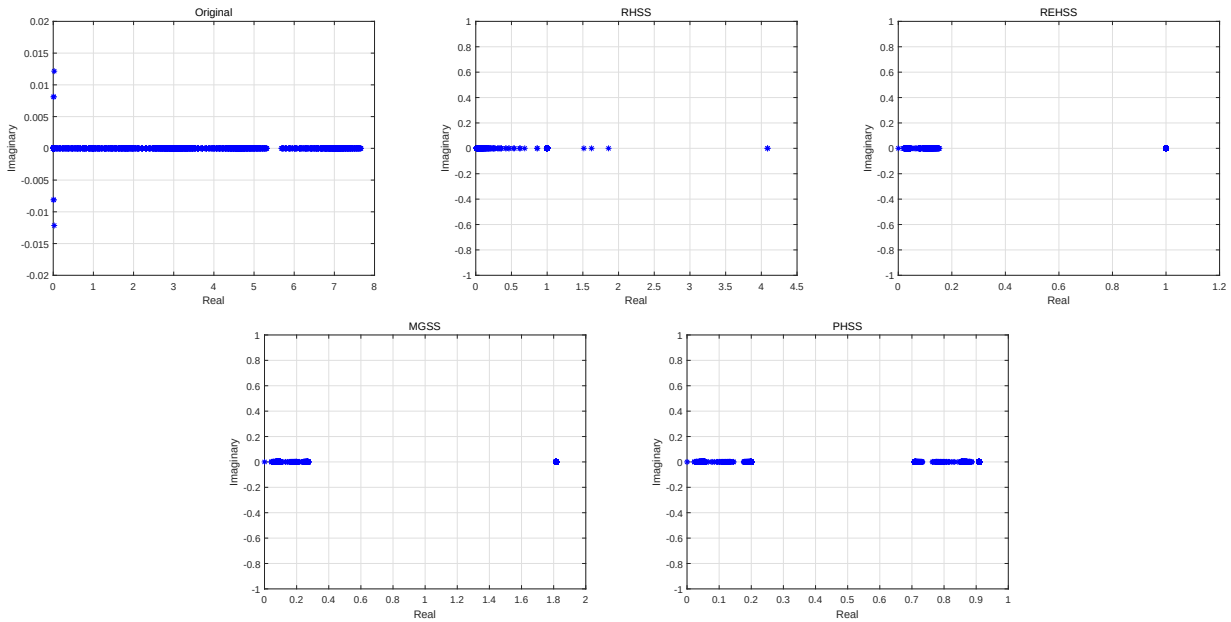

Figure 1. Spectrum of the preconditioned matrices for Case I on 32×32 grid ($\alpha = 0.1$).

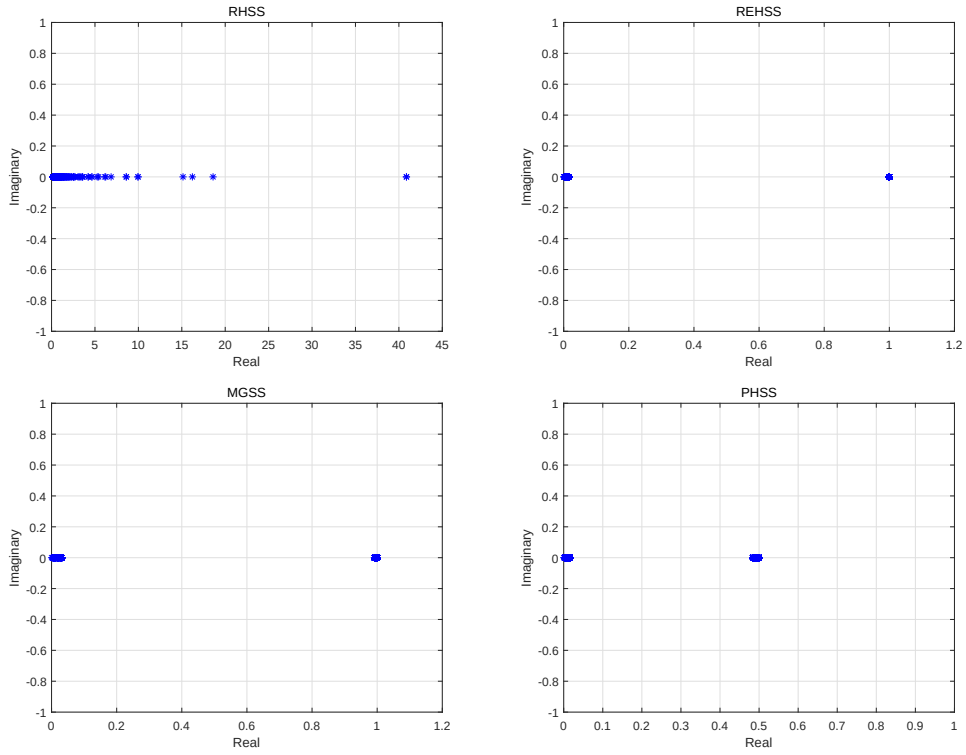

Figure 2. Spectrum of the preconditioned matrices for Case I on 32×32 grid ($\alpha = 1$).

Figures 3 and 4 present the spectral distributions of $\mathcal{A}$ and several preconditioned matrices for Case II on 32×32 grid with $\alpha = 0.1$ and $\alpha = 1$, respectively. From Figures 3 and 4, the PHSS preconditioned matrices generally exhibit a well-clustered spectrum. However, an exception is observed when $\alpha = 0.1$.

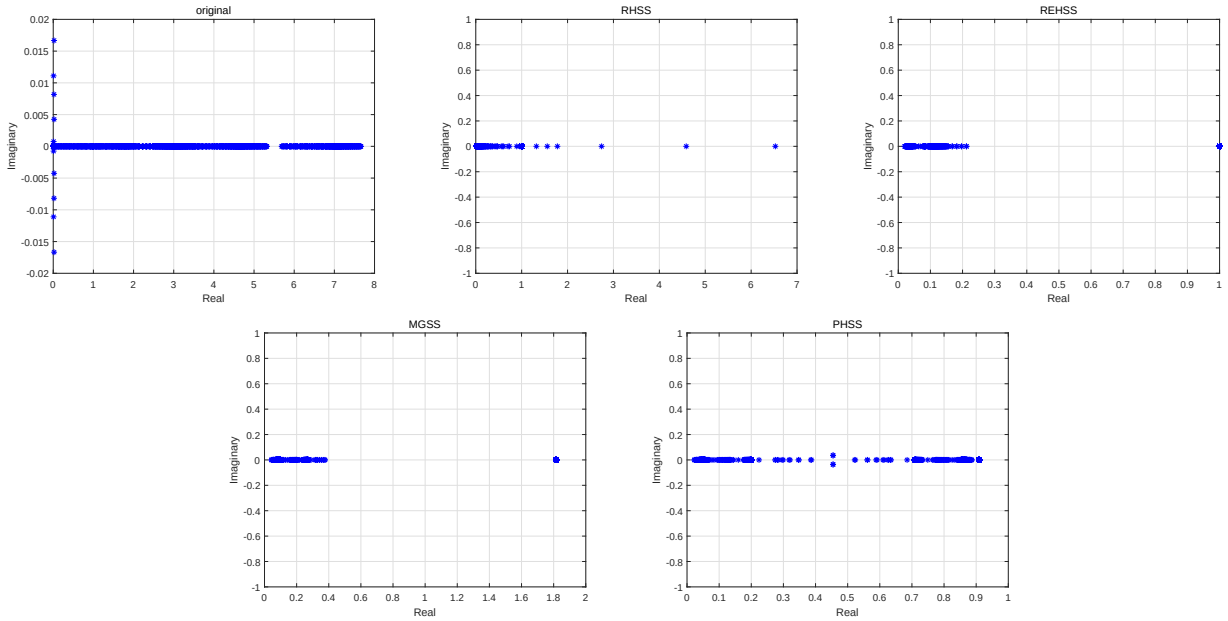

Figure 3. Spectrum of the preconditioned matrices for Case II on 32×32 grid ($\alpha = 0.1$).

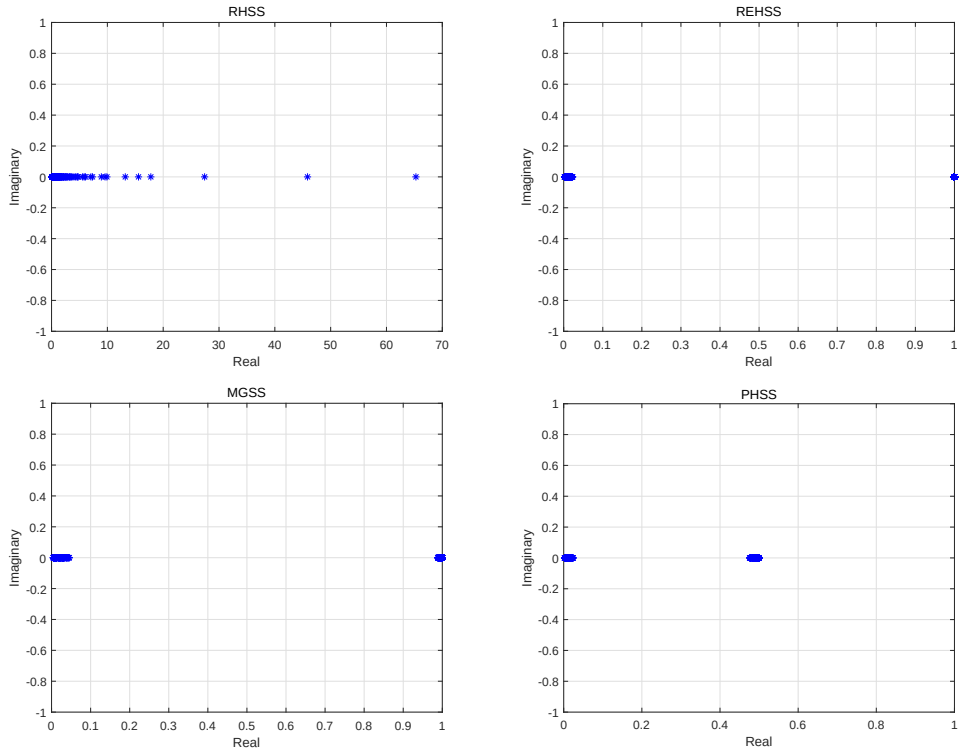

Figure 4. Spectrum of the preconditioned matrices for Case II on 32×32 grid ($\alpha = 1$).

To demonstrate the advantage of the proposed block diagonal preconditioner with varying α, we illustrate that the proposed PHSS preconditioner outperforms several existing methods. We also consider the case of $P = \alpha H$ and $Q = I$ (denote as "PHSS-I"). Figures 5 and 6 present the number of iterations and the CPU time of GMRES method for solving the preconditioned systems with the preconditioners $\mathcal{P}_{RHSS}, \mathcal{P}_{REHSS}, \mathcal{P}_{MGSS}, \mathcal{P}_{PHSS}$ and its variant PHSS-I for Cases I and II with 64×64 and 128×128 grids for varying α. Figures 5 and 6 show that the PHSS preconditioners possess less iteration counts compared to the RHSS preconditioner. Although we see that the REHSS preconditioner and MGSS preconditioner are superior to the proposed PHSS preconditioner in terms of the iteration counts, the PHSS preconditioners are more effective in terms of the CPU time. Therefore, it is expected that the proposed block diagonal preconditioner may perform well when is used to solve large-scale problems. Furthermore, we observe that $Q = I$ is a suitable choice for our proposed block diagonal preconditioner. This highlights the flexibility of the proposed PHSS preconditioner.

Furthermore, we consider the following different choices of Q, such as

- $Q = B\Lambda^{-1}B^T$, where $\Lambda = \text{diag}(A)$;
- $Q = \beta BB^T$;
- $Q = \alpha I + \beta BB^T$.

Here, we set $\alpha = 1$ and $\beta = 10$. Table 4 presents the numerical results for different Q. It is evident that the preconditioning matrix Q is vital to the PHSS preconditioner, as a suitable choice leads to better performance.

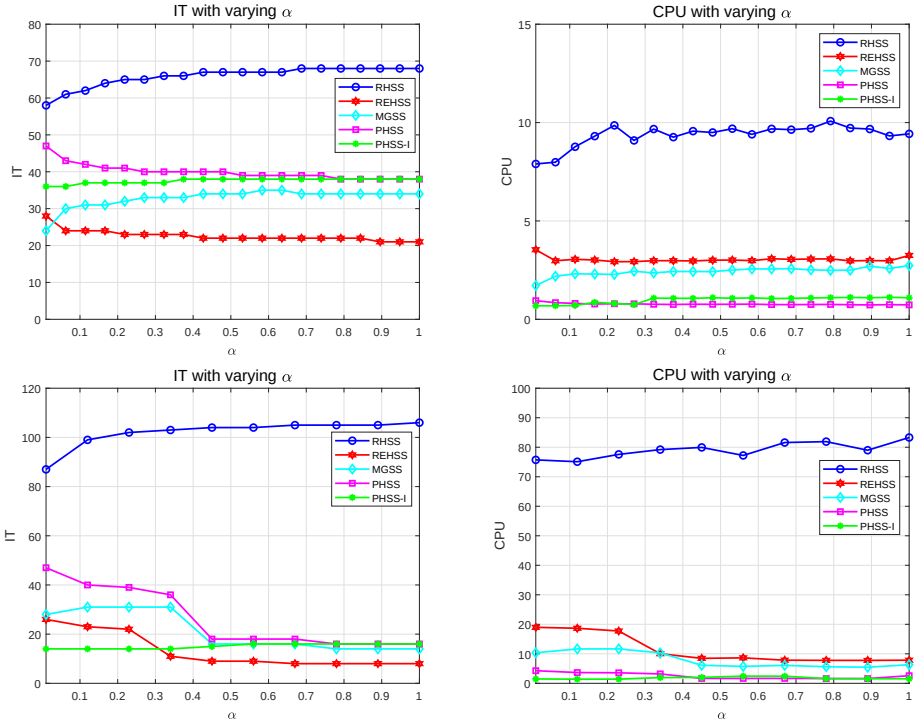

Figure 5. Number of iterations and CPU time with varying α for Case I (Top: 64×64; Bottom: 128×128).

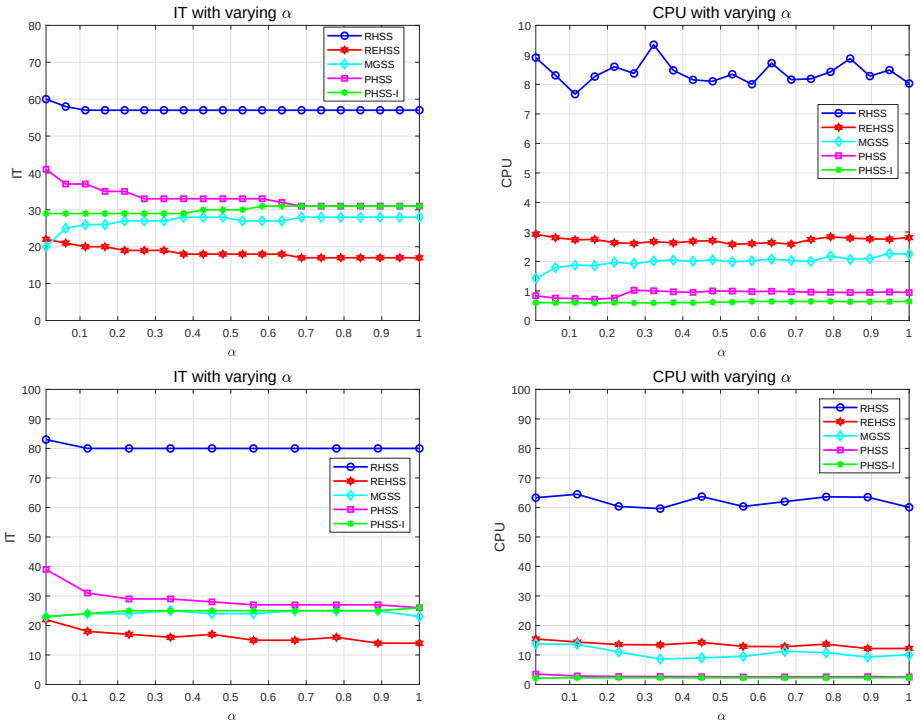

Figure 6. Number of iterations and CPU time with varying α for Case II (Top: 64×64; Bottom: 128×128).

Table 4. Numerical results for different Q on $2^l \times 2^l$ grid.

Q		$l = 4$	$l = 5$	$l = 6$	$l = 7$
$Q = B\Lambda^{-1}B^T$	IT	49	73	103	141
	CPU	0.0535	0.4573	3.0498	20.5831
$Q = \beta BB^T$	IT	51	75	105	143
	CPU	0.0566	0.4719	3.1119	20.9957
$Q = \alpha I + \beta BB^T$	IT	35	34	31	26
	CPU	0.0340	0.2184	0.9485	3.8557

6. Conclusions

For the solution of saddle point problems, we propose a block diagonal preconditioner. The proposed approach not only generalizes existing preconditioners in the literature but also induces new ones simultaneously. To this end, we further investigate the convergence performance of the resulting iterative method as well as its related spectral properties, with a particular focus on eigenvalue clustering. The advantages of the proposed preconditioner are validated through numerical experiments in comparison with those of existing methods. Future work will be dedicated to addressing the critical issue of selecting effective matrices P and Q.

Use of AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

This work was supported by the 2024 Scientific Research Start-up Fund of Hunan First Normal University (Grant No. 09240006).

Conflict of interest

The author declares there is no conflict of interest.

References

1. Z. Bai, Structured preconditioners for nonsingular matrices of block two-by-two structures, *Math. Comp.,* **75** (2006), 791–815. https://doi.org/10.1090/S0025-5718-05-01801-6
2. Z. Bai, J. Pan, *Matrix Analysis and Computations*, Society for Industrial and Applied Mathematics, 2021. https://doi.org/10.1137/1.9781611976632
3. 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
4. Z. Bai, A two-step matrix splitting iteration paradigm based on one single splitting for solving systems of linear equations, *Numer. Linear Algebra Appl.*, **31** (2024), 2510. https://doi.org/10.1002/nla.2510

5. 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** (2003), 603–626. https://doi.org/10.1137/S0895479801395458

6. C. Li, S. Wu, A single-step HSS method for non-Hermitian positive definite linear systems, *Appl. Math. Lett.*, **44** (2015), 26–29. https://doi.org/10.1016/j.aml.2014.12.013

7. X. Wang, X. Xiao, Q. Zheng, A single-step iteration method for non-Hermitian positive definite linear systems, *J. Comput. Appl. Math.*, **346** (2019), 471–482. https://doi.org/10.1016/j.cam.2018.07.021

8. M. Benzi, A. J. Wathen, Some preconditioning techniques for saddle point problems, in *Model Order Reduction: Theory, Research Aspects and Applications*, **13** (2008), 195–211. https://doi.org/10.1007/978-3-540-78841-6_10

9. F. P. A. Beik, M. Benzi, S. A. Chaparpordi, On block diagonal and block triangular iterative schemes and preconditioners for stabilized saddle point problems, *J. Comput. Appl. Math.*, **326** (2017), 15–30. https://doi.org/10.1016/j.cam.2017.05.009

10. B. S. Southworth, A. A. Sivas, S. Rhebergen, On fixed-point, Krylov, and 2×2 block preconditioners for nonsymmetric problems, *SIAM J. Matrix Anal. Appl.*, **41** (2020), 871–900. https://doi.org/10.1137/19M1298317

11. E. de Sturler, J. Liesen, Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. Part I: Theory, *SIAM J. Sci. Comput.*, **26** (2005), 1598–1619. https://doi.org/10.1137/S1064827502411006

12. S. H. A. Chaparpordi, F. P. A. Beik, D. K. Salkuyeh, Block triangular preconditioners for stabilized saddle point problems with nonsymmetric (1,1)-block, *Comput. Math. Appl.*, **76** (2018), 1544–1553. https://doi.org/10.1016/j.camwa.2018.07.006

13. A. Klawonn, G. Starke, Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis, *Numer. Math.*, **81** (1999), 577–594. https://doi.org/10.1007/s002110050405

14. X. Xiao, C. Wang, A block upper trangular preconditioner with two parameters for saddle-point problems, *Comput. Math. Appl.*, **160** (2024), 15–29. https://doi.org/10.1016/j.camwa.2024.02.015

15. Z. Bai, M. K. Ng, Z. Wang, Constraint preconditioners for symmetric indefinite matrices, *SIAM J. Matrix Anal. Appl.*, **31** (2009), 410–433. https://doi.org/10.1137/080720243

16. C. Keller, N. I. M. Gould, A. J. Wathen, Constraint preconditioning for indefinite linear systems, *SIAM J. Matrix Anal. Appl.*, **21** (2000), 1300–1317. https://doi.org/10.1137/S0895479899351805

17. H. Wu, A general class of constraint preconditioners for generalized saddle point linear systems, *Appl. Math. Comput.*, **489** (2025), 129148. https://doi.org/10.1016/j.amc.2024.129148

18. X. Shao, H. Meng, A new generalized variant of the deteriorated PSS preconditioner for nonsymmetric saddle point problems, *Comput. Math. Appl.*, **118** (2022), 208–213. https://doi.org/10.1016/j.camwa.2022.05.030

19. N. Wang, J. Li, A class of preconditioners based on symmetric-triangular decomposition and matrix splitting for generalized saddle point problems, *IMA J. Numer. Anal.*, **43** (2023), 2998–3025. https://doi.org/10.1093/imanum/drac061

20. H. Wang, N. Zhang, On the NESS iteration method for singular saddle point problems, *J. Comput. Appl. Math.*, **476** (2026), 117082. https://doi.org/10.1016/j.cam.2025.117082

21. 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

22. Y. Cao, L. Yao, M. Jiang, Q. Niu, A relaxed HSS preconditioner for saddle point problems from meshfree discretization, *J. Comput. Math.*, **31** (2013), 398–421. https://doi.org/10.4208/jcm.1304-m4209

23. D. K. Salkuyeh, M. Masoudi, A new relaxed HSS preconditioner for saddle point problems, *Numer. Algor.*, **74** (2017), 781–795. https://doi.org/10.1007/s11075-016-0171-2

24. Z. Bai, M. Benzi, Regularized HSS iteration methods for saddle-point linear systems, *BIT Numer. Math.*, **57** (2017), 287–311. https://doi.org/10.1007/s10543-016-0636-7

25. Q. Liu, An alternating LHSS preconditioner for saddle point problems, *Comput. Appl. Math.*, **31** (2012), 339–352. https://doi.org/10.1590/S1807-03022012000200007

26. C. Li, S. Wu, The SHSS preconditioner for saddle point problems, *J. Appl. Anal. Comput.*, **13** (2023), 3221–3230. https://doi.org/10.11948/20220552

27. D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, 1971. https://doi.org/10.1016/C2013-0-11733-3

28. Q. Zheng, L. Lu, Extended shift-splitting preconditioners for saddle point problems, *J. Comput. Appl. Math.*, **313** (2017), 70–81. https://doi.org/10.1016/j.cam.2016.09.008

29. D. K. Salkuyeh, M. Rahimian, A modification of the generalized shift-splitting method for singular saddle point problems, *Comput. Math. Appl.*, **74** (2017), 2940–2949. https://doi.org/10.1016/j.camwa.2017.07.029

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

31. H. C. Elman, D. J. Silvester, A. J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2005. https://doi.org/10.1093/oso/9780198528678.001.0001

AIMS Press