Received: 25 September 2021 | Revised: 18 October 2022 | Accepted: 29 October 2022

DOI: 10.1002/nla.2478

RESEARCH ARTICLE

WILEY

A parameterized extended shift-splitting preconditioner for nonsymmetric saddle point problems

Seryas Vakili[1] | **Ghodrat Ebadi**[1,2] | **Cornelis Vuik**[3]

[1]Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

[2]Research Department of Computational Algorithms and Mathematical Models, University of Tabriz, Tabriz, Iran

[3]Delft University of Technology, Delft Institute of Applied Mathematics, Delft, The Netherlands

Correspondence
Ghodrat Ebadi, Faculty of Mathematical Sciences, University of Tabriz, Tabriz 51666-14766, Iran.
Email: ghodrat_ebadi@yahoo.com

Abstract

In this article, a parameterized extended shift-splitting (PESS) method and its induced preconditioner are given for solving nonsingular and nonsymmetric saddle point problems with nonsymmetric positive definite (1,1) part. The convergence analysis of the *PESS* iteration method is discussed. The distribution of eigenvalues of the preconditioned matrix is provided. A number of experiments are given to verify the efficiency of the *PESS* method for solving nonsymmetric saddle-point problems.

KEYWORDS

convergence, preconditioning, saddle point problem, shift-splitting

1 | INTRODUCTION

Consider the following nonsymmetric saddle point problem:

$$\mathfrak{A}u = \begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} f \\ -g \end{pmatrix} \equiv b, \tag{1}$$

where $B \in \mathbb{R}^{m\times n}$ has full column rank with $m \geq n, A \in \mathbb{R}^{m\times m}$ is nonsymmetric positive definite. These hypotheses ensure the existence and uniqueness of the solution of (1). The matrices A and B are also large and sparse, $f \in \mathbb{R}^m$ and $g \in \mathbb{R}^n$. In various engineering applications, solving linear system (1) is required, such as networks computer graphics, optimal control, and computational fluid dynamics; see References 1 and 2 and references therein.

For large and sparse matrices A and B, iterative methods are better than direct methods to solve saddle point problems. There are two options for matrix B. First, if B is a rank deficient matrix, matrix $\mathfrak{A}$ is singular and (1) is a singular saddle point problem. A number of useful stationary iterative methods are proposed for solving singular saddle point problems in recent years. For instance, Uzawa-type,[3,4] HSS-type,[5-7] SOR-type methods,[8,9] also efficient preconditioners are known to accelerate the convergence of Krylov subspace methods (such as, shift-splitting preconditioners).[10-19]

Second, the coefficient matrix $\mathfrak{A}$ is nonsingular if B in (1) has full column rank. This type of problems, which is called the nonsingular saddle point problems, can be solved by iterative techniques, such as Uzawa-type methods,[2,20,21] parameterized inexact Uzawa (PIU) methods,[22,23] SOR-type method,[21,24-26] Hermitian and skew-Hermitian method.[27-31] Since A and B are large and sparse matrices, linear system (1) can be solved by a Krylov subspace method.[32,33] The Krylov subspace methods tend to converge slowly and good preconditioners are required to achieve fast convergence.[34,35] Recently, very efficient preconditioners have been studied, such as constrained preconditioners,[36,37] structured preconditioners,[1] HSS-based preconditioners,[27,29,38-40] deteriorated positive definite and skew-Hermitian splitting (DPSS) preconditioner,[41,42] dimensional split (DS) preconditioner,[43,44] block definite, indefinite and triangular preconditioners,[45-50] and shift splitting preconditioners.[16-19,51-56]

Bai et al.[19] presents a shift-splitting preconditioner for solving non-Hermitian positive definite linear system. Based on the shift splitting of Bai et al.,[19] Cao et al.[51] gave the *SS* preconditioner as

$$\mathcal{P}_{SS} = \frac{1}{2}\begin{pmatrix} \hat{\alpha}I + A & B \\ -B^T & \hat{\alpha}I \end{pmatrix},$$

where $\hat{\alpha} \geq 0$ and I represents the identity matrix. The saddle point problem is solved through efficient shift-splitting and its generalized preconditioners. Chen and Ma[52] and Cao et al.[53] introduced the generalized shift-splitting (*GSS*) preconditioner for nonsymmetric saddle point problem with replacing $\hat{\beta}$ instead of $\hat{\alpha}$ in the last block of $\mathcal{P}_{SS}$. In addition, $\mathcal{P}_{SS}$ is a special case of the *GSS* preconditioner when $\hat{\alpha} = \hat{\beta}$. Numerical experiments in References 53 and 52 show that the *GSS* preconditioner has a better performance compared to the *SS* preconditioner. Cao et al.[54] presented an efficient preconditioned generalized shift-splitting (*PGSS*) iteration for saddle point problems with symmetric (1, 1)-block and established its unconditional convergence. In order to solve a nonsymmetric saddle point problem, Zhou et al.[55] developed a modified shift-splitting (*MSS*) preconditioner. Then, Huang et al.[16] replaced α by parameter β in (2,2)-block of the *MSS* and introduced the generalized *MSS* (GMSS) preconditioner. Huang and Su[56] employed a modified *SS* preconditioner (*MSSP*) to accelerate the convergence of the GSS method for solving saddle point problem that has a symmetric, positive definite (1,1)-block. A modified *GSS* (*MGSS*) preconditioner was obtained for solving nonsymmetric saddle point problems by Huang et al.[17] Numerical experiments in Reference 17 show that the *MGSS* preconditioner is more efficient in terms of runtime and number of iterations.

Zheng et al.[57] introduced an extended shift-splitting (*ESS*) preconditioner

$$\mathcal{P}_{ESS} = \frac{1}{2}(\Omega_Q + \mathfrak{A}) = \frac{1}{2}\begin{pmatrix} Q_1 + A & B \\ -B^T & Q_2 \end{pmatrix},$$

for symmetric saddle point problems, where $\Omega_Q = \text{diag}(Q_1, Q_2), Q_1 \in \mathbb{R}^{n\times n}$ and $Q_2 \in \mathbb{R}^{m\times m}$ are symmetric positive definite. Salkuyeh et al.[11] considered a MGSS method for singular saddle point problems. Spectral analysis of the *MGSS* preconditioner was given in Reference 12. The *ESS* was studied for both the singular and nonsingular nonsymmetric generalized saddle point problem in Reference 58. Huang et al.[18] recently utilized constant l within the matrix $\mathfrak{A}$ to introduce the following parameterized *GSS* preconditioner (*PGSS*)

$$\mathfrak{A} = \mathcal{P}_{PGSS} - \mathcal{Q}_{PGSS} = \begin{pmatrix} \hat{\alpha}I + lA & lB \\ -lB^T & \hat{\beta}I \end{pmatrix} - \begin{pmatrix} lA + \hat{\alpha}I - A & lB - B \\ -(l-1)B^T & \hat{\beta}I \end{pmatrix},$$

such that $\hat{\alpha} \geq 0, \hat{\beta} > 0$. Wang et al.[59] presented a class of new extended shift-splitting (*NESS*) preconditioners for solving symmetric positive definite saddle point and they presented the *NESS* preconditioner as

$$\mathcal{P}_{NESS} = \begin{pmatrix} P + lA & lB \\ -lB^T & Q \end{pmatrix}.$$

There is no discussion on the *NESS*[59] and *PGSS*[54] iteration methods for the nonsymmetric saddle point problem with nonsymmetric positive definite A. In this article, inspired by the $\mathcal{P}_{NESS}$ and $\mathcal{P}_{PGSS}$, we propose a parameterized extended shift-splitting (*PESS*) preconditioner for nonsymmetric and nonsingular saddle point problems with nonsymmetric (1,1)-block. The present work also evaluates the convergence of the introduced iterative method and examines the spectral characteristics of the *PESS* preconditioned matrix. The numerical results in Section 5 show that the *PESS* method has a faster rate of convergence than *PGSS*,[18] *MGSS*, *GMSS*, *PIU*, and *DPSS* methods.

It is numerically demonstrated that *PESS*, and GMRES with *PESS* preconditioning are efficient methods. The remainder of the study is organized as follows: Section 2 describes the *PESS* preconditioner and its implementation. Section 3 discusses *PESS* convergence properties. Section 4 presents the spectral analysis of the *PESS* preconditioned matrix. Section 5 contains numerical results, and Section 6 presents some conclusions.

2 | DESCRIPTION AND IMPLEMENTATION OF THE PESS PRECONDITIONER

In this section, using the idea of References 18 and 59 for nonsymmetric saddle point problems, the following new splitting of $\mathfrak{A}$ is given as

$$\begin{aligned}\mathfrak{A} &= \mathcal{P}_{PESS} - \mathcal{Q}_{PESS}\\ &= \begin{pmatrix} \hat{\alpha}P + lA & lB \\ -lB^T & \hat{\beta}Q \end{pmatrix} - \begin{pmatrix} \hat{\alpha}P + (l-1)A & (l-1)B \\ -(l-1)B^T & \hat{\beta}Q \end{pmatrix},\end{aligned} \tag{2}$$

where $\hat{\alpha} \geq 0$, $\hat{\beta} > 0$, $l \in \mathbb{R}^+$, and $P \in \mathbb{R}^{m\times m}$, $Q \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices. Therefore, applying (2) emerges the following new method.

PESS iteration method: Suppose $l \in \mathbb{R}^+$ and $\hat{\alpha} \geq 0$, $\hat{\beta} > 0$. Assume $(\hat{x}^{(0)^T}, \hat{y}^{(0)^T})^T = \hat{X}^{(0)}$ be an initial guess. We compute

$$\mathcal{P}_{PESS}\hat{X}^{(k+1)} = \mathcal{Q}_{PESS}\hat{X}^{(k)} + b, \tag{3}$$

where $\hat{X}^{(k+1)} = \begin{pmatrix} \hat{x}^{(k+1)} \\ \hat{y}^{(k+1)} \end{pmatrix}$, until convergency of $\hat{X}^{(k)}$, $k = 0, 1, \ldots$. Iteration scheme (3) can be rewritten as follows

$$\hat{X}^{(k+1)} = \Gamma(\hat{\alpha}, \hat{\beta}, l)\hat{X}^{(k)} + C_1, \tag{4}$$

where

$$\Gamma(\hat{\alpha}, \hat{\beta}, l) = \begin{pmatrix} \hat{\alpha}P + lA & lB \\ -lB^T & \hat{\beta}Q \end{pmatrix}^{-1} \begin{pmatrix} lA + \hat{\alpha}P - A & lB - B \\ B^T - lB^T & \hat{\beta}Q \end{pmatrix}$$

is the iteration matrix of the PESS method and

$$C_1 = \begin{pmatrix} \hat{\alpha}P + lA & lB \\ -lB^T & \hat{\beta}Q \end{pmatrix}^{-1} \begin{pmatrix} f \\ -g \end{pmatrix}.$$

Note that each splitting matrix introduces a splitting preconditioner for the Krylov subspace methods and produces a splitting iteration method.

The preconditioner related to the splitting (2) can be presented by

$$\mathcal{P}_{PESS} = \begin{pmatrix} \hat{\alpha}P + lA & lB \\ -lB^T & \hat{\beta}Q \end{pmatrix} = \Omega + l\mathfrak{A}, \quad \text{with} \quad \Omega = \begin{pmatrix} \hat{\alpha}P & 0 \\ 0 & \hat{\beta}Q \end{pmatrix}, \tag{5}$$

where $\mathcal{P}_{PESS}$ is the *PESS* preconditioner for $\mathfrak{A}$.

With the different choice of parameters $\hat{\alpha}, \hat{\beta}$ and matrices P, Q, we can easily get a series of existing splitting preconditioners for the saddle point problem (1), such as *SS*, *GSS*, *PGSS*, and *ESS* preconditioners. The *PESS* method benefits from all advantages of these methods.

I. If $l = \frac{1}{2}, \hat{\alpha} = \hat{\beta}, P = Q = \frac{1}{2}I$, then the *PESS* preconditioner yields the *SS*[51] preconditioner.
II. If $l = \frac{1}{2}, P = Q = \frac{1}{2}I$, then the *PESS* preconditioner becomes the *GSS*[52,53] preconditioner.
III. If $l = 2, P = Q = I$, then the *PESS* preconditioner becomes the *MGSS*[17] preconditioner.
IV. If $P = Q = I$, then the *PESS* preconditioner gets the *PGSS*[18] preconditioner.
V. *PESS* preconditioner with $\hat{\alpha} = \hat{\beta} = l = \frac{1}{2}$ is the *ESS*[57] preconditioner.

At each step of (4) or applying $\mathcal{P}_{PESS}$ within a Krylov subspace methods, we need to solve a linear system of the following form

$$\begin{pmatrix} \hat{\alpha}P + lA & lB \\ -lB^T & \hat{\beta}Q \end{pmatrix} z = r, \tag{6}$$

where $z = (\hat{z_1}^T, \hat{z_2}^T)^T$, $r = (\hat{r_1}^T, \hat{r_2}^T)^T$ and $\hat{r_1}, \hat{z_1} \in R^m$, $\hat{r_2}, \hat{z_2} \in R^n$. It is clear that

$$\mathcal{P}_{PESS} = \begin{pmatrix} I & \frac{l}{\hat{\beta}}BQ^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T & 0 \\ 0 & \hat{\beta}Q \end{pmatrix} \begin{pmatrix} I & 0 \\ -\frac{l}{\hat{\beta}}Q^{-1}B^T & I \end{pmatrix}. \tag{7}$$

Substituting (7) into the coefficient matrix of (6), we will get $\hat{z_1}, \hat{z_2}$ from the following expression

$$\begin{pmatrix} \hat{z_1} \\ \hat{z_2} \end{pmatrix} = \begin{pmatrix} I & 0 \\ \frac{l}{\hat{\beta}}Q^{-1}B^T & I \end{pmatrix} \begin{pmatrix} \hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T & 0 \\ 0 & \hat{\beta}Q \end{pmatrix}^{-1} \begin{pmatrix} I & -\frac{l}{\hat{\beta}}BQ^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{r_1} \\ \hat{r_2} \end{pmatrix}. \tag{8}$$

In order to find $\hat{z_1}, \hat{z_2}$, we define the following algorithm.

In Algorithm 1, the solution process of (8) is presented and shows that in each iteration we need to solve $(\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T)\hat{z_1} = t_1$. For all $\hat{\alpha} \geq 0$, $\hat{\beta} > 0$ and $l > 0$, the matrix $\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T$ is positive definite. The systems could be solved approximately, for this GMRES can be employed for solving sub-linear systems accurately enough. Furthermore, LU factorization with AMD or column AMD reordering[51,60,61] can solve it exactly. Although inexact solvers can reduce iteration costs, they will also slow down convergence somewhat in experiments. In this article, we apply the LU factorization in combination with column AMD reordering to solve $(\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T)\hat{z_1} = t_1$.

3 | THE CONVERGENCE OF THE *PESS* ITERATION METHOD

In this section, we investigate the *PESS* iteration method's convergence properties. Fixed-point equation (4), for nonsingular saddle point problems, converges to $u = \mathfrak{A}^{-1}b$ for right-hand side b and arbitrary initial guess $(x(0)^T, y(0)^T)^T$ if and only if $\rho(\Gamma(\hat{\alpha}, \hat{\beta}, l)) < 1$.

Lemma 1 (2). *Assume $A \in \mathbb{R}^{m\times m}$ is nonsymmetric positive definite and $B \in \mathbb{R}^{m\times n}$ has full column rank. Then matrix $\mathfrak{A}$ is positive stable, that is, $Re(\lambda) > 0$ for all $\lambda \in \sigma(\mathfrak{A})$, where $\sigma(\mathfrak{A})$ is the spectral set of $\mathfrak{A}$.*

The following lemma is a generalization of Lemma 3.2 from Reference 15 to *PESS* method.

Lemma 2. *Assume $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{m\times n}$ are nonsymmetric positive definite and a full column rank matrix, respectively. Take $\hat{\alpha}$ and $\hat{\beta}$ positive. Then the real part of all eigenvalues of $\Omega^{-1}\mathfrak{A}$ is positive, where Ω is defined as in (5).*

Proof. The iteration matrix $\Gamma(\hat{\alpha}, \hat{\beta}, l)$ in (4) can be rewritten as

$$\Gamma(\hat{\alpha}, \hat{\beta}, l) = (\Omega + l\mathfrak{A})^{-1}(\Omega + (l-1)\mathfrak{A}) = (I + l\Omega^{-1}\mathfrak{A})^{-1}(I + (l-1)\Omega^{-1}\mathfrak{A}).$$

Algorithm 1. PESS iteration method

input $r = (\hat{r_1}^T, \hat{r_2}^T)^T$
compute $t_1 = \hat{r_1} - \frac{l}{\hat{\beta}}BQ^{-1}\hat{r_2}$;
solve $\hat{z_1}$ from $(\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T)\hat{z_1} = t_1$;
compute $\hat{z_2} = \frac{Q^{-1}}{\hat{\beta}}(lB^T\hat{z_1} + \hat{r_2})$.

Let λ and μ be an arbitrary eigenvalue of matrix $\Omega^{-1}\mathfrak{A}$ and iteration matrix $\Gamma(\hat{\alpha}, \hat{\beta}, l)$, respectively. Then it holds that $\mu = \frac{1+(l-1)\lambda}{1+l\lambda}$ and $\mathfrak{A}w = \lambda\Omega w$, where w is the eigenvector corresponding to eigenvalue λ. The eigenvalue problem $\mathfrak{A}w = \lambda\Omega w$ is equivalent to

$$\Omega^{-\frac{1}{2}}\mathfrak{A}\Omega^{-\frac{1}{2}}\overline{w} = \lambda\overline{w} \qquad \text{with} \quad \overline{w} = \Omega^{\frac{1}{2}}w,$$

where

$$\Omega^{-\frac{1}{2}}\mathfrak{A}\Omega^{-\frac{1}{2}} = \begin{pmatrix} \frac{1}{\hat{\alpha}}P^{-\frac{1}{2}}AP^{-\frac{1}{2}} & \frac{1}{\sqrt{\alpha\beta}}P^{-\frac{1}{2}}BQ^{-\frac{1}{2}} \\ -\frac{1}{\sqrt{\alpha\beta}}Q^{-\frac{1}{2}}B^TP^{-\frac{1}{2}} & 0 \end{pmatrix}. \tag{9}$$

Since A is nonsymmetric and positive definite and $P^{-\frac{1}{2}}$ is symmetric and positive definite, $P^{-\frac{1}{2}}AP^{-\frac{1}{2}}$ is nonsymmetric and positive definite. Since B is full column rank and $Q^{-\frac{1}{2}}$ is symmetric and positive definite, $P^{-\frac{1}{2}}BQ^{-\frac{1}{2}}$ is full column rank and the upper matrix (9) has the same block structure as the nonsymmetric saddle point matrix $\mathfrak{A}$. By Lemma 1 all eigenvalues of the matrix $\Omega^{-1}\mathfrak{A}$ have a positive real part. ■

The following theorem is a generalization of Theorem 3.1 from Reference 18 to *PESS* method. Since the proof is similar, it is omitted.

Theorem 1. *Assume that λ is an eigenvalue of $\Gamma(\hat{\alpha}, \hat{\beta}, l)$ and the conditions in Lemma 2 are satisfied. If $l > \max\left\{\frac{1}{2} - \frac{\lambda_{\min}(\overline{H})}{\rho(\overline{\mathfrak{A}})^2}, 0\right\}$, where $\overline{\mathfrak{A}} = \Omega^{-1}\mathfrak{A}$ and $\overline{H} = \frac{1}{2}(\overline{\mathfrak{A}} + \overline{\mathfrak{A}}^T)$, then the PESS iteration method converges to the exact solution of saddle point problem (1).*

Remark 1. In Theorem 1, if the size of $\mathfrak{A}$ is large, it is difficult to find $\lambda_{\min}(\overline{H})$ and $\rho(\overline{\mathfrak{A}})$. Thus, the condition $l > \max\left\{\frac{1}{2} - \frac{\lambda_{\min}(\overline{H})}{\rho(\overline{\mathfrak{A}})^2}, 0\right\}$ is impractical in many cases, so it can be replaced by $l \geq \frac{1}{2}$.

4 | THE SPECTRAL ANALYSIS OF THE *PESS* PRECONDITIONED MATRIX

The rate of convergence significantly correlates with the eigenvalue distribution of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$, that is why we investigate the spectral features of the $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$. Let $(\lambda, \gamma = (u^*, v^*)^*)$ be an eigenpair of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$, we have $\mathcal{P}_{PESS}^{-1}\mathfrak{A}\gamma = \lambda\gamma$, so

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \lambda\begin{pmatrix} \hat{\alpha}P + lA & lB \\ -lB^T & \hat{\beta}Q \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}.$$

This can also be written as:

$$\begin{cases} Au + Bv = \lambda(\hat{\alpha}Pu + lAu + lBv), \\ -B^Tu = -\lambda(lB^Tu - \hat{\beta}Qv), \end{cases}$$

or

$$\begin{cases} Au = \lambda(\hat{\alpha}H + lA)u + (\lambda l - 1)Bv, & (10) \\ (\lambda l - 1)B^Tu = \lambda\hat{\beta}v. & (11) \end{cases}$$

The following theorems are a generalization of Theorem 5.1 from References 17 and 18 to the *PESS* method.

Theorem 2. *Let the preconditioner of the PESS method be defined as in (5) and B has full column rank and $(\lambda, (u^*, v^*)^*)$ be an eigenpair of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$. If $B^Tu = 0$, then $\lambda = \frac{1}{l}$ as $\hat{\alpha} = 0$; if $\hat{\alpha} > 0$, then*

$$l_1 \leq Re(\lambda) \leq u_1 \quad \text{and} \quad |Im(\lambda)| \leq u_2, \tag{12}$$

where

$$l_1 = \frac{\lambda_{\min}(H)(\hat{\alpha}\lambda_{\min}(P) + l\lambda_{\min}(H))}{(\hat{\alpha}\rho(P) + l\rho(H))^2 + l^2\rho(S)^2},$$
$$u_1 = \frac{\rho(H)(\hat{\alpha}\rho(P) + l\rho(H)) + l\rho(S)^2}{(\hat{\alpha}\lambda_{\min}(P) + l\lambda_{\min}(H))^2},$$
$$u_2 = \frac{\hat{\alpha}\rho(P)\rho(S)}{(\hat{\alpha}\lambda_{\min}(P) + l\lambda_{\min}(H))^2},$$

with $S = \frac{1}{2}(A - A^T)$, $\lambda = Re(\lambda) + iIm(\lambda)$.

Proof. Since $B^T u = 0$, equality $\lambda\hat{\beta}Qv = 0$ from (11) gets $v = 0$. If $u = 0$, we have $(u^*, v^*)^* = 0$, which contradicts the definition of an eigenvector. Substituting $v = 0$ into (10) yields

$$Au = \lambda(\hat{\alpha}P + lA)u. \tag{13}$$

The expression $\frac{u^*}{u^*u}$ is well defined due to $u \neq 0$. Multiplying the left-hand side of (13) by $\frac{u^*}{u^*u}$ and using

$$\frac{u^*Pu}{u^*u} = d_1, \quad \frac{u^*Au}{u^*u} = a_1 + ib_1, \tag{14}$$

gets

$$\lambda = \frac{a_1 + ib_1}{\hat{\alpha}d_1 + l(a_1 + b_1 i)} = \frac{\hat{\alpha}a_1 d_1 + la_1^2 + lb_1^2 + \hat{\alpha}d_1 b_1 i}{(\hat{\alpha}d_1 + la_1)^2 + l^2 b_1^2}, \tag{15}$$

if $\hat{\alpha} = 0$, then $\lambda = \frac{1}{l}$. If $\hat{\alpha} > 0$, then from (15) we get

$$Re(\lambda) = \frac{\hat{\alpha}a_1 d_1 + la_1^2 + lb_1^2}{(\hat{\alpha}d_1 + la_1)^2 + l^2 b_1^2}, \quad Im(\lambda) = \frac{\hat{\alpha}d_1 b_1}{(\hat{\alpha}d_1 + la_1)^2 + l^2 b_1^2}.$$

Since

$$0 \leq |b_1| = \left| \frac{1}{2i} \left(\frac{u^*Au}{u^*u} - \frac{u^*A^Tu}{u^*u} \right) \right| = \left| \frac{u^*iSu}{u^*u} \right| \leq \rho(S),$$
$$\lambda_{\min}(H) \leq a_1 = \frac{1}{2} \left(\frac{u^*Au}{u^*u} + \frac{u^*A^Tu}{u^*u} \right) = \frac{u^*Hu}{u^*u} \leq \rho(H),$$
$$\lambda_{\min}(P) \leq d_1 \leq \rho(P),$$

it is easy to prove (12). It is also clear from (12) that $Re(\lambda) > 0$. ■

Lemma 3. *Let* $\hat{z}_1, \hat{z}_2$ *are real numbers and* $\hat{z}_1 + i\hat{z}_2$ *is one of the square roots of* $a_2 + ib_2$*, where*

$$a_2 = \hat{\beta}^2 \left(a_1^2 - b_1^2\right) - 4\hat{\alpha}\hat{\beta}c_1 d_1, \quad b_2 = 2\hat{\beta}^2 a_1 b_1,$$

then

$$\hat{z}_1 = \sqrt{\frac{\sqrt{d^2 + 4\hat{\beta}^4 a_1^2 b_1^2} + d}{2}},$$
$$\hat{z}_2 = \text{sign}(b_1)\sqrt{\frac{\sqrt{d^2 + 4\hat{\beta}^4 a_1^2 b_1^2} - d}{2}}, \tag{16}$$

where

$$d = \hat{\beta}^2\left(a_1^2 - b_1^2\right) - 4\hat{\alpha}\hat{\beta}c_1d_1$$

and the second root of $a_2 + ib_2$ *is* $-(\hat{z}_1 + i\hat{z}_2)$.

Proof. Let $z = z_1 + iz_2$, solving equation $z^2 = a_2 + ib_2$ gets the proof of lemma. ∎

Theorem 3. *Assume that the conditions in Theorem 2 are satisfied. If* $B^T u \neq 0$, *then the preconditioned matrix* $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ *has eigenvalue* $\lambda = \frac{1}{l}$ *and the remaining eigenvalues are enclosed in the rectangle*:

$$\left[\frac{\hat{\beta}\lambda_{\min}(H)\lambda_{\min}(BQ^{-1}B^T) + l\lambda_{\min}(BQ^{-1}B^T)^2}{(\hat{\beta}\rho(H) + l\rho(BQ^{-1}B^T))^2 + \hat{\beta}^2\rho(S)^2}, \frac{\hat{\beta}\rho(H)\rho(BQ^{-1}B^T) + l\rho(BQ^{-1}B^T)\,)^2}{(\hat{\beta}\lambda_{\min}(H) + l\lambda_{\min}(BQ^{-1}B^T))^2}\right] \times \left[-\frac{\hat{\beta}\rho(S)\rho(BQ^{-1}B^T)}{(\hat{\beta}\lambda_{\min}(H) + l\lambda_{\min}(BQ^{-1}B^T))^2}, \frac{\hat{\beta}\rho(S)\rho(BQ^{-1}B^T)}{(\hat{\beta}\lambda_{\min}(H) + l\lambda_{\min}(BQ^{-}B^T))^2}\right], \tag{17}$$

as $\hat{\alpha} = 0$. *Furthermore, the eigenvalues of* $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ *converge to* $(\frac{1}{l}, 0)$ *as* $\hat{\beta} \to 0_+$. *If* $\hat{\alpha} > 0$, *then the eigenvalues of the preconditioned matrix* $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ *satisfy*

$$\lambda_+ = \frac{1}{l} + \frac{\left(\hat{z}_1 - \hat{\beta}a_1 - \frac{2\hat{\alpha}\hat{\beta}a_1}{l}\right) + i(\hat{z}_2 - \hat{\beta}b_1)}{2(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2c_1 + i\hat{\beta}lb_1)},$$
$$\lambda_- = \frac{1}{l} - \frac{\left(\hat{z}_1 + \hat{\beta}a_1 + \frac{2\hat{\alpha}\hat{\beta}a_1}{l}\right) + i(\hat{z}_2 + \hat{\beta}b_1)}{2(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2c_1 + i\hat{\beta}lb_1)}, \tag{18}$$

where

$$\frac{u^*BQ^{-1}B^Tu}{u^*u} = c_1, \quad \frac{u^*Pu}{u^*u} = d_1, \quad \frac{u^*Au}{u^*u} = a_1 + ib_1, \tag{19}$$

and z_1, z_2 *are defined in (16). The eigenvalues* $\lambda_\pm$ *satisfy the inequality*:

$$\left|\lambda_\pm - \frac{1}{l}\right|^2 \leq \frac{\left(2\hat{\beta}\rho(H) + \frac{2\hat{\alpha}\hat{\beta}\rho(P)}{l}\right)^2 + (\hat{\beta}\rho(S) + \sqrt{\hat{\beta}^2\rho(S)^2 + 4\hat{\alpha}\hat{\beta}\rho(BQ^{-1}B^T)\rho(P)})^2}{4(\hat{\alpha}\hat{\beta}\lambda_{\min}(P) + l\hat{\beta}\lambda_{\min}(H) + l^2\lambda_{\min}(BQ^{-1}B^T))^2}. \tag{20}$$

When $\beta \to 0_+$, *it holds that*

$$\lambda_\pm \to \frac{1}{l}, \tag{21}$$

that is, the eigenvalues of the $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ *converge to* $\left(\frac{1}{l}, 0\right)$.

Proof. Since $B^Tu \neq 0$, we can write (11) as $v = \frac{(l\lambda-1)Q^{-1}B^Tu}{\lambda\hat{\beta}}$, where $\lambda \neq 0$ and $u \neq 0$. Then (10) becomes

$$\lambda^2(\hat{\alpha}\hat{\beta}P + l\hat{\beta}A + l^2BQ^{-1}B^T)u - \lambda(\hat{\beta}A + 2lBQ^{-1}B^T)u + BQ^{-1}B^Tu = 0. \tag{22}$$

By multiplying (22) by $\frac{u^*}{u^*u}$ and using (19), we obtain

$$\lambda^2 - \lambda\frac{\hat{\beta}a_1 + 2lc_1 + i\hat{\beta}b_1}{\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + i\hat{\beta}lb_1 + l^2c_1} + \frac{c_1}{\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + i\hat{\beta}lb_1 + l^2c_1} = 0. \tag{23}$$

The roots of (23) are

$$\lambda_{+} = \frac{1}{l} + \frac{\left(\hat{z}_1 - \hat{\beta}a_1 - \frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right) + i(\hat{z}_2 - \hat{\beta}b_1)}{2(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2c_1 + i\hat{\beta}lb_1)},$$
$$\lambda_{-} = \frac{1}{l} - \frac{\left(\hat{z}_1 + \hat{\beta}a_1 + \frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right) + i(\hat{z}_2 + \hat{\beta}b_1)}{2(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2c_1 + i\hat{\beta}lb_1)}. \tag{24}$$

By some calculations, we get the following inequalities

$$Re(\lambda_{+}) = \frac{K + (\hat{\alpha}\hat{\beta}d_1 + l^2c_1)(2lc_1 + \hat{z}_1) + l\hat{\beta}(a_1\hat{z}_1 + |b_1\hat{z}_2|)}{2[(\hat{\alpha}\hat{\beta}a_1 + l\hat{\beta}a_1 + l^2c_1)^2 + l^2\hat{\beta}^2b_1^2]} > 0,$$
$$Re(\lambda_{-}) = \frac{K + (\hat{\alpha}\hat{\beta}d_1 + l^2c_1)(2lc_1 - \hat{z}_1) - l\hat{\beta}(a_1\hat{z}_1 + |b_1\hat{z}_2|)}{2[(\hat{\alpha}\hat{\beta}a_1 + l\hat{\beta}a_1 + l^2c_1)^2 + l^2\hat{\beta}^2b_1^2]}$$
$$\geq \frac{l^2c_1(\hat{\beta}a_1 + lc_1)}{(\hat{\alpha}\hat{\beta}a_1 + l\hat{\beta}a_1 + l^2c_1)^2 + l^2\hat{\beta}^2b_1^2} > 0,$$

where $K = \hat{\alpha}\hat{\beta}^2a_1d_1 + l\hat{\beta}^2(a_1^2 + b_1^2) + 3l^2\hat{\beta}a_1c_1$ and $\hat{z}_1, \hat{z}_2$ are presented by (16). From (16)

$$\hat{z}_1 = \sqrt{\frac{\sqrt{d^2 + 4\hat{\beta}^4a_1^2b_1^2} + d}{2}}$$
$$= \sqrt{\frac{\sqrt{\hat{\beta}^4(a_1^2 - b_1^2)^2 - 8\hat{\alpha}\hat{\beta}^3c_1d_1(a_1^2 - b_1^2) + 4\hat{\beta}^4a_1^2b_1^2 + 16\hat{\alpha}^2\hat{\beta}^2d_1^2d_1^2} + d}{2}}$$
$$\leq \sqrt{\frac{\sqrt{[\hat{\beta}^2(a_1^2 + b_1^2) + 4\hat{\alpha}\hat{\beta}c_1d_1]^2} + d}{2}} = \hat{\beta}a_1, \tag{25}$$
$$|\hat{z}_2| = \sqrt{\frac{\sqrt{d^2 + 4\hat{\beta}^4a_1^2b_1^2} - d}{2}} \leq \sqrt{\frac{\sqrt{\left[\hat{\beta}^2(a_1^2 + b_1^2) + 4\hat{\alpha}\hat{\beta}c_1d_1\right]^2} - d}{2}}$$
$$= \sqrt{\hat{\beta}^2b_1^2 + 4\hat{\alpha}\hat{\beta}c_1d_1}. \tag{26}$$

Thus, by using (25) and (26), we derive the following inequalities from (24)

$$\left|\lambda_{\pm} - \frac{1}{l}\right|^2 = \frac{\left(\hat{\beta}a_1 + \frac{2\hat{\alpha}\hat{\beta}d_1}{l} \pm \hat{z}_1\right)^2 + (\hat{\beta}b_1 \pm \hat{z}_2)^2}{4[(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2c_1)^2 + l^2\hat{\beta}^2b_1^2]}$$
$$\leq \frac{\left(2\hat{\beta}a_1 + \frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right)^2 + \left(\hat{\beta}|b_1| + \sqrt{\hat{\beta}^2b_1^2 + 4\hat{\alpha}\hat{\beta}c_1d_1}\right)^2}{4\left[(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2c_1)^2 + l^2\hat{\beta}^2b_1^2\right]} = f(a_1, b_1, c_1, d_1). \tag{27}$$

It is evident that an upper bound of $|\lambda_{\pm} - \frac{1}{l}|^2$ is $\max_{a_1,b_1,c_1,d_1} f(a_1, b_1, c_1, d_1)$ with a_1, b_1, c_1, d_1 being bounded as follows:

$$\lambda_{\min}(H) \leq a_1 \leq \rho(H), \quad 0 \leq |b_1| \leq \rho(S),$$
$$\lambda_{\min}(BQ^{-1}B^T) \leq c_1 \leq \rho(BQ^{-1}B^T), \quad 0 \leq b_1^2 \leq \rho(S)^2,$$
$$\lambda_{\min}(P) \leq d_1 \leq \rho(P) \tag{28}$$

thus

$$\left|\lambda_{\pm}-\frac{1}{l}\right|^2 \leq f(a_1,b_1,c_1,d_1)$$
$$\leq \frac{\left(2\hat{\beta}\rho(H)+\frac{2\hat{\alpha}\hat{\beta}\rho(P)}{l}\right)^2+(\hat{\beta}\rho(S)+\sqrt{\hat{\beta}^2\rho(S)^2+4\hat{\alpha}\hat{\beta}\rho(BQ^{-1}B^T)\rho(P)})^2}{4(\hat{\alpha}\hat{\beta}\lambda_{\min}(P)+l\hat{\beta}\lambda_{\min}(H)+l^2\lambda_{\min}(BQ^{-1}B^T))^2},$$

then it yields (20). On the other hand, when $\hat{\beta}\to 0_+$, we have $\hat{z_1},\hat{z_2}\to 0$. Therefore, for $\hat{\alpha}\geq 0$, we have $\lambda_+,\lambda_-\to(\frac{1}{l},0)$ as $\hat{\beta}\to 0_+$. Moreover, when $\hat{\alpha}=0$, we have $\hat{z_1}=\hat{\beta}a_1$ and $\hat{z_2}=\hat{\beta}b_1$ in (16). Accordingly,

$$\begin{cases}\lambda_+=\dfrac{1}{l}+\dfrac{(\hat{z_1}-\hat{\beta}a_1)+i(\hat{z_2}-\hat{\beta}b_1)}{2(\hat{\beta}la_1+l^2c_1+i\hat{\beta}lb_1)}\to\dfrac{1}{l},\\ \lambda_-=\dfrac{1}{l}-\dfrac{(\hat{z_1}+\hat{\beta}a_1)+i(\hat{z_2}+\hat{\beta}b_1)}{2(\hat{\beta}la_1+l^2c_1+i\hat{\beta}lb_1)}=\dfrac{1}{l}-\dfrac{\hat{\beta}a_1+i\hat{\beta}b_1}{\hat{\beta}la_1+i\hat{\beta}lb_1+l^2c_1}=\dfrac{\hat{\beta}a_1c_1+lc_1^2-i\hat{\beta}b_1c_1}{(\hat{\beta}a_1+lc_1)^2+\hat{\beta}^2b_1^2},\end{cases}$$

implies that for $\hat{\beta}>0$, the eigenvalues of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ are $\frac{1}{l}$ and $\frac{\hat{\beta}a_1c_1+lc_1^2-i\hat{\beta}b_1c_1}{(\hat{\beta}a_1+lc_1)^2+\hat{\beta}^2b_1^2}$ as $\alpha=0$. We obtain

$$\frac{\hat{\beta}\lambda_{\min}(H)\lambda_{\min}(BQ^{-1}B^T)+l\lambda_{\min}(BQ^{-1}B^T)^2}{(\hat{\beta}\rho(H)+l\rho(BQ^{-1}B^T))^2+\hat{\beta}^2\rho(S)^2}\leq\frac{\hat{\beta}a_1c_1+lc_1^2}{(\hat{\beta}a_1+lc_1)^2+\hat{\beta}^2b_1^2}$$
$$\leq\frac{\hat{\beta}\rho(H)\rho(BQ^{-1}B^T)+l\rho(BQ^{-1}B^T)^2}{(\hat{\beta}\lambda_{\min}(H)+l\lambda_{\min}(BQ^{-1}B^T))^2},$$

and

$$\left|\frac{\hat{\beta}b_1c_1}{(\hat{\beta}a_1+lc_1)^2+\hat{\beta}^2b_12}\right|\leq\frac{\hat{\beta}\rho(S)\rho(BQ^{-1}B^T)}{(\hat{\beta}\lambda_{\min}(H)+l\lambda_{\min}(BQ^{-1}B^T))^2},$$

using inequality (28). Therefore, $\frac{1}{l}$ is one of the eigenvalues of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ as $\hat{\beta}>0,\hat{\alpha}=0$, and the other eigenvalues are bounded by rectangle (17). ∎

Remark 2. Using Theorem 2, it can be seen that $\lambda=\frac{1}{l}>0$ when $B^Tu=0$ and $\hat{\alpha}=0$, and from (12), we have $Re(\lambda)>0$ as $\hat{\alpha}>0$, where $(\lambda,(u^*,v^*)^*)$ is an eigenpair of the preconditioned matrix $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$. Thus all eigenvalues of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ have a positive real part and lie within a positive box. Moreover, when $B^Tu=0$ and $\hat{\alpha}=0$, we have $\lambda=\frac{1}{l}$ or $\lambda=0$; and Theorem 3 gets $\lambda\to\left(\frac{1}{l},0\right)$ when $B^Tu\neq 0$, $\hat{\beta}\to 0_+$ and $\hat{\alpha}\geq 0$. Therefore, these results imply that the *PESS* preconditioned matrix that is, $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ with suitable $\hat{\alpha}$ and $\hat{\beta}$ parameters will have a more clustered spectrum compared with the spectrum of matrix $\mathfrak{A}$. As a result, GMRES with $\mathcal{P}_{PESS}$ preconditioner leads to rapid convergence rate of that. Tables in Section 5 confirm this conclusion. Also, when $c_1>0$, we have from (25) and (26) that

$$\left(2\hat{\beta}a_1+\frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right)^2+\left(\hat{\beta}|b_1|+\sqrt{\hat{\beta}^2b_1^2+4\hat{\alpha}\hat{\beta}c_1d_1}\right)^2$$
$$=\left(2\hat{\beta}a_1+\frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right)^2+\left(2\hat{\beta}^2b_1^2+4\hat{\alpha}\hat{\beta}c_1d_1+2\hat{\beta}|b_1|\sqrt{\hat{\beta}^2b_1^2+4\hat{\alpha}\hat{\beta}c_1d_1}\right)$$
$$\leq\left(2\hat{\beta}a_1+\frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right)^2+2\hat{\beta}^2b_1^2+4\hat{\alpha}\hat{\beta}c_1d_1+2\hat{\beta}|b_1|\sqrt{\hat{\beta}^2b_1^2+4\hat{\alpha}\hat{\beta}c_1d_1+\left(\frac{2\hat{\alpha}c_1d_1}{|b_1|}\right)^2}$$
$$=\left(2\hat{\beta}a_1+\frac{2\hat{\alpha}\hat{\beta}d_1}{l}\right)^2+4\hat{\beta}^2b_1^2+4\hat{\alpha}\hat{\beta}c_1d_1$$
$$<\left(\frac{2\hat{\alpha}\hat{\beta}d_1}{l}+2\hat{\beta}a_1+2lc_1\right)^2+4\hat{\beta}^2b_1^2,$$

TABLE 1 Numerical results for $\mu = 0.1$.

Method	p	16	32	48	64	128
GMSS	$\hat{\alpha}_{exp}$	22	36	39	38	42
	$\hat{\beta}_{exp}$	16	8.3	6.8	5.9	4.7
	IT.	66	73	79	89	-
	CPU	0.572	10.446	9.837	211.427	-
MGSS	$\hat{\alpha}_{exp}$	0.2	0.5	0.2	0.2	0.2
	$\hat{\beta}_{exp}$	0.1	0.1	0.1	0.1	0.1
	IT.	21	21	21	21	21
	CPU	0.175	2.985	16.093	50.187	791.919
PGSS	$\hat{\alpha}$	0.2	0.2	0.2	0.2	0.2
	$\hat{\beta}$	0.2	0.2	0.2	0.2	0.2
	IT.	8	9	10	10	12
	CPU	0.078	1.110	6.959	21.616	445.930
PESS	$\hat{\alpha}$	0.1	0.1	0.1	0.1	0.1
	$\hat{\beta}$	0.1	0.1	0.1	0.1	0.1
	IT.	4	4	4	4	4
	CPU	0.048	0.408	2.338	6.986	124.033

which together with inequality (27) we get

$$\begin{aligned}|\lambda_{\pm} - \frac{1}{l}|^2 &\leq \frac{(2\hat{\beta}a_1 + \frac{2\hat{\alpha}\hat{\beta}d_1}{l})^2 + \left(\hat{\beta}|b_1| + \sqrt{\hat{\beta}^2 b_1^2 + 4\hat{\alpha}\hat{\beta}c_1 d_1}\right)^2}{4[(\hat{\alpha}\hat{\beta}d_1 + l\hat{\beta}a_1 + l^2 c_1)^2 + l^2\hat{\beta}^2 b_1^2]} \\ &= \frac{(2\hat{\beta}a_1 + \frac{2\hat{\alpha}\hat{\beta}d_1}{l})^2 + (\hat{\beta}|b_1| + \sqrt{\hat{\beta}^2 b_1^2 + 4\hat{\alpha}\hat{\beta}c_1 d_1})^2}{l^2\left[\left(\frac{2\hat{\alpha}\hat{\beta}d_1}{l} + 2\hat{\beta}a_1 + 2lc_1\right)^2 + 4\hat{\beta}^2 b_1^2\right]} \\ &< \frac{\left(\frac{2\hat{\alpha}\hat{\beta}d_1}{l} + 2\hat{\beta}a_1 + 2lc_1\right)^2 + 4\hat{\beta}^2 b_1^2}{l^2\left[\left(\frac{2\hat{\alpha}\hat{\beta}d_1}{l} + 2\hat{\beta}a_1 + 2lc_1\right)^2 + 4\hat{\beta}^2 b_1^2\right]} = \frac{1}{l^2},\end{aligned} \tag{29}$$

which implies that $\left|\lambda_{\pm} - \frac{1}{l}\right| < \frac{1}{l}$ as $B^T u \neq 0$. When $B^T u = 0$ then $\lambda = 0$ or λ satisfies (15), from (15) we have

$$\left|\lambda_{\pm} - \frac{1}{l}\right|^2 = \frac{\hat{\alpha}^2}{l^2(\hat{\alpha}d_1 + la_1)^2 + l^2 b_1^2} < \frac{1}{l^2}.$$

According to the above discussion, all eigenvalues of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ lie in a circle centered at $(\frac{1}{l}, 0)$ with radius $\frac{1}{l}$.

We know that the eigenvectors of the preconditioned matrix play an important rule for the convergence of the Krylov subspace methods except when the preconditioned matrix is symmetric.[33] Therefore, we present the following theorem which explains the distribution of eigenvectors of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$.

Theorem 4. *Let PESS preconditioner $\mathcal{P}_{PESS}$ be as defined in (5) and $\hat{\alpha} = 0$, then $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ have $m + p$ $(0 \leq p \leq m)$ linearly independent eigenvectors, with m eigenvectors of the form $\begin{pmatrix} u_t \\ 0 \end{pmatrix}$ $(1 \leq t \leq m)$ for the eigenvalue $\frac{1}{l}$, where $u_t \neq 0$ $(1 \leq t \leq m)$*

TABLE 2 Numerical results of the GMRES method for $\mu = \hat{\alpha} = 0.1$ and $p = 32$ with six preconditioners

Preconditioner	p	16	32	48	64	128	256
I	IT.	131	287	447	-	-	-
	CPU	0.039	1.186	4.349	-	-	-
$\mathcal{P}_{GMSS}$	$\hat{\beta}$	4.9974	4.9996	4.9999	5	5	5
	IT.	24	26	26	26	28	32
	CPU	0.109	0.602	1.795	4.192	19.246	31.748
$\mathcal{P}_{MGSS}$	$\hat{\beta}$	19.9861	19.9983	19.9995	19.9998	20	20
	IT.	16	18	18	19	21	24
	CPU	0.077	0.430	1.202	2.922	12.588	23.696
$\mathcal{P}_{PIU}$	$\hat{\beta}$	9.9931	9.9992	9.9998	9.9999	10	10
	IT.	46	51	53	56	60	71
	CPU	0.189	1.234	3.846	6.982	21.775	35.912
$\mathcal{P}_{DPSS}$	IT.	37	51	63	72	99	185
	CPU	0.667	4.865	19.778	37.743	319.812	645.052
$\mathcal{P}_{PGSS}(l = 6)$	$\hat{\beta}$	59.9583	59.9950	59.9986	59.9994	59.9999	60
	IT.	14	15	15	15	16	19
	CPU	0.036	0.232	0.629	1.280	6.36	18.198
$\mathcal{P}_{PESS}(l = 6)$	$\hat{\beta}$	59.9583	59.9950	59.9986	59.9994	59.9999	60
	IT.	7	8	8	8	9	10
	CPU	0.020	0.120	0.294	0.650	3.545	10.780
$\mathcal{P}_{PGSS}(l = 3)$	$\hat{\beta}$	29.9792	29.9975	29.9993	29.9997	29.9999	30
	IT.	14	15	15	15	16	19
	CPU	0.035	0.197	0.475	1.150	5.087	17.956
$\mathcal{P}_{PESS}(l = 3)$	$\hat{\beta}$	29.9792	29.9975	29.9993	29.9997	29.9999	30
	IT.	7	8	8	8	9	9
	CPU	0.026	0.133	0.236	0.724	3.091	10.076
$\mathcal{P}_{PGSS}(l = 8)$	$\hat{\beta}$	79.9445	79.9934	79.9981	79.9992	79.9999	80
	IT.	14	15	15	15	16	18
	CPU	0.039	0.171	0.496	0.998	4.972	17.742
$\mathcal{P}_{PESS}(l = 8)$	$\hat{\beta}$	79.9445	79.9934	79.9981	79.9992	79.9999	80
	IT.	7	8	8	8	8	10
	CPU	0.023	0.144	0.308	0.646	2.903	9.980

are arbitrary linearly independent vectors, and p eigenvectors of the form $\begin{pmatrix} u_t^1 \\ \frac{(l\lambda-1)Q^{-1}B^T u_t^1}{\lambda\hat{\beta}} \end{pmatrix}$ $(1 \le t \le p)$, *which correspond to the eigenvalue* $\lambda \neq \frac{1}{l}$, *where* $u_t^1 \neq 0$ $(1 \le t \le p)$ *satisfy* $\lambda\hat{\beta}Au_t^1 = l\hat{\beta}\lambda^2 Au_t^1 + (l\hat{\lambda} - 1)^2 BQ^{-1}B^T u_t^1$ *and if* $\hat{\alpha} > 0$, *then* $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ *has q* $(0 \le q \le m)$ *linearly independent eigenvectors of the form* $\begin{pmatrix} u_t^2 \\ \frac{(l\lambda-1)Q^{-1}B^T u_t^2}{\lambda\hat{\beta}} \end{pmatrix}$ $(1 \le t \le q)$ *that correspond to the eigenvalue* $\lambda \neq \frac{1}{l}$, *when* $u_t^2 \neq 0$ $(1 \le t \le q)$ *satisfy* $\lambda\hat{\beta}Au_t^1 = \hat{\beta}\lambda^2(\hat{\alpha}P + lA)u_t^2 + (l\hat{\lambda} - 1)^2 BQ^{-1}B^T u_t^2$.

Proof. Suppose λ is an eigenvalue of $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ and $(u^*, v^*)^*$ is the corresponding eigenvector. If $u = 0$, then we get $\lambda Qv = 0$ from (11), and therefore, $v = 0$, which is a contradiction, so $u \neq 0$. Now let $\lambda = \frac{1}{l}$, we have $\hat{\alpha}Pu = 0$ from (10) and $v = 0$

TABLE 3 Numerical results of the GMRES method with six preconditioners for $\mu = \hat{\alpha} = 1$ and $p = 32$.

Preconditioner	p	16	32	48	64	128	256
I	IT.	206	436	-	-	-	-
	CPU	0.085	2.67	-	-	-	-
$\mathcal{P}_{GMSS}$	$\hat{\beta}$	0.4997	0.5	0.5	0.5	0.5	0.5001
	IT.	12	14	14	14	16	18
	CPU	0.058	0.350	0.947	2.170	9.477	17.968
$\mathcal{P}_{MGSS}$	$\hat{\beta}$	1.9989	1.9999	2	2	2	2.0003
	IT.	11	12	13	13	15	17
	CPU	0.058	0.311	0.882	2.060	9.116	17.137
$\mathcal{P}_{PIU}$	$\hat{\beta}$	0.9995	0.9999	1	1	1	1.0002
	IT.	34	36	38	39	44	53
	CPU	0.147	0.843	2.455	4.393	15.846	24.408
$\mathcal{P}_{DPSS}$	IT.	45	65	81	94	132	185
	CPU	0.778	6.501	24.909	54.029	435.909	656.270
$\mathcal{P}_{PGSS}(l = 5)$	$\hat{\beta}$	4.9974	4.9996	4.9999	5	5	5
	IT.	9	11	1	12	14	15
	CPU	0.077	0.158	0.375	0.976	5.975	14.454
$\mathcal{P}_{PESS}(l = 5)$	$\hat{\beta}$	4.9974	4.9996	4.9999	5	5	5
	IT.	5	6	7	7	7	7
	CPU	0.068	0.107	0.269	0.552	3.305	8.121
$\mathcal{P}_{PGSS}(l = 7)$	$\hat{\beta}$	6.9963	6.9994	6.9998	7	7	7
	IT.	9	10	11	11	14	15
	CPU	0.048	0.145	0.387	0.768	5.497	15.423
$\mathcal{P}_{PESS}(l = 7)$	$\hat{\beta}$	6.9963	6.9994	6.9998	7	7	7
	IT.	6	6	6	6	7	8
	CPU	0.046	0.084	0.238	0.214	3.298	8.976

from (11). If $\hat{\alpha} = 0$, (10) always holds for the case $\lambda = \frac{1}{l}$. Thus, there are m eigenvectors of the form $\begin{pmatrix} u_t \\ 0 \end{pmatrix}$ $(t = 1, 2, \ldots, m)$ which are linearly independent related to eigenvalue $\frac{1}{l}$, in which u_t are arbitrary linearly independent vectors. If $\hat{\alpha} > 0$ holds, then $v = 0$ and $u = 0$ is a contradiction. Now we check the case $\lambda \neq \frac{1}{l}$. If $\hat{\alpha} = 0$, then it follows from (11) that $v = \frac{(\lambda l - 1)B^T u}{\lambda \hat{\beta} Q}$. Substituting this relation into (10), we have

$$\lambda \hat{\beta} A u = l \hat{\beta} \lambda^2 A u + (l\lambda - 1)^2 B Q^{-1} B^T u. \tag{30}$$

Suppose there exists a $u \neq 0$ that satisfies (30), thus, we will have p $(1 \leq p \leq m)$ linearly independent eigenvectors of the form $\begin{pmatrix} u_t^1 \\ v_t^1 \end{pmatrix}$ $(1 \leq t \leq p)$ corresponding to the eigenvalue $\lambda \neq \frac{1}{l}$. Here, $u_t^1 (1 \leq t \leq p)$ satisfies $\lambda \hat{\beta} A u_t^1 = l \hat{\beta} \lambda^2 A u_t^1 + (l\lambda - 1)^2 B Q^{-1} B^T u_t^1$ and $v_t^1 (1 \leq t \leq p)$ is giving by

$$v_t^1 = \frac{(l\lambda - 1) Q^{-1} B^T u_t^1}{\lambda \hat{\beta}}. \tag{31}$$

If $\hat{\alpha} > 0$, then similar to (30), we can write

$$\lambda \hat{\beta} A u = \hat{\beta} \lambda^2 (\hat{\alpha} P + l A) u + (l\lambda - 1)^2 B Q^{-1} B^T u. \tag{32}$$

TABLE 4 Numerical results of the GMRES method for $\mu = 0.1$ and $p = 128$ with five preconditioners.

Preconditioner $(\hat{\alpha}, \hat{\beta})$		$\mathcal{P}_{PIU}$	$\mathcal{P}_{GMSS}$	$\mathcal{P}_{MGSS}$	$\mathcal{P}_{PGSS(l=6)}$	$\mathcal{P}_{PESS(l=6)}$
(0.6, 0.8)	IT.	89	20	7	6	3
	CPU	9.011	4.091	1.669	1.303	0.974
(0.2, 0.5)	IT.	86	18	6	5	3
	CPU	8.580	3.688	1.500	1.075	0.950
(0.2, 1.3)	IT.	85	22	8	6	4
	CPU	8.377	4.428	1.904	1.342	0.968
(1, 0.8)	IT.	93	20	7	6	3
	CPU	9.118	4.154	1.603	1.306	0.947
(1.2, 1.5)	IT.	87	23	9	7	4
	CPU	8.408	4.538	2.018	1.428	1.187
(1.5, 1.2)	IT.	100	22	8	7	4
	CPU	9.913	4.357	2.031	1.501	0.934
(1.8, 1.5)	IT.	87	23	9	7	4
	CPU	8.785	4.662	2.058	1.442	1.202
(0.02, 0.5)	IT.	55	18	6	5	3
	CPU	5.430	3.714	1.268	1.091	0.937
(0.89, 1.37)	IT.	98	22	8	7	4
	CPU	9.526	4.527	1.785	1.475	0.927
(0.93, 0.16)	IT.	97	16	5	5	3
	CPU	9.794	3.3	1.93	1.111	0.983
(0.01, 0.2)	IT.	45	14	7	4	3
	CPU	4.562	3.622	1.099	0.971	0.884

If there is a $u \neq 0$ that satisfies (32), there will be $q(1 \leq q \leq m)$ linearly independent eigenvectors of the form $\begin{pmatrix} u_t^2 \\ v_t^2 \end{pmatrix}$ $(1 \leq t \leq q)$ that corresponding to the eigenvalues $\lambda \neq \frac{1}{l}$. Here, $u_t^2 \neq 0$ $(1 \leq t \leq q)$ satisfies $\lambda \hat{\beta} A u_t^2 = \hat{\beta} \lambda^2 (\hat{\alpha} P + lA) u_t^2 + (l\hat{\lambda} - 1)^2 B Q^{-1} B^T u_t^2$ and v_t^2 $(1 \leq t \leq q)$ satisfies (31).

The linear independence of $m + p$ eigenvectors of the $\mathcal{P}_{PESS}^{-1} \mathfrak{A}$ as $\alpha = 0$ can be proved similar to that of Theorem 5.2 of Reference 17 and Theorem 3.2 of Reference 42. ■

5 | NUMERICAL RESULTS

We provide two examples to show the effectiveness of the *PESS* method to solve (1). Numerical results of *PESS*, *PGSS*,[18] *MGSS*,[17] *PIU*,[22] *DPSS*,[41] and *GMSS*[16] methods are reported based on the number of iterations (denoted by "IT") and the CPU times (denoted by "CPU") which is in seconds. All systems are solved by applying MATLAB R2015b on a PC with Intel(R) Core (TM) i7 CPU 4.20 GHz and 8.0 GB memory. In both examples, we select $P = 0.01H$, $Q = 0.1I_{n \times n}$, that $H = \frac{1}{2}(A + A^T)$, the vector $x^{(0)} = (0, 0, \ldots, 0)$ is used as initial guess and the right-hand side vector is chosen $b = rand(m + n, 1)$ except for Table 1, it is chosen in a way that $(1, 1, \ldots, 1)^T$ is the exact solution of (1). In Table 1, the optimal parameters have been determined experimentally such that, it results in the least number of iterations. The computation of the optimal parameter is often problem-dependent, so in this table the right hand side vector is not chosen random vector.

TABLE 5 Numerical results of the GMRES method for $\mu = 1$ and $p = 128$ with five preconditioners.

Preconditioner $(\hat{\alpha}, \hat{\beta})$		$\mathcal{P}_{PIU}$	$\mathcal{P}_{GMSS}$	$\mathcal{P}_{MGSS}$	$\mathcal{P}_{PGSS(l=5)}$	$\mathcal{P}_{PESS(l=5)}$
(0.6, 0.8)	IT.	52	19	12	9	4
	CPU	5.287	3.918	2.547	1.859	1.316
(0.2, 0.5)	IT.	47	18	10	7	3
	CPU	4.831	3.762	2.155	1.415	1.149
(0.2, 1.3)	IT.	45	21	14	9	5
	CPU	4.608	4.240	2.984	2.049	1.279
(1, 0.8)	IT.	53	19	12	9	4
	CPU	5.173	3.887	2.576	1.819	1.182
(1.2, 1.5)	IT.	54	22	15	11	5
	CPU	5.404	4.458	3.092	2.267	1.284
(1.5, 1.2)	IT.	53	21	14	10	5
	CPU	5.359	4.251	3.168	1.997	1.247
(1.8, 1.5)	IT.	55	22	15	11	5
	CPU	5.300	4.515	3.153	2.224	1.247
(0.02, 0.5)	IT.	31	18	10	7	47
	CPU	3.169	3.635	2.240	1.481	1.114
(0.89, 1.37)	IT.	52	21	14	10	5
	CPU	5.068	4.250	2.964	2.010	1.263
(0.93, 0.16)	IT.	52	14	7	6	3
	CPU	5.132	2.932	1.631	1.282	0.918
(0.01, 0.2)	IT.	19	17	5	6	3
	CPU	1.878	3.098	1.667	1.296	1.158

Example 1. Consider the saddle point problem (1) with the following matrices[18,27]

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_1 \end{pmatrix} \in \mathbb{R}^{2p^2 \times 2p^2}, \qquad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \in \mathbb{R}^{2p^2 \times p^2},$$

where $A_1 = I \otimes T + T \otimes I$, $B_1 = I \otimes F$, $B_2 = F \otimes I$, and

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

where $h = \frac{1}{p+1}$, $\otimes$ denotes the Kronecker product and μ indicates the viscosity.

All runs terminate when the number of iterations exceeds $\kappa_{\max} = 500$ or $RES < 10^{-6}$, where

$$RES = \frac{\sqrt{\|f - A\hat{x}^{(k)} - B\hat{y}^{(k)}\|_2^2 + \|g - B^T\hat{x}^{(k)}\|_2^2}}{\sqrt{\|f\|_2^2 + \|g\|_2^2}} < 10^{-6}.$$

The termination criterion of inner GMRES method is $\frac{\|r^{(k)}\|}{\|r^{(0)}\|} < 10^{-7}$, in which $r^{(k)}$ is the residual of the kth GMRES iteration. We list numerical results of Example 1 in Tables 1–8.

TABLE 6 Numerical results of the GMRES method for $\mu = 0.1$ and $\hat{\alpha} = 0.8$ with five preconditioners.

	Preconditioner	$\hat{\beta}$	0.05	0.2	0.9	1.3	2.6
$P = 32$	$\mathcal{P}_{GMSS}$	IT.	13	14	17	18	21
		CPU	0.290	0.324	0.410	0.438	0.462
	$\mathcal{P}_{MGSS}$	IT.	4	5	7	8	8
		CPU	0.073	0.078	0.101	0.108	0.209
	$\mathcal{P}_{PIU}$	IT.	76	69	65	65	69
		CPU	1.875	1.679	1.688	1.512	1.663
	$\mathcal{P}_{PGSS}(l = 6)$	IT.	4	4	5	5	6
		CPU	0.053	0.066	0.071	0.072	0.088
	$\mathcal{P}_{PESS}(l = 6)$	IT.	2	3	3	3	4
		CPU	0.042	0.048	0.050	0.066	0.055
	$\mathcal{P}_{PGSS}(l = 8)$	IT.	3	4	5	5	6
		CPU	0.049	0.053	0.064	0.080	0.078
	$\mathcal{P}_{PESS}(l = 8)$	IT.	2	3	3	3	3
		CPU	0.036	0.043	0.054	0.050	0.050
$P = 128$	$\mathcal{P}_{GMSS}$	IT.	15	17	20	20	26
		CPU	3.157	3.477	4.097	4.383	4.974
	$\mathcal{P}_{MGSS}$	IT.	4	5	7	8	11
		CPU	1.071	1.292	1.663	1.800	2.314
	$\mathcal{P}_{PIU}$	IT.	100	98	93	97	88
		CPU	10.075	9.762	9.067	9.371	8.637
	$\mathcal{P}_{PGSS}(l = 6)$	IT.	4	5	6	7	7
		CPU	0.873	1.096	1.299	1.408	1.440
	$\mathcal{P}_{PESS}(l = 6)$	IT.	3	3	3	4	4
		CPU	0.497	0.906	0.928	0.959	1.088
	$\mathcal{P}_{PGSS}(l = 8)$	IT.	5	5	6	6	7
		CPU	0.856	1.081	1.230	1.240	1.478
	$\mathcal{P}_{PESS}(l = 8)$	IT.	2	3	3	4	4
		CPU	0.482	0.870	0.902	0.916	1.113

In Table 1 of this example, for *GMSS*, *MGSS*, *PGSS*, and *PESS* iteration methods, we solve linear subsystems $(\hat{\alpha}I + 2H + \frac{1}{\hat{\beta}}BB^T)x = b$, $(\hat{\alpha}I + 2A + \frac{4}{\hat{\beta}}BB^T)x = b$, $(\hat{\alpha}I + lA + \frac{l^2}{\hat{\beta}}BB^T)x = b$ and $(\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T)x = b$, respectively, using the LU factorization with combination column AMD reordering. In this table, the optimal parameters of *MGSS* and *GMSS* have been obtained experimentally, while we find $\hat{\alpha}$, $\hat{\beta}$, and l for the *PESS* iteration method and for the *PGSS* method, as a special case of the *PESS* method with $P = Q = I$, so that $\rho(\Gamma(\hat{\alpha}, \hat{\beta}, l))$ be minimized. For this aim, in a similar way of Reference 18, the following function is minimized.

$$\begin{aligned}\tau(\hat{\alpha}, \hat{\beta}, l) &= \|Q_{PESS}\|_F^2 \\ &= \hat{\alpha}^2 tr(P^TP) + 2\hat{\alpha}(l-1)tr(P^TA) + (l-1)^2 tr(A^TA) \\ &+ 2(l-1)^2 tr(B^TB) + \hat{\beta}^2 tr(Q^TQ) > 0.\end{aligned}$$

Now we select parameters $\hat{\alpha}$, $\hat{\beta}$, and l to make $\tau(\hat{\alpha}, \hat{\beta}, l)$ as small as possible. Since

$$\lim_{\hat{\alpha},\hat{\beta}\to 0_+} \tau(\hat{\alpha}, \hat{\beta}, l) = (l-1)^2 tr(A^TA)(l-1)^2 tr(A^TA).$$

TABLE 7 Numerical results of the GMRES method for $\mu = 0.1$ and $\hat{\beta} = 0.5$ with six preconditioners.

	Preconditioner	$\hat{\alpha}$	0.01	0.3	0.8	1.5	2.7	5
$P = 32$	$\mathcal{P}_{GMSS}$	IT.	16	16	16	16	16	16
		CPU	0.379	0.397	0.388	0.393	0.391	0.387
	$\mathcal{P}_{MGSS}$	IT.	5	5	5	6	6	7
		CPU	0.075	0.139	0.170	0.095	0.194	0.199
	$\mathcal{P}_{PIU}$	IT.	50	67	72	66	66	69
		CPU	1.102	1.507	1.736	1.469	1.460	1.503
	$\mathcal{P}_{DPSS}$	IT.	45	59	64	67	69	69
		CPU	4.125	5.266	5.561	5.796	6.320	6.448
	$\mathcal{P}_{PGSS}(l = 6)$	IT.	4	4	5	5	5	5
		CPU	0.054	0.060	0.063	0.70	0.077	0.077
	$\mathcal{P}_{PESS}(l = 6)$	IT.	3	3	3	3	3	3
		CPU	0.046	0.042	0.052	0.053	0.050	0.061
	$\mathcal{P}_{PGSS}(l = 9)$	IT.	4	4	4	4	5	5
		CPU	0.059	0.056	0.063	0.060	0.064	0.074
	$\mathcal{P}_{PESS}(l = 9)$	IT.	3	3	3	3	3	3
		CPU	0.054	0.051	0.054	0.054	0.060	0.068
	$\mathcal{P}_{PGSS}(l = 4)$	IT.	4	5	5	5	6	6
		CPU	0.063	0.059	0.072	0.077	0.085	0.092
	$\mathcal{P}_{PESS}(l = 4)$	IT.	3	3	3	3	4	4
		CPU	0.051	0.052	0.051	0.058	0.055	0.052
$P = 128$	$\mathcal{P}_{GMSS}$	IT.	18	18	18	19	19	19
		CPU	3.121	3.569	3.696	3.723	3.743	3.713
	$\mathcal{P}_{MGSS}$	IT.	6	6	6	7	7	8
		CPU	1.253	1.404	1.460	1.643	1.598	1.814
	$\mathcal{P}_{PIU}$	IT.	48	89	90	90	94	98
		CPU	4.696	8.471	8.476	8.543	9.120	9.446
	$\mathcal{P}_{DPSS}$	IT.	99	153	167	176	183	192
		CPU	55.895	78.383	93.470	88.886	92.346	97.884
	$\mathcal{P}_{PGSS}(l = 6)$	IT.	4	5	5	6	6	7
		CPU	0.919	1.131	1.252	1.298	1.385	1.485
	$\mathcal{P}_{PESS}(l = 6)$	IT.	3	3	3	3	3	3
		CPU	0.0909	0.928	0.938	0.945	0.969	0.972
	$\mathcal{P}_{PGSS}(l = 9)$	IT.	4	5	5	5	6	6
		CPU	0.936	1.081	1.060	1.085	1.260	1.375
	$\mathcal{P}_{PESS}(l = 9)$	IT.	3	3	3	3	3	3
		CPU	0.911	0.935	0.952	0.973	1.090	1.206
	$\mathcal{P}_{PGSS}(l = 4)$	IT.	5	6	6	6	7	8
		CPU	1.087	1.107	1.127	1.198	1.421	1.625
	$\mathcal{P}_{PESS}(l = 4)$	IT.	3	3	3	3	3	4
		CPU	0.840	0.899	0.909	0.926	0.9302	1.082

TABLE 8 Condition number of $M^{-1}\mathfrak{A}$ with $p = 48$ for *PGSS* and *PESS*

Method	$\mu = 0.1$	$\mu = 1$
$PGSS(l = 5)$	29.0623	18.0808
$PESS(l = 5)$	4.1820	2.5131
$PGSS(l = 6)$	29.0828	18.0931
$PESS(l = 6)$	4.0758	2.5132
$PGSS(l = 1)$	29.0467	17.7931
$PESS(l = 1)$	8.1418	2.5126

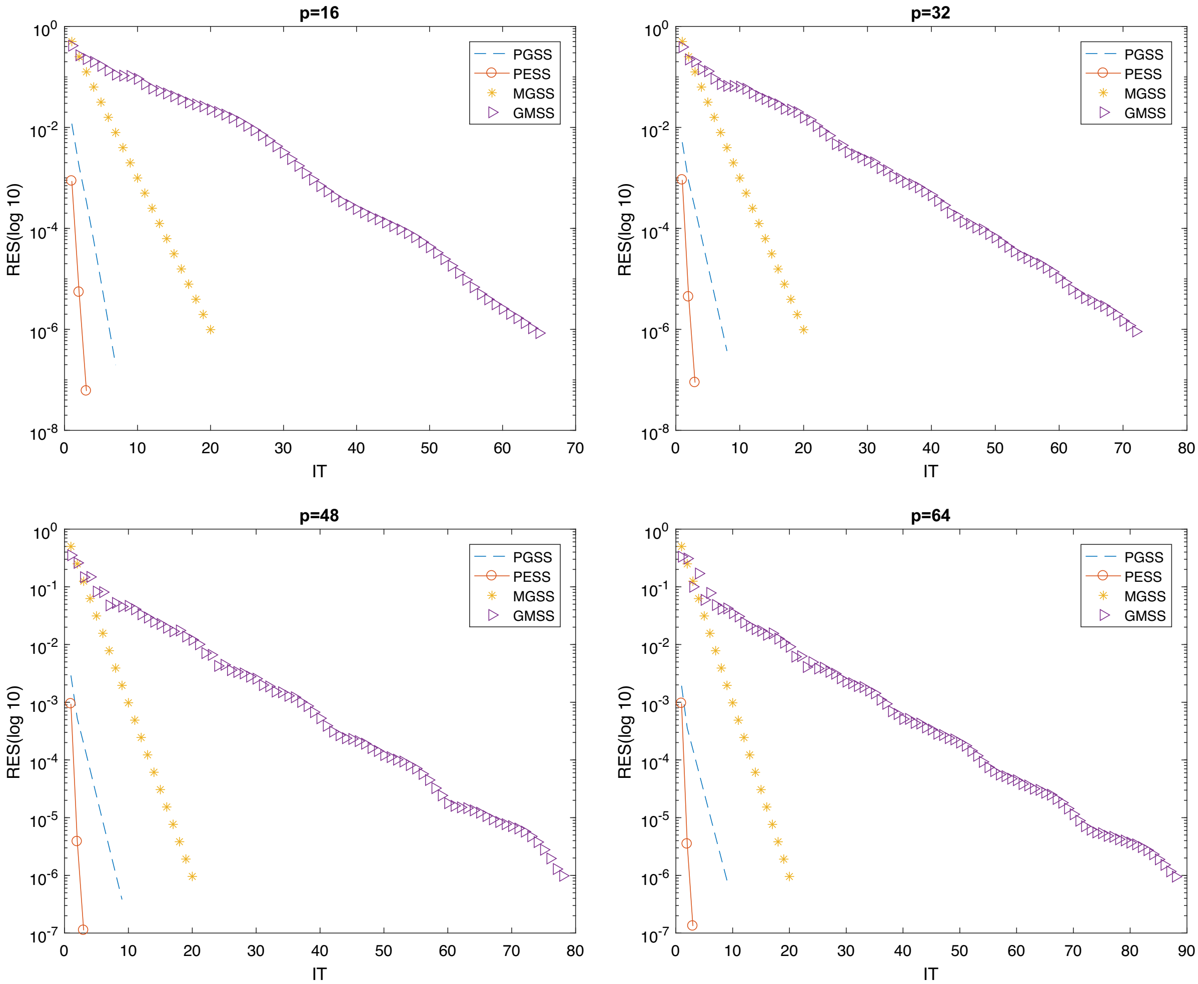

FIGURE 1 Convergence curves for $p = 16, 32, 48$, and 64 with $\mu = 0.1$

We can select $l = 1$ and $\hat{\alpha}, \hat{\beta} \to 0_+$ such that $\tau(\hat{\alpha}, \hat{\beta}, l) \to 0_+$ and $Q_{PESS} \to 0$, so it is easy to see that $\tau(\hat{\alpha}, \hat{\beta}, l) = \hat{\alpha}^2 tr(P^T P) + \hat{\beta}^2 tr(Q^T Q)$ as $l = 0$. The values of $\hat{\alpha}, \hat{\beta}$ are chosen small enough such that $\tau(\hat{\alpha}, \hat{\beta}, l)$ is as small as possible, but β is not so small that $\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}} BQ^{-1}B^T$ is very ill-conditioned. The symbol of exp indices indicate that the optimal parameters are found experimentally.

Table 1 lists the numerical results of the various methods with respect to different problem sizes and $\mu = 0.1$ for Example 1. The efficiency of the *PESS* method is shown with $l = 1$ and small values for both $\hat{\alpha}$ and $\hat{\beta}$. The processing time and iteration numbers of *PESS* iteration method for solving Example 1 is less than those of the other methods. The IT of *GMSS* in comparison with the *MGSS*, *PGSS*, and *PESS* methods shows more sensitivity to the value of p.

Tables 2 and 3, present numerical experiments of the *GMSS*, *MGSS*, *PIU*, *DPSS*, *PGSS*, and *PESS* preconditioned GMRES method with $\mu = 0.1, 1$ on different uniform grids. The GMRES method without preconditioning is indicated by I in these tables. The notation - is used to show that the method until $\kappa_{\max}$ iterations, does not satisfy the $RES < 10^{-6}$. Moreover, parameters considered for the chosen preconditioners are

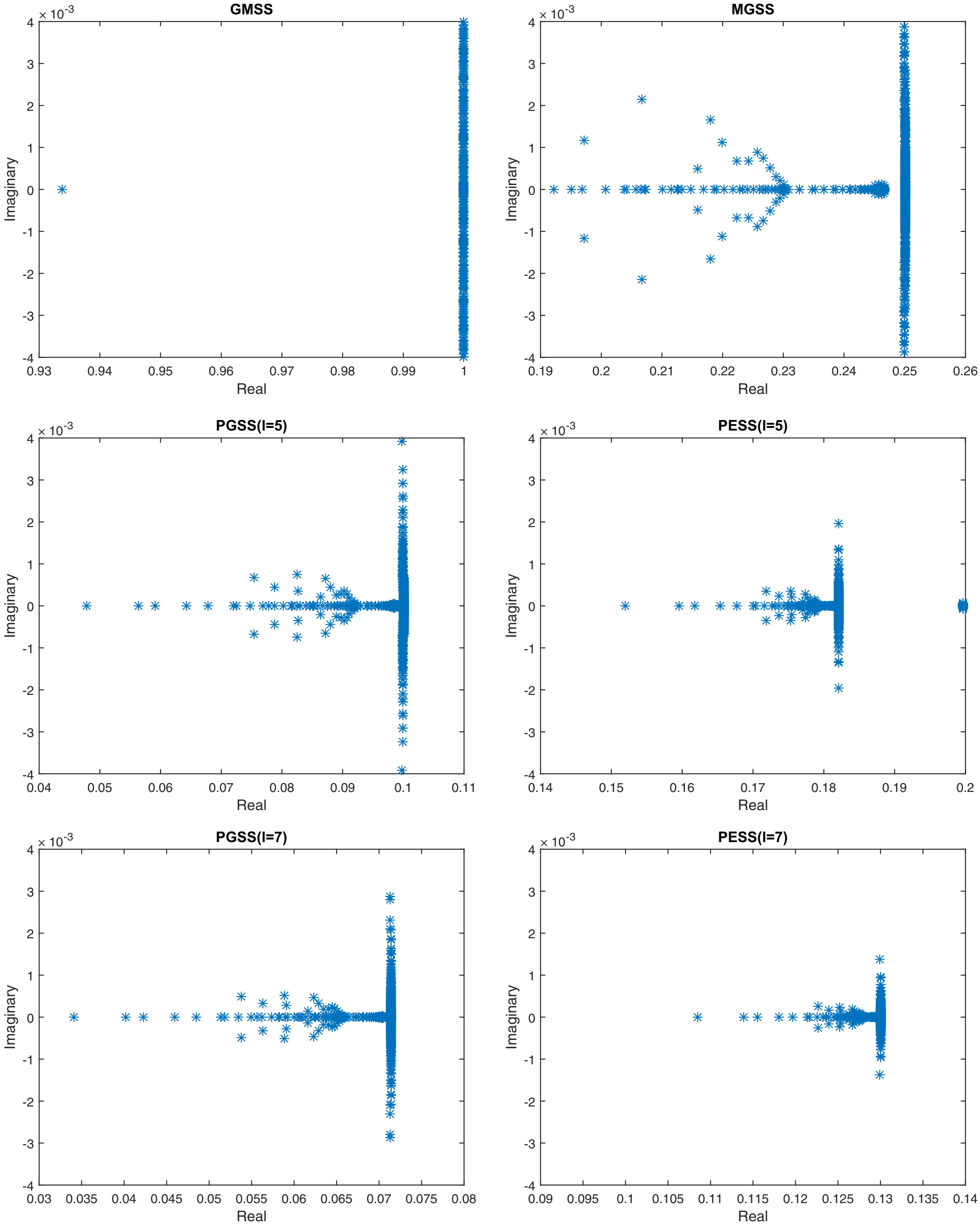

FIGURE 2 The eigenvalue distributions of the five preconditioned matrices for $\mu = 1$ and $p = 32$.

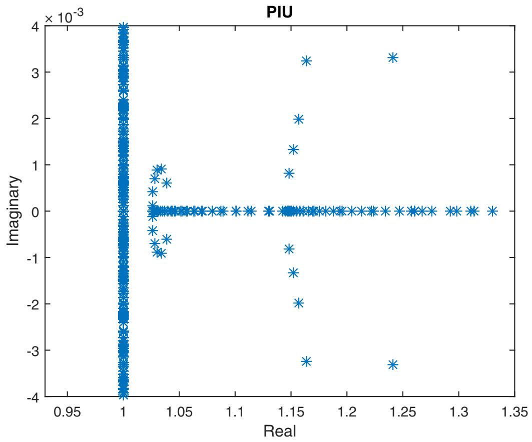

FIGURE 2 (Continued)

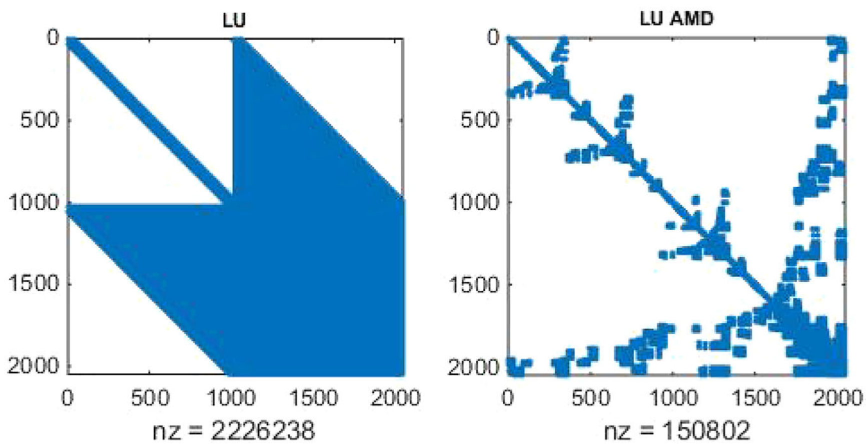

FIGURE 3 The pattern of nonzero entries of LU factorization (left) and LU factorization with combination column AMD reordering (right) for PESS with $\mu = 0.1$ and $p = 32$.

given from References 18 and 53:

$$\begin{aligned}
&\hat{\alpha}_{GMSS} = \mu, \quad \hat{\beta}_{GMSS} = \frac{\|B\|_2^2}{2\|H\|_2}; \qquad \hat{\alpha}_{MGSS} = \mu, \quad \hat{\beta}_{MGSS} = \frac{2\|B\|_2^2}{\|A\|_2};\\
&\hat{\alpha}_{PIU} = \mu, \quad \hat{\beta}_{PIU} = \frac{\|B\|_2^2}{\|A\|_2}; \qquad \hat{\alpha}_{DPSS} = \mu;\\
&\hat{\alpha}_{PGSS} = \mu, \quad \hat{\beta}_{PGSS} = \frac{l\|B\|_2^2}{\|A\|_2}; \qquad \hat{\alpha}_{PESS} = \mu, \quad \hat{\beta}_{PESS} = \frac{l\|B\|_2^2}{\|A\|_2}.
\end{aligned} \tag{33}$$

We can use $\hat{\mathcal{P}}_{PESS} = \begin{pmatrix} \frac{\hat{\alpha}}{l}P + A & B \\ -B^T & \frac{\hat{\beta}}{l}Q \end{pmatrix}$ as a scaled *PESS* preconditioner. For the given parameters $\hat{\alpha} \geq 0$, $\hat{\beta} > 0$, we have

$$\lim_{l \to +\infty} (\hat{\mathcal{P}}_{PESS} - \mathcal{A}) = \lim_{l \to +\infty} \left[\frac{1}{l} \begin{pmatrix} \hat{\alpha}P & 0 \\ 0 & \hat{\beta}Q \end{pmatrix} \right] = 0.$$

Also, because of ill-conditioning of $\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T$, we should not select a large value for l and the selection of its optimal parameter is difficult. For Tables 2 and 3, the parameter $l = 3, 6, 8$ and $l = 5, 7$ is chosen, respectively.

TABLE 9 Results for the Oseen problem with different uniform grids ($Q2-Q1FEM$, $\nu = 0.1, \hat{\alpha} = \nu$).

Preconditioner	Grids	16 × 16	32 × 32	64 × 64
$\mathcal{P}_{DPSS}$	IT.	38	86	226
	CPU	0.201	2.415	29.272
$\mathcal{P}_{PIU}$	$\hat{\beta}$	0.0592	0.0153	0.0038
	IT.	67	76	33
	CPU	0.075	0.489	1.072
$\mathcal{P}_{GMSS}$	$\hat{\beta}$	0.0296	0.0076	0.0019
	IT.	24	41	79
	CPU	0.085	0.548	5.345
$\mathcal{P}_{MGSS}$	$\hat{\beta}$	0.1184	0.0305	0.0077
	IT.	19	35	68
	CPU	0.067	0.469	4.658
$\mathcal{P}_{PGSS}(l=6)$	$\hat{\beta}$	0.3553	0.0915	0.0231
	IT.	16	22	41
	CPU	0.062	312	2.851
$\mathcal{P}_{PESS}(l=6)$	$\hat{\beta}$	0.3553	0.0915	0.0231
	IT.	6	6	6
	CPU	0.025	0.098	0.487
$\mathcal{P}_{PGSS}(l=3)$	$\hat{\beta}$	0.1776	0.0458	0.0115
	IT.	18	29	57
	CPU	0.234	1.487	3.891
$\mathcal{P}_{PESS}(l=3)$	$\hat{\beta}$	0.1776	0.0458	0.0115
	IT.	6	6	6
	CPU	0.033	0.102	0.591

Tables 2 and 3 show that the $\mathcal{P}_{PESS}$ preconditioned GMRES is feasible and efficient.

As further evidence of the advantage of the *PESS* preconditioner over the *GMSS*, *MGSS*, *PGSS*, and *PIU* preconditioners for GMRES method, numerical results for Example (1) with different values of $\hat{\alpha}, \hat{\beta}$ are presented in Tables 4 and 5.

Tables 4 and 5 give numerical experiments for several preconditioned GMRES methods with various parameters $\hat{\alpha}, \hat{\beta}$ and $p = 128$ with $\mu = 0.1, 1$, respectively. The optimal parameters of the PIU preconditioner have been experimentally obtained, and this value is (0.01, 0.2). As we can see, the *PESS* preconditioned GMRES method takes less IT and CPU time for different values of parameters $\hat{\alpha}, \hat{\beta}$ than the other preconditioners, which means that the *PESS* preconditioner accelerates GMRES convergence for Example 1 more than the other preconditioners.

Table 6 gives the numerical results for the constant $\hat{\alpha} = 0.8$ and various values of $\hat{\beta}$. This table shows that the CPU time increases with increasing $\hat{\beta}$ for all methods except *PIU*. Table 7 presents with different values of $\hat{\alpha}$, the constant $\hat{\beta} = 0.5$, and $p = 32,128$. Table 8 presents the condition number of $M^{-1}\mathfrak{A}$ for the *PGSS* and *PESS* methods with the values of $\hat{\alpha}, \hat{\beta}$ from (33). This table shows with the appropriate selection of matrices P, Q and parameters $\hat{\alpha}, \hat{\beta}$, and l, the condition numbers of $M^{-1}\mathfrak{A}$ for *PESS* method is less than those of the *PGSS* method.

In Figure 1, the convergence curves are given for *GMSS*, *MGSS*, *PGSS*, and *PESS* methods with $\mu = 0.1$, for $p = 16$, 32, 48, and 64, respectively, with the parameters in Table 1. It shows that the *PESS* method gives a better convergence rate for all values of p.

In Figure 2, the eigenvalue distributions of the *GMSS*, *MGSS*, *PGSS*, *PIU*, and *PESS* preconditioned matrices as given in Table 3 for $\mu = 1$ and $p = 32$ are shown. According to Figure 2, the *PESS* preconditioned matrix has a better eigenvalue clustering as compared to the others. $\mathcal{P}_{PESS}^{-1}\mathfrak{A}$ has eigenvalues positioned within a circle with a radius $\frac{1}{l}$. Comparing the spy plot of the LU factorization and LU factorization with combination column AMD reordering of the $\hat{\alpha}P + lA + \frac{l^2}{\hat{\beta}}BQ^{-1}B^T$ in

TABLE 10 Results for the Oseen problem with different uniform grids ($Q2 - Q1FEM$, $\nu = 1$, $\hat{\alpha} = \nu$).

Preconditioner	Grids	16 × 16	32 × 32	64 × 64
$\mathcal{P}_{DPSS}$	IT.	72	158	330
	CPU	0.397	3.837	43.548
$\mathcal{P}_{PIU}$	$\hat{\beta}$	0.0078	0.0020	$5.013e-4$
	IT.	25	17	4
	CPU	0.030	0.134	0.196
$\mathcal{P}_{GMSS}$	$\hat{\beta}$	0.2256	0.0584	0.0148
	IT.	42	71	124
	CPU	0.148	0.942	8.760
$\mathcal{P}_{MGSS}$	$\hat{\beta}$	0.0155	0.0040	0.0010
	IT.	14	24	45
	CPU	0.051	0.361	3.055
$\mathcal{P}_{PGSS}(l = 5)$	$\hat{\beta}$	0.0389	0.0100	0.0025
	IT.	10	16	30
	CPU	0.039	0.247	2.187
$\mathcal{P}_{PESS}(l = 5)$	$\hat{\beta}$	0.0389	0.0100	0.0025
	IT.	4	5	5
	CPU	0.017	0.083	0.397
$\mathcal{P}_{PGSS}(l = 8)$	$\hat{\beta}$	0.0620	0.0159	0.0040
	IT.	10	14	25
	CPU	0.037	0.213	1.876
$\mathcal{P}_{PESS}(l = 8)$	$\hat{\beta}$	0.0620	0.0159	0.0040
	IT.	4	5	5
	CPU	0.016	0.083	0.408

Figure 3, shows that minimum degree reduces the nonzero number by a factor of 14.76. The nonzero counts are 2,226,238 and 150,802, respectively.

Example 2. Using the Picard iteration to linearize the steady incompressible Navier–Stokes equation gets the Oseen equation:

$$\begin{cases} -\nu\Delta u + w.\nabla u + \nabla p = f, \\ \nabla .u = 0, \end{cases}$$

in $\Omega = (-1, 1) \times (-1, 1) \subset \mathbb{R}^2$. The Δ and ∇ are the Laplacian operator and gradient, respectively, and $\upsilon > 0$ stands for viscosity. Here, the w is the approximation of u from the previous Picard iteration. The linear system is taken at the ninth Picard iteration. The test problem is a two-dimensional leaky-lid driven cavity problem on Ω and it is discretized by the $Q2 - Q1$ mixed finite element method on a uniform grid. We generate the test problem with the IFISS software written by Elman et al.[62] to generate linear systems corresponding to 16×16, 32×32, and 64×64 meshes. For each grid, we test two viscosity values, that is, $\nu = 1$ and $\nu = 0.1$.

All runs are started from the initial zero vector and terminated if the current iterations satisfy $\frac{\|r^{(k)}\|}{\|r^{(0)}\|} < 10^{-6}$, where $r^{(k)}$ is the residual at the kth iteration, or if the number of iterations exceeded $\kappa_{\max} = 1500$. In this example, we first select $l = 3, 6$ for Table 9, $l = 5, 8$ for Table 10, then $\hat{\alpha} = \nu$ and $\hat{\beta}$ obtain using (33).

In Tables 9 and 10, numerical experiments demonstrate that the *PESS* iteration method consistently performs better than the other five iteration methods in terms of IT and CPU times, and this advantage is accentuated as system size

TABLE 11 Numerical results of the GMRES methods for $\mu = 1$ with five preconditioners

	Preconditioner $(\hat{\alpha}, \hat{\beta})$		$\mathcal{P}_{PIU}$	$\mathcal{P}_{GMSS}$	$\mathcal{P}_{MGSS}$	$\mathcal{P}_{PGSS(l=5)}$	$\mathcal{P}_{PESS(l=5)}$
$p = 32$	(0.6, 0.8)	IT.	31	100	99	67	18
		CPU	1.263	7.933	7.933	5.071	0.478
	(0.2, 1.3)	IT.	37	76	74	55	20
		CPU	1.535	5.914	5.850	4.174	0.516
	(1.2, 1.5)	IT.	29	135	132	89	22
		CPU	1.217	10.728	10.405	7.055	0.570
	(0.02, 0.5)	IT.	40	46	41	33	16
		CPU	1.657	3.341	3.200	2.543	0.427
	(0.89, 1.37)	IT.	28	122	118	81	23
		CPU	1.176	9.630	9.484	6.244	0.511
	(0.93, 0.16)	IT.	23	90	82	51	11
		CPU	0.683	6.876	6.247	3.955	0.366
$p = 128$	(0.6, 0.8)	IT.	29	409	409	264	29
		CPU	10.815	143.385	136.824	87.628	9.664
	(0.2, 1.3)	IT.	36	275	274	181	29
		CPU	12.666	90.708	90.462	58.849	9.81
	(1.2, 1.5)	IT.	28	561	560	365	30
		CPU	10.513	192.700	190.131	121.083	10.008
	(0.02, 0.5)	IT.	40	111	109	79	27
		CPU	11.349	35.946	35.448	25.692	9.296
	(0.89, 1.37)	IT.	28	499	496	324	30
		CPU	11.506	172.840	168.213	106.899	10.084
	(0.93, 0.16)	IT.	25	384	380	244	26
		CPU	9.621	129.547	126.919	82.676	8.714

increases. By utilizing the preconditioners stated above, the GMRES method converges quickly, and the proposed *PESS* preconditioned GMRES method converges faster than the other five preconditioned GMRES methods. Furthermore, we see that the *PESS* preconditioned GMRES method's convergence behavior is not overly sensitive to the size of the problem.

Table 11 presents numerical experiments of Example 2 for several preconditioned GMRES methods with various parameters $\hat{\alpha}, \hat{\beta}$ and $p = 32{,}128$ with $\mu = 1$.

6 | CONCLUSIONS

In this work, we presented a parameterized extended shift-splitting (PESS) method and the *PESS* preconditioner for solving (1) and theoretically discussed the convergence behavior of *PESS* method. Also, eigen properties of the corresponding preconditioned matrix are investigated. The numerical results show the superiority of *PESS* method in terms of IT and CPU times for solving saddle point problem compared to *PGSS*, *MGSS*, *GMSS*, *PIU*, and *DPSS* methods with the appropriate choice of P, Q.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous referees for their valuable suggestions and comments which improved this article.

CONFLICT OF INTEREST

This study does not have any conflicts to disclose.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

Ghodrat Ebadi https://orcid.org/0000-0002-0214-9314
Cornelis Vuik https://orcid.org/0000-0002-5988-9153

REFERENCES

1. Bai Z-Z. Structured preconditioners for nonsingular matrices of block two-by-two structures. Math Comput. 2006;75(254):791–815.
2. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. Acta Numer. 2005;14:1–137.
3. Zhang N-M, Lu T-T, Wei Y-M. Semi-convergence analysis of Uzawa methods for singular saddle point problems. J Comput Appl Math. 2014;255:334–45.
4. Dou Y, Yang A-L, Wu Y-J. Modified parameterized inexact Uzawa method for singular saddle-point problems. Numer Algorithms. 2016;72:325–39.
5. Bai Z-Z. On semi-convergence of Hermitian and skew-Hermitian splitting methods for singular linear systems. Computing. 2010;89:171–97.
6. Huang Z-G, Wang L-G, Xu Z, Cui J-J. Improved PPHSS iterative methods for solving nonsingular and singular saddle point problems. Comput Math Appl. 2016;72:92–109.
7. Chao Z, Zhang N-M. A generalized preconditioned HSS methods for singular saddle point problems. Numer Algorithms. 2014;66:203–21.
8. Zheng B, Bai Z-Z, Yang X. On semi-convergence of parameterized Uzawa methods for singular saddle point problems. Linear Algebra Appl. 2009;431:808–17.
9. Li J, Zhang N-M. A triple-parameter modified SSOR method for solving singular saddle point problems. BIT Numer Math. 2016;56:501–21.
10. Wang S-S, Zhang G-F. Preconditioned AHSS iteration method for singular saddle point problems. Numer Algorithms. 2013;63:521–35.
11. Salkuyeh DK, Rahimian M. A modification of the generalized shift-splitting method for singular saddle point problems. Comput Math Appl. 2017;74:2940–9.
12. Rahimian M, Salkuyeh DK. Spectral analysis of the *MGSS* preconditioner for singular saddl point problems. J Appl Math Inform. 2020;38:175–87.
13. Shen Q-Q, Shi Q. Generalized shift-splitting preconditioners for nonsingular and singular generalized saddle point problems. Comput Math Appl. 2016;72:632–41.
14. Chen C-R, Ma C-F. A generalized shift-splitting preconditioner for singular saddle point problems. Appl Math Comput. 2015;269:947–55.
15. Cao Y, Miao S-X. On semi-convergence of the generalized shift-splitting iteration method for singular nonsymmetric saddle point problems. Comput Math Appl. 2016;71:1503–11.
16. Huang Z-G, Wang L-G, Xu Z, Cui J-J. The generalized modified shift-splitting preconditioners for nonsymmetric saddle point problem. Appl Math Comput. 2017;299:95–118.
17. Huang Z-G, Wang L-G, Xu Z, Cui J-J. A modified generalized shift-splitting preconditioner for nonsymmetric saddle point problems. Numer Algorithms. 2018;78:297–331.
18. Huang Z-G, Wang L-G, Xu Z, Cui J-J. Parameterized generalized shift-splitting preconditioner for nonsymmetric saddle point problems. Comput Math Appl. 2018;75:349–73.
19. Bai Z-Z, Yin J-F, Su Y-F. A shift-splitting preconditioner for non-Hermitian positive definite matrices. J Comput Math. 2006;24(4):539–52.
20. Elman HC, Golub GH. Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J Numer Anal. 1994;31(6):1645–61.
21. Bai Z-Z, Parlett BN, Wang Z-Q. On generalized successive overrelaxation methods for augmented linear systems. Numer Math. 2005;102:1–38.
22. Bai Z-Z, Wang Z-Q. On parameterized inexact Uzawa methods for generalized saddle point problems. Linear Algebra Appl. 2008;428:2900–32.
23. Dai L, Liang M, Fan H. A new generalized parameterized inexact Uzawa method for solving saddle point problems. Appl Math Comput. 2015;265:414–30.
24. Golub GH, Wu X, Yuan J-Y. SOR-like methods for augmented systems. BIT. 2001;41(1):71–85.
25. Njeru PN, Guo X-P. Accelerated SOR-like method for augmented linear systems. BIT Numer Math. 2016;56:557–71.
26. Guo P, Li C-X, Wu S-L. A modified SOR-like method for augmented systems. J Comput Appl Math. 2015;274:58–69.
27. Bai Z-Z, Golub GH, Pan J-Y. Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer Math. 2004;98:1–32.
28. Bai Z-Z. Optimal parameters in the HSS-like methods for saddle point problems. Numer Linear Algebra Appl. 2009;16:447–79.
29. Bai Z-Z, Golub GH. Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J Numer Anal. 2007;27:1–23.
30. Bai Z-Z, Golub GH, Ng MK. On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. Linear Algebra Appl. 2008;428:413–40.

31. Bai Z-Z, Golub GH, Li C-K. Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices. SIAM J Sci Comput. 2006;28(2):583–603.
32. Fischer B, Ramage A, Silvester DJ, Wathen AJ. Minimum residual methods for augmented systems. BIT. 1998;38(3):527–43.
33. Bai Z-Z. Motivations and realizations of Krylov subspace methods for large sparse linear systems. J Comput Appl Math. 2015;283:71–8.
34. Ebadi G, Alipour N, Vuik C. Deflated and augmented global Krylov subspace methods for the matrix equations. Appl Numer Math. 2016;99:137–50.
35. Ebadi G, Rashedi S. New variants of the global Krylov type methods for linear systems with multiple right-hand sides arising in elliptic PDEs. Comput Methods Differ Equ. 2018;6(2):111–27.
36. Keller C, Gould NIM, Wathen AJ. Constraint preconditioning for indefinite linear systems. SIAM J Matrix Anal Appl. 2000;21(4):1300–17.
37. Bai Z-Z, Ng MK, Wang Z-Q. Constraint pre-conditioners for symmetric indefinite matrices. SIAM J Matrix Anal Appl. 2009;31(2):410–33.
38. Bai Z-Z, Golub GH, Ng MK. Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J Matrix Anal Appl. 2003;24(3):603–26.
39. Benzi M, Golub GH. A preconditioner for generalized saddle point problems. SIAM J Matrix Anal Appl. 2004;26(1):20–41.
40. Cao Y, Yao L, Jiang M, Niu Q. A relaxed HSS preconditioner for saddle point problems from meshfree discretization. J Comput Math. 2013;31(4):398–421.
41. Pan J-Y, Ng MK, Bai Z-Z. New preconditioners for saddle point problems. Appl Math Comput. 2006;172:762–71.
42. Cao Y, Dong J-L, Wang Y-M. A relaxed deteriorated PSS preconditioner for nonsymmetric saddle point problems from the steady Navier-Stokes equation. J Comput Appl Math. 2015;273:41–60.
43. Cao Y, Yao L-Q, Jiang M-Q. A modified dimensional split preconditioner for generalized saddle point problems. J Comput Appl Math. 2013;250:70–82.
44. Benzi M, Guo X-P. A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations. Appl Numer Math. 2011;61:66–76.
45. Perugia I, Simoncini V. Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. Numer Linear Algebra Appl. 2000;7:585–616.
46. Bai Z-Z, Ng MK. On inexact preconditioners for nonsymmetric matrices. SIAM J Sci Comput. 2005;26:1710–24.
47. Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. SIAM J Sci Comput. 2000;21(6):1969–72.
48. He X, Vuik C, Klaij C. Block-preconditioners for the incompressible Navier-Stokes equations discretized by a finite volume method. J Numer Math. 2017;25(2):89–105.
49. Vandenplas J, Eding H, Calus MPL, Vuik C. Defated preconditioned conjugate gradient method for solving single-step BLUP models efciently. Genet Sel Evol. 2018;50(1):1–17.
50. Hornikova H, Vuik C, Egermaier J. A comparison of block preconditioners for isogeometric analysis discretizations of the incompressible Navier-Stokes equations. Int J Numer Methods Fluids. 2021;93(6):1788–815.
51. Cao Y, Du J, Niu Q. Shift-splitting preconditioners for saddle point problems. J Comput Appl Math. 2014;272:239–50.
52. Chen C, Ma C-F. A generalized shift-splitting preconditioner for saddle point problems. Appl Math Lett. 2015;43:49–55.
53. Cao Y, Li S, Yao L-Q. A class of generalized shift-splitting preconditioners for nonsymmetric saddle point problems. Appl Math Lett. 2015;49:20–7.
54. Cao Y, Miao S-X, Ren Z-R. On preconditioned generalized shift-splitting iteration methods for saddle point problems. Comput Math Appl. 2017;74:859–72.
55. Zhou S-W, Yang A-L, Dou Y, Wu Y-J. The modified shift-splitting preconditioners for nonsymmetric saddle-point problems. Appl Math Lett. 2016;59:109–14.
56. Huang Z-H, Su H. A modified shift splitting method for nonsymmetric saddle point problems. J Comput Appl Math. 2017;317:535–46.
57. Zheng Q-Q, Lu L-Z. Extended shift-splitting preconditioners for saddle point problems. J Comput Appl Math. 2017;313:70–81.
58. Li J. Extanded shift splitting iteration method for nonsymmetric generalized saddle point problems. Appl Math E-Notes. 2018;18:106–15.
59. Wang N-N, Li J-C. A class of new extended shift-splitting preconditioners for saddle point problems. J Comput Appl Math. 2019;357:123–45.
60. Cao Y. A general class of shift splitting preconditioners for non-Hermitian saddle point problems with applications to time harmonic eddy current models. Comput Math Appl. 2019;77:1124–43.
61. Cao Y, Ren Z-R. A local positive-(semi) definite shift-splitting preconditioner for non-Hermitian saddle point problems with applications to time-harmonic eddy current models. East Asian J Appl Math. 2020;10(1):135–57.
62. Elman HC, Ramage A, Silvester DJ. Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow. ACM Trans Math Softw. 2007;33(2):14-es.

How to cite this article: Vakili S, Ebadi G, Vuik C. A parameterized extended shift-splitting preconditioner for nonsymmetric saddle point problems. Numer Linear Algebra Appl. 2022;e2478. https://doi.org/10.1002/nla.2478