BIT Numer Math (2016) 56:587–604
DOI 10.1007/s10543-015-0590-9

A variant of the deteriorated PSS preconditioner for nonsymmetric saddle point problems

Juli Zhang[1,2] · **Chuanqing Gu**[1]

Received: 7 April 2015 / Accepted: 2 November 2015 / Published online: 4 January 2016

Abstract For nonsymmetric saddle point problems, Pan et al. (Appl Math Comput 172:762–771, 2006) proposed a deteriorated positive-definite and skew-Hermitian splitting (DPSS) preconditioner. In this paper, a variant of the DPSS preconditioner is proposed to accelerate the convergence of the associated Krylov subspace methods. The new preconditioner is much closer to the coefficient matrix than the DPSS preconditioner. The spectral properties of the new preconditioned matrix are analyzed. Theorem which provides the dimension of the Krylov space for the preconditioned matrix is obtained. Numerical experiments of a model Navier–Stokes problem are presented to illustrate the effectiveness of the new preconditioner.

Keywords Saddle point problems · Preconditioning · Spectral properties · Krylov subspace method · Navier–Stokes equation

Mathematics Subject Classification 65F10 · 65N22

Communicated by Zhong-Zhi Bai.

The work is supported by National Natural Science Foundation (11371243), Innovation major project of Shanghai Municipal Education Commission (13ZZ068), Key Disciplines of Shanghai Municipality (S30104), and by Foundation of Zhejiang Educational Committee (Y201431769).

✉ Chuanqing Gu
cqgu@shu.edu.cn

Juli Zhang
xzhzhangjuli@163.com

[1] Department of Mathematics, Shanghai University, Shanghai 200444, People's Republic of China

[2] Department of Mathematics, Zhejiang A&F University, Zhejiang 311300, People's Republic of China

1 Introduction

Consider the solution of the large and sparse saddle point linear system

$$\begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ -g \end{bmatrix}, \quad \text{or} \quad \mathscr{A}u = \mathbf{b}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsymmetric and positive definite (i.e., its symmetric part is positive definite), $B \in \mathbb{R}^{m\times n}$ has full row rank, B^T is the transpose of the matrix B, $x, f \in \mathbb{R}^n$, $y, g \in \mathbb{R}^m$ and $m \le n$. Under these conditions, $\mathscr{A}$ is nonsingular so that the solution of (1.1) exists and is unique. The system (1.1) is important and arises in a variety of scientific and engineering applications, such as computational fluid dynamics, constrained optimization, mixed or hybrid finite element approximations of second-order elliptic problems, and so on. See [2,15] and the references therein for a detailed discussion.

Since the coefficient matrix of (1.1) is large and sparse, iterative methods for solving (1.1) are more attractive than direct methods in terms of storage requirements and computing time. Recently, many effective iterative methods have been proposed to solve the saddle point problem (1.1), which include, to name just a few of them, Uzawa method [1,15], SOR-like and GSOR iteration methods [4,11,12,24], HSS (Hermitian and skew-Hermitian splitting) based methods [5,8–10]. See also [2] and [15] for a comprehensive survey and detailed study. However, some iterative methods, such as Krylov subspace methods, often suffer from slow convergence or even stagnation in some special engineering problems. In order to accelerate the convergence of the associated Krylov subspace method, the preconditioning technique is often used [13, 30,32].

In the past few years, in light of the special structure of problem (1.1), numerous preconditioners have been proposed. These include, block diagonal and block triangular preconditioners [2,3,7,18,25,27,29] (which are based on the matrix splitting or matrix factorization of the coefficient matrix $\mathscr{A}$), constraint preconditioners [14,21,22,26,29] (which are seldom used when the coefficient matrix $\mathscr{A}$ is nonsymmetric), HSS-based preconditioners [6,8,10,17,33], dimensional splitting (DS) preconditioners [16,19] (which have difficulties dealing with low-viscosity problems on stretched grids), and so on.

In this paper, for nonsymmetric saddle point problem (1.1), a new preconditioner based on the DPSS preconditioner in [28] will be proposed, which is referred to as *a variant of the deteriorated positive-definite and skew-Hermitian splitting* (VDPSS) preconditioner. After a brief introduction of the DPSS preconditioner, we present the new preconditioner in Sect. 2. The spectral properties of the preconditioned matrix are derived in Sect. 3. In that section, we also investigate the impact upon the convergence of the corresponding Krylov subspace method. In Sect. 4, numerical experiments are presented to illustrate the effectiveness of our preconditioner, including comparisons with other preconditioners. Finally, this paper is ended with some conclusions in Sect. 5.

For convenience, we briefly explain some of the terminologies used in this paper. $\mathbb{R}^l$ means the usual l-dimensional Euclidean space. For $P \in \mathbb{R}^{l\times l}$, P^T and P^{-1} indicate

the transpose and the inverse of the matrix P, respectively. $\lambda(P)$ denotes the eigenvalue of P. u^T, u^* and $\| u \|_2$ are the transpose, conjugate transpose and the 2-norm of a vector u. I_n denotes the identity matrix of size n.

2 A variant of the DPSS preconditioner

2.1 The DPSS preconditioner

For nonsymmetric saddle point problem (1.1), Pan, Ng and Bai in [28] proposed the DPSS preconditioner, which is induced by a deteriorated positive-definite and skew-Hermitian splitting (DPSS) iterative method [3]. Here we give a brief introduction to the DPSS preconditioner, and readers can consult [28] for more details. In [28], for the special property of the (1,1)-block matrix A, the authors split the coefficient matrix $\mathscr{A}$ into

$$\mathscr{A} = \mathscr{M} + \mathscr{N},$$

where

$$\mathscr{M} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathscr{N} = \begin{bmatrix} 0 & B^T \\ -B & 0 \end{bmatrix}.$$

Then we have two splittings of $\mathscr{A}$, i.e.,

$$\mathscr{A} = (\alpha I + \mathscr{M}) - (\alpha I - \mathscr{N}) = (\alpha I + \mathscr{N}) - (\alpha I - \mathscr{M}),$$

where I is the identity matrix and α is a given positive parameter. Since $\mathscr{M}$ is positive semi-definite and $\mathscr{N}$ is skew-Hermitian, $\alpha I + \mathscr{M}$ and $\alpha I + \mathscr{N}$ are both nonsingular. Motivated by the classical ADI iteration technique, the splitting iteration is proposed

$$\begin{cases} (\alpha I + \mathscr{M})u^{k+\frac{1}{2}} = (\alpha I - \mathscr{N})u^k + \mathbf{b}, \\ (\alpha I + \mathscr{N})u^{k+1} = (\alpha I - \mathscr{M})u^{k+\frac{1}{2}} + \mathbf{b}. \end{cases} \qquad k = 0, 1, 2 \ldots.$$

Thus, the deteriorated positive-definite and skew-Hermitian splitting (DPSS) iteration can be obtained

$$\begin{aligned} u^{k+1} = {} & (\alpha I + \mathscr{N})^{-1}(\alpha I - \mathscr{M})(\alpha I + \mathscr{M})^{-1}(\alpha I - \mathscr{N})u^k \\ & + 2\alpha(\alpha I + \mathscr{N})^{-1}(\alpha I + \mathscr{M})^{-1}\mathbf{b}. \end{aligned} \tag{2.1}$$

It is easy to observe that (2.1) can also be induced by the splitting $\mathscr{A} = \mathscr{P}_{\text{DPSS}} - \mathscr{Q}_{\text{DPSS}}$ (with $\mathscr{P}_{\text{DPSS}}$ nonsingular), which yields

$$u^{k+1} = \mathscr{P}_{\text{DPSS}}^{-1}\mathscr{Q}_{\text{DPSS}}u^k + \mathscr{P}_{\text{DPSS}}^{-1}\mathbf{b}, \tag{2.2}$$

where

$$\mathscr{P}_{\text{DPSS}} = \frac{1}{2\alpha}(\alpha I + \mathscr{M})(\alpha I + \mathscr{N}), \ \mathscr{Q}_{\text{DPSS}} = \frac{1}{2\alpha}(\alpha I - \mathscr{M})(\alpha I - \mathscr{N}).$$

Then the DPSS preconditioner $\mathscr{P}_{\mathrm{DPSS}}$ can be obtained. In [20], Cao et al. have proved that the DPSS iterative method (2.1) is convergent unconditionally. For the DPSS preconditioner, eigenvalue distribution of the preconditioned matrix $\mathscr{P}_{\mathrm{DPSS}}^{-1}\mathscr{A}$ is discussed in [28]. It is demonstrated in [28] that the eigenvalues of the preconditioned matrix gather into two cluster, i.e., (0, 0) and (2, 0), when α tends to 0_+.

2.2 A variant of the DPSS preconditioner

Since the factor $1/2$ has no effect on the preconditioned system, we can take any other constant instead of it. In this paper, we omit it for convenience. Then the DPSS preconditioner can be written as

$$\begin{aligned}\mathscr{P}_{\mathrm{DPSS}} &= \frac{1}{\alpha}(\alpha I + \mathscr{M})(\alpha I + \mathscr{N}) \\ &= \frac{1}{\alpha}\begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix}.\end{aligned} \tag{2.3}$$

From (1.1) and (2.3), we get the difference between the preconditioner $\mathscr{P}_{\mathrm{DPSS}}$ and the coefficient matrix $\mathscr{A}$

$$\begin{aligned}\mathscr{P}_{\mathrm{DPSS}} - \mathscr{A} &= \begin{bmatrix} \alpha I + A & \left(I + \frac{1}{\alpha}A\right)B^T \\ -B & \alpha I \end{bmatrix} - \begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \\ &= \begin{bmatrix} \alpha I & \frac{1}{\alpha}AB^T \\ 0 & \alpha I \end{bmatrix}.\end{aligned} \tag{2.4}$$

It shows that when α tends to 0_+, the weight of the two diagonal blocks in the matrix $\mathscr{P}_{\mathrm{DPSS}} - \mathscr{A}$ also approaches 0, while the weight of the nonzero off-diagonal block approaches $+\infty$. Hence, the choice of α needs to be balanced.

A general criterion for an efficient preconditioner is that it should be as close as possible to the coefficient matrix $\mathscr{A}$, such that the preconditioned matrix will have a clustered spectrum (away from 0) [15]. Therefore, by replacing the shift term αI in the (1,1) block of the first matrix in (2.3) with zero matrix, we get a variant of the DPSS preconditioner $\mathscr{P}_{\mathrm{VDPSS}}$ as follows

$$\begin{aligned}\mathscr{P}_{\mathrm{VDPSS}} &= \frac{1}{\alpha}\begin{bmatrix} A & 0 \\ 0 & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix} \\ &= \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & \frac{1}{\alpha}B^T \\ -B & \alpha I \end{bmatrix} \\ &= \begin{bmatrix} A & \frac{1}{\alpha}AB^T \\ -B & \alpha I \end{bmatrix}.\end{aligned} \tag{2.5}$$

The difference between $\mathscr{P}_{\text{VDPSS}}$ and $\mathscr{A}$ is

$$\mathscr{R} = \mathscr{P}_{\text{VDPSS}} - \mathscr{A} = \begin{bmatrix} 0 & \frac{1}{\alpha}AB^T - B^T \\ 0 & \alpha I \end{bmatrix}. \tag{2.6}$$

Although the (1,2)-block matrix is different from that in (2.4), the (1,1)-block matrix in (2.6) now vanishes, which indicates that $\mathscr{P}_{\text{VDPSS}}$ gives a better approximation to the coefficient matrix $\mathscr{A}$ for the same α. Therefore, $\mathscr{P}_{\text{VDPSS}}$ is expected to be a better preconditioner than $\mathscr{P}_{\text{DPSS}}$. Furthermore, the structure of (2.6) facilitates the analysis of the eigenvalue distribution of the preconditioned matrix; see the discussion in the next section.

In fact, the VDPSS preconditioner $\mathscr{P}_{\text{VDPSS}}$ can also be obtained by the following splitting of the coefficient matrix $\mathscr{A}$

$$\mathscr{A} = \mathscr{P}_{\text{VDPSS}} - \mathscr{R} = \begin{bmatrix} A & \frac{1}{\alpha}AB^T \\ -B & \alpha I \end{bmatrix} - \begin{bmatrix} 0 & \frac{1}{\alpha}AB^T - B^T \\ 0 & \alpha I \end{bmatrix},$$

which recasts in a fixed-point iteration, i.e., the VDPSS iteration

$$u^{k+1} = \Gamma u^k + \Theta, \tag{2.7}$$

with $u^0 = ((x^0)^T, (y^0)^T)^T$ being an initial vector, and $\Gamma = \mathscr{P}_{\text{VDPSS}}^{-1}\mathscr{R}$, $\Theta = \mathscr{P}_{\text{VDPSS}}^{-1}\mathbf{b}$. But it should also come to our attention that the VDPSS preconditioner $\mathscr{P}_{\text{VDPSS}}$ no longer relates to an alternating iteration method.

3 Eigenvalue analysis of the preconditioned matrix

In this section, we examine the spectral properties of the preconditioned matrix $\mathscr{P}_{\text{VDPSS}}^{-1}\mathscr{A}$ whose eigenvalue distribution influences the convergence of an iterative method. In particular, it is desirable that the number of distinct eigenvalues, or at least the number of clusters, is small, then the rate of convergence will be rapid. To be more precise, if there are only a few distinct eigenvalues, then the iterative method will terminate within a small number of steps.

We first give a lemma about the explicit expression of $\mathscr{P}_{\text{VDPSS}}^{-1}$.

Lemma 3.1 *Let*

$$\mathscr{P}_1 = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}, \quad \mathscr{P}_2 = \begin{bmatrix} I & \frac{1}{\alpha}B^T \\ -B & \alpha I \end{bmatrix}.$$

Here $\mathscr{P}_2$ has the block-triangular factorization

$$\mathscr{P}_2 = \begin{bmatrix} I & 0 \\ -B & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \tilde{A} \end{bmatrix} \begin{bmatrix} I & \frac{1}{\alpha}B^T \\ 0 & I \end{bmatrix}, \tag{3.1}$$

with $\tilde{A} = \alpha I + \frac{1}{\alpha} B B^T$. *Then we have*

$$\mathscr{P}_{VDPSS}^{-1} = \mathscr{P}_2^{-1} \mathscr{P}_1^{-1} = \begin{bmatrix} A^{-1} - \frac{1}{\alpha} B^T \tilde{A}^{-1} B A^{-1} & -\frac{1}{\alpha} B^T \tilde{A}^{-1} \\ \tilde{A}^{-1} B A^{-1} & \tilde{A}^{-1} \end{bmatrix}. \tag{3.2}$$

Theorem 3.1 *Let the VDPSS preconditioner be defined in (2.5), then the preconditioned matrix* $\mathscr{P}_{VDPSS}^{-1}\mathscr{A}$ *has eigenvalue* 1 *of algebraic multiplicity at least n. The real part of the remaining eigenvalues of the preconditioned matrix* $\mathscr{P}_{VDPSS}^{-1}\mathscr{A}$ *satisfies*

$$\frac{\alpha\sigma_m^2 \lambda_{\min}(\tilde{H})}{\alpha^2 + \sigma_m^2} \leq Re(\lambda) \leq \frac{\alpha\sigma_1^2 \lambda_{\max}(\tilde{H})}{\alpha^2 + \sigma_1^2},$$

where σ_m *and* σ_1 *are the smallest and largest singular values of matrix* B, $\tilde{H} = (A^{-1} + (A^{-1})^T)/2$ *is the symmetric part of matrix* A^{-1}.

Proof From Lemma 3.1 we have

$$\begin{aligned}
\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} &= I - \mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{R} \\
&= I - \begin{bmatrix} A^{-1} - \frac{1}{\alpha} B^T \tilde{A}^{-1} B A^{-1} & -\frac{1}{\alpha} B^T \tilde{A}^{-1} \\ \tilde{A}^{-1} B A^{-1} & \tilde{A}^{-1} \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{\alpha} A B^T - B^T \\ 0 & \alpha I \end{bmatrix} \\
&= I - \begin{bmatrix} 0 & \frac{1}{\alpha} B^T - A^{-1} B^T - \frac{1}{\alpha^2} B^T \tilde{A}^{-1} B B^T + \frac{1}{\alpha} B^T \tilde{A}^{-1} B A^{-1} B^T - B^T \tilde{A}^{-1} \\ 0 & \frac{1}{\alpha} \tilde{A}^{-1} B B^T - \tilde{A}^{-1} B A^{-1} B^T + \alpha \tilde{A}^{-1} \end{bmatrix} \\
&= \begin{bmatrix} I_n & S_1 \\ 0 & S_2 \end{bmatrix},
\end{aligned} \tag{3.3}$$

where $S_1 = -(\frac{1}{\alpha} B^T - A^{-1} B^T - \frac{1}{\alpha^2} B^T \tilde{A}^{-1} B B^T + \frac{1}{\alpha} B^T \tilde{A}^{-1} B A^{-1} B^T - B^T \tilde{A}^{-1})$, $S_2 = \tilde{A}^{-1} B A^{-1} B^T$.

Therefore, the preconditioned matrix $\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}$ has eigenvalue 1 of algebraic multiplicity at least n. The remaining non-unit eigenvalues of $\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}$ are the solution of the eigenvalue problem

$$\tilde{A}^{-1} B A^{-1} B^T u = \lambda u. \tag{3.4}$$

Since $\tilde{A} = \alpha I + \frac{1}{\alpha} B B^T$, (3.4) is equivalent to

$$B A^{-1} B^T u = \lambda \left(\alpha I + \frac{1}{\alpha} B B^T \right) u. \tag{3.5}$$

It is obvious that $u \neq 0$. Without loss of generality, we assume $\| u \|_2 = 1$. Multiplying the equation (3.5) by u^* yields

$$u^* B A^{-1} B^T u = \lambda \left(\alpha + \frac{1}{\alpha} u^* B B^T u \right). \tag{3.6}$$

Let $v = B^T u$, then (3.6) reduces to

$$v^* A^{-1} v = \lambda \left(\alpha + \frac{1}{\alpha} v^* v \right). \tag{3.7}$$

The conjugate transpose of (3.7) is

$$v^* (A^{-1})^T v = \bar{\lambda} \left(\alpha + \frac{1}{\alpha} v^* v \right). \tag{3.8}$$

Since $\tilde{H} = (A^{-1} + (A^{-1})^T)/2$ is the symmetric part of matrix A^{-1}, from (3.7) and (3.8), it is easy to obtain

$$\frac{v^* \tilde{H} v}{v^* v} = Re(\lambda) \left(\frac{\alpha}{v^* v} + \frac{1}{\alpha} \right).$$

According to the Courant-Fisher theorem,

$$\lambda_{\min}(\tilde{H}) \leq \frac{v^* \tilde{H} v}{v^* v} \leq \lambda_{\max}(\tilde{H}),$$
$$\sigma_m^2 \leq v^* v = u^* B B^T u \leq \sigma_1^2,$$

where, σ_m and σ_1 are the smallest and largest singular values of matrix B, respectively.

Then, we have

$$\frac{\alpha \sigma_m^2 \lambda_{\min}(\tilde{H})}{\alpha^2 + \sigma_m^2} \leq Re(\lambda) \leq \frac{\alpha \sigma_1^2 \lambda_{\max}(\tilde{H})}{\alpha^2 + \sigma_1^2}. \tag{3.9}$$

Thus, the proof of the theorem is completed. □

Remark 3.1 From (3.7), the non-unit eigenvalues of the preconditioned matrix $\mathscr{P}_{\text{VDPSS}}^{-1} \mathscr{A}$ satisfy

$$\lambda = \frac{\alpha v^* A^{-1} v}{\alpha^2 + v^* v}. \tag{3.10}$$

Taking $\alpha \to 0_+$ and $\alpha \to +\infty$, we get the non-unit eigenvalues $\lambda_i \to 0$, which will be confirmed by the figures in the next section.

It has been mentioned above that the idea of preconditioning attempts to improve on the spectral properties, such that the total number of iterations required to solve the system to within some tolerance will be indeed decreased. Among the iterative methods currently available, Krylov subspace methods are the most important for solving the underlying system. The iterative method with an optimal property, such as GMRES method, will terminate when the degree of the minimal polynomial is attained [31]. In particular, the degree of the minimal polynomial is equal to the dimension of the corresponding Krylov subspace [30]. Next theorem provides some analysis to the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}, b)$.

Theorem 3.2 *Let the VDPSS preconditioner be defined in (2.5), then the dimension of the Krylov subspace* $\mathscr{K}(\mathscr{P}_{VDPSS}^{-1}\mathscr{A}, b)$ *is at most* $m+1$. *Specially, once the matrix* $S_2 = \tilde{A}^{-1}BA^{-1}B^T$ *has* k $(1 \leq k \leq m)$ *distinct eigenvalues* μ_i $(1 \leq i \leq k)$, *of respective multiplicity* θ_i, *where* $\sum_{i=1}^{k}\theta_i = m$, *the dimension of the Krylov subspace* $\mathscr{K}(\mathscr{P}_{VDPSS}^{-1}\mathscr{A}, b)$ *is at most* $k+1$.

Proof According to the form of $\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}$ in (3.3) and the eigenvalue distribution described in Theorem 3.1, it is evident that the characteristic polynomial of the preconditioned matrix $\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}$ is

$$(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} - I)^n \prod_{i=1}^{m}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} - \lambda_i I).$$

Expanding the polynomial $(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} - I)\prod_{i=1}^{m}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} - \lambda_i I)$ of degree $m+1$, we have

$$(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} - I)\prod_{i=1}^{m}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A} - \lambda_i I) = \begin{bmatrix} 0 & S_1\prod_{i=1}^{m}(S_2 - \lambda_i I) \\ 0 & (S_2 - I)\prod_{i=1}^{m}(S_2 - \lambda_i I) \end{bmatrix}.$$

Since $\lambda_i^{'}$s are the eigenvalues of matrix S_2, then

$$\prod_{i=1}^{m}(S_2 - \lambda_i I) = 0.$$

Therefore, the degree of the minimal polynomial of $\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}$ is at most $m+1$. Consequently, the dimension of the corresponding Krylov subspace $\mathscr{K}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}, b)$ is at most $m+1$.

In the case when the matrix $S_2 = \tilde{A}^{-1}BA^{-1}B^T$ has k $(1 \leq k \leq m)$ distinct eigenvalues μ_i of respective multiplicity θ_i. We write the characteristic polynomial of

the preconditioned matrix $\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}$ as

$$(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}-I)^{n-1}\left[\prod_{i=1}^{k}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}-\mu_i I)^{\theta_i-1}\right]$$
$$(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}-I)\left[\prod_{i=1}^{k}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}-\mu_i I)\right].$$

Let $\Phi=(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}-I)\left[\prod_{i=1}^{k}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A}-\mu_i I)\right]$, then

$$\Phi=\begin{bmatrix}0 & S_1\prod_{i=1}^{k}(S_2-\mu_i I)\\ 0 & (S_2-I)\prod_{i=1}^{k}(S_2-\mu_i I)\end{bmatrix}.$$

Since $\prod_{i=1}^{k}(S_2-\mu_i I)=0$, it is obvious that Φ is a zero matrix. Therefore, the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{\mathrm{VDPSS}}^{-1}\mathscr{A},b)$ is at most $k+1$. Thus, the proof of the theorem is completed. □

Remark 3.2 Theorem 3.2 indicates that if a Krylov subspace method (such as GMRES) preconditioned by the VDPSS preconditioner is used for solving (1.1), then it will terminate in at most $m+1$ steps within some tolerance. Even in some special case, it will terminate in at most $k+1$ ($1\le k\le m$) iterations. This theorem reveals the excellent acceleration effect of the VDPSS preconditioner, which will also be confirmed in the section of numerical experiments.

Before the end of this section, we shall also analyze some implementation aspects about the preconditioner $\mathscr{P}_{\mathrm{VDPSS}}$. At each step of the VDPSS iteration or applying the VDPSS preconditioner $\mathscr{P}_{\mathrm{VDPSS}}$ within a Krylov subspace method, a linear system with $\mathscr{P}_{\mathrm{VDPSS}}$ as the coefficient matrix needs to be solved. That is to say, a linear system of the form

$$\begin{bmatrix}A & \frac{1}{\alpha}AB^T\\ -B & \alpha I\end{bmatrix}z=r$$

needs to be solved for a given vector r at each step, where $z=[z_1^T,z_2^T]^T$, $r=[r_1^T,r_2^T]^T$, $z_1,r_1\in R^n$, $z_2,r_2\in R^m$. Based on the matrix factorization of $\mathscr{P}_{\mathrm{VDPSS}}$ in Lemma 3.1, we have

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} I & -\frac{1}{\alpha}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \tilde{A}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ B & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \tag{3.11}$$

Hence, the following algorithmic version of the VDPSS iterative method can be derived.

Algorithm 3.1 For a given $r = [r_1^T, r_2^T]^T$, we can compute the vector $z = [z_1^T, z_2^T]^T$ by (3.11) from the following steps:

(i) solve $Ap_1 = r_1$;
(ii) solve $(\alpha I + \frac{1}{\alpha}BB^T)z_2 = Bp_1 + r_1$;
(iii) solve $z_1 = p_1 - \frac{1}{\alpha}B^T z_2$.

Remark 3.3 From Algorithm 3.1, it is required to solve two sub-linear systems with coefficient matrices A and $\tilde{A} = \alpha I + \frac{1}{\alpha}BB^T$ at each iteration. Obviously, according to the assumption given in Sect. 1, we know that the matrix A is positive definite, and $\tilde{A} = \alpha I + \frac{1}{\alpha}BB^T$ is symmetric positive definite. Therefore, in inexact manner, we can employ the GMRES method and the conjugate gradient (CG) method to solve the two sub-linear systems, respectively. In actual implementations, the inexact solvers can be used to reduce the cost of each iteration, but they will also lead to somewhat slower convergence. Thus we can also solve the two sub-linear systems exactly. The system with the coefficient matrix A can be solved with the sparse LU factorization, and the system with the coefficient matrix $\tilde{A} = \alpha I + \frac{1}{\alpha}BB^T$ can be solved with the sparse Cholesky factorization.

Remark 3.4 The RDPSS preconditioner proposed in [20] is similar to the VDPSS preconditioner, just different in the (2,2) block. In [20], the shift term αI in the (2,2) block of the RDPSS preconditioner also vanishes. Consequently, two sub-linear systems with coefficient matrices A and $\frac{1}{\alpha}BB^T$ need to be solved at each iteration when applying the RDPSS preconditioner within a Krylov subspace method. It is easy to observe that the condition number of $\alpha I + \frac{1}{\alpha}BB^T$ should be better than $\frac{1}{\alpha}BB^T$ when choosing large value of α, which will bring about substantial improvement on the number of iterations. However, it does not mean that we should use some unduly large α when applying the VDPSS preconditioner. In fact, experiments indicate that $\alpha = 100$ is appropriate, which is also the reason why we choose $\alpha = 100$ in Sect. 4. This attractive merit of the VDPSS preconditioner will be stressed by the numerical results in the next section.

4 Numerical experiments

In this section, we carry out some numerical experiments to illustrate the effectiveness of the VDPSS preconditioner for the nonsymmetric saddle point problem (1.1). The DPSS preconditioner [28] and the RDPSS preconditioner [20] are adopted to highlight the proposed preconditioner in terms of eigenvalue distribution, iteration and CPU time. Unless otherwise specified, we use left preconditioning with the restarted GMRES method [31] which has restarting frequency 30, i.e., GMRES(30). The zero

vector is adopted as the initial vector. The iteration stops when $\| r^k \|_2 / \| r^0 \|_2 < 10^{-6}$ or the prescribed maximum number of restarts $k_{\max} = 1000$ is exceeded, where r^k is the residual at the kth iteration. The main criteria used for comparison are the total number of iteration steps and CPU time which are abbreviated to Iter and CPU. All experiments are run on a PC using MATLAB 2008 under Windows 7 operating system.

Here we consider the Oseen problem which is obtained from the linearization of the steady-state Navier–Stokes equation with suitable boundary condition on $\partial\Omega$:

$$\begin{cases} -\nu\Delta\mathbf{u} + (\omega \cdot \nabla)\mathbf{u} + \nabla\mathbf{p} = \mathbf{f}, \quad \text{in } \Omega, \\ \text{div}\mathbf{u} = 0, \quad \text{in } \Omega, \\ \mathbf{u} = \mathbf{g}, \quad \text{on } \partial\Omega, \end{cases} \tag{4.1}$$

where $\nu > 0$ represents the kinematic viscosity, div is the divergence, Δ is the vector Laplace operator, ∇ is the gradient, u, p stand for the velocity and pressure of the fluid, respectively. We mainly focus on the "regularized" two-dimensional lid-driven cavity problems discretized by Q2-P1 finite element on both uniform and stretched grids. The IFISS software package [23] developed by Elman, Ramage and Silvester is used to generate the test problems. In each case, we consider three viscosity values, that is, $\nu = 1$, $\nu = 0.1$, $\nu = 0.01$ to generate linear systems corresponding to 16×16, 32×32, 64×64 and 128×128 meshes. The resulting linear systems have the form

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

where $A \in \mathbb{R}^{n\times n}$ corresponds to the discretization of the convection-diffusion term, the rectangular matrix $B^T \in \mathbb{R}^{n\times m}$ represents the discrete gradient operator, and $B \in \mathbb{R}^{m\times n}$ represents the divergence operator.

It should be emphasized that the sub-linear systems arising from the application of the preconditioners are solved by direct methods. In Matlab, this corresponds to computing the Cholesky or LU factorization in combination with AMD or column AMD reordering.

4.1 Discretizing with Q2-P1 finite element on uniform grids

We first consider discretizing the Oseen problem with Q2-P1 finite element on uniform grids. Although the rate of convergence of nonsymmetric Krylov iterations preconditioned by a parameter-dependent preconditioner depends on the particular choice of the parameter, the analysis to the optimal parameter seems to be quite a difficult problem. In our numerical experiments, the parameters are determined experimentally. Since in Remark 3.1, we have analyzed theoretically that the non-unit eigenvalues will tend to 0 when $\alpha \to 0_+$ and $\alpha \to +\infty$. In order to verify this property, but also to justify Remark 3.4, $\alpha = 0.001$, $\alpha = 1$, $\alpha = 10$ and $\alpha = 100$ are adopted as samples to investigate the trait of the new preconditioner $\mathscr{P}_{\text{VDPSS}}$. Here, $\alpha = 1$ are also considered in [20].

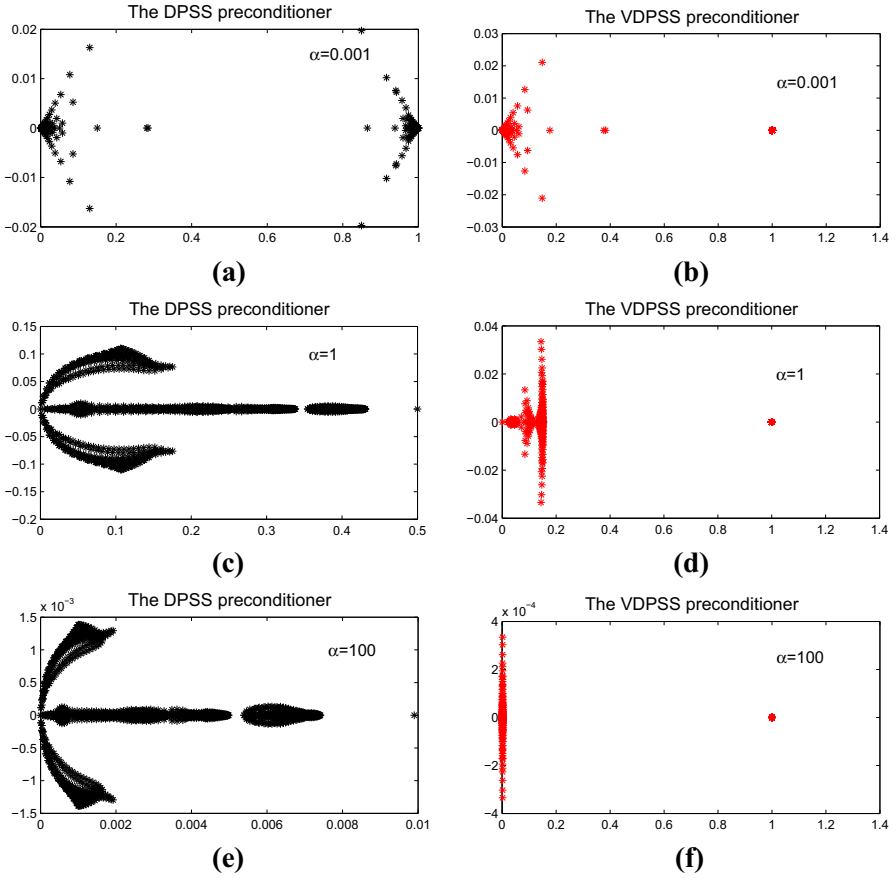

Fig. 1 Eigenvalue distribution of the preconditioned matrix (Q2P1FEM, $\nu = 0.1$, 32×32 uniform grid)

Figure 1 demonstrates the eigenvalue distribution of the new preconditioned matrix $\mathscr{P}_{\text{VDPSS}}^{-1}\mathscr{A}$ and the DPSS preconditioned matrix $\mathscr{P}_{\text{DPSS}}^{-1}\mathscr{A}$ for $\alpha = 0.001$, $\alpha = 1$ and $\alpha = 100$ on 32×32 uniform grids with $\nu = 0.1$. Numerical results for different choices of ν and α on the uniform grids are tabulated in Tables 1, 2, 3. We compare the three preconditioners in terms of the iteration steps and CPU time. It should be noted that in these tables, the symbol "–" means that the corresponding algorithm will not be convergent within the prescribed number of restarts $k_{\max} = 1000$.

From the figures and tables, we can get

(i) Figure 1 echoes Theorem 3.1 that the preconditioned matrix $\mathscr{P}_{\text{VDPSS}}^{-1}\mathscr{A}$ has at least n eigenvalues 1. In fact, there are 2178 (the dimension n) eigenvalues of value 1 in (b)(d)(f) of Fig. 1 as compared to none in the subplot (a)(c)(e) (using the DPSS preconditioner). From the figures, we find that the non-unit eigenvalues converge to 0 when $\alpha \to 0_+$ and $+\infty$. This phenomenon confirms the statement in Remark 3.1.

Table 1 Numerical results for Oseen equation with different uniform grids (Q2P1FEM, $\nu = 1$)

Grid	α	VDPSS		RDPSS		DPSS	
		Iter	CPU	Iter	CPU	Iter	CPU
16×16	0.001	21	0.0312	21	0.0312	48	0.1092
	1	22	0.0468	27	0.0936	132	0.1560
	10	20	0.0312	35	0.0624	360	0.0624
	100	11	0.0312	48	0.0936	168	0.2028
32×32	0.001	40	0.3744	40	0.3120	77	0.5928
	1	23	0.2496	57	0.3744	668	1.0608
	10	13	0.1092	82	0.4836	337	0.8268
	100	10	0.1716	129	0.3900	288	0.7488
64×64	0.001	69	1.2636	68	1.6536	157	1.9344
	1	14	1.1076	156	2.2308	351	4.0404
	10	11	0.9048	172	2.0592	560	4.9764
	100	8	1.1388	351	3.4632	488	3.9624
128×128	0.001	220	11.2477	142	9.7813	587	21.6373
	1	13	4.6644	267	13.1040	610	22.2769
	10	10	5.4288	458	19.5001	1035	35.8334
	100	6	4.1964	1732	58.5316	784	27.0662

Table 2 Numerical results for Oseen equation with different uniform grids (Q2P1FEM, $\nu = 0.1$)

Grid	α	VDPSS		RDPSS		DPSS	
		Iter	CPU	Iter	CPU	Iter	CPU
16×16	0.001	23	0.0312	23	0.0312	45	0.0936
	1	24	0.0468	35	0.0468	183	0.1716
	10	23	0.0468	48	0.0624	195	0.1248
	100	14	0.1092	73	0.0624	132	0.1248
32×32	0.001	42	0.2028	42	0.2496	79	0.4836
	1	27	0.2496	73	0.4680	342	0.7332
	10	15	0.1560	180	0.4992	338	1.4040
	100	12	0.1716	181	0.5460	241	0.5722
64×64	0.001	75	1.2012	75	1.3104	115	1.6068
	1	17	0.9984	167	1.9968	819	6.2244
	10	14	1.0452	420	3.6504	912	6.7548
	100	11	0.8580	1686	11.5441	574	4.6644
128×128	0.001	163	9.4381	147	9.0325	203	10.6237
	1	16	5.9748	387	17.1133	2358	77.2985
	10	13	4.6488	3073	101.3694	2262	73.4453
	100	9	4.6800	–	–	1151	39.8583

Table 3 Numerical results for Oseen equation with different uniform grids (Q2P1FEM, $\nu = 0.01$)

Grid	α	VDPSS		RDPSS		DPSS	
		Iter	CPU	Iter	CPU	Iter	CPU
16×16	0.001	53	0.0780	53	0.0780	96	0.2652
	1	78	0.0624	144	0.1092	705	0.2496
	10	69	0.0780	298	0.1404	786	0.3248
	100	64	0.0936	1392	0.4680	510	0.2496
32×32	0.001	81	0.3120	81	0.3120	97	0.4524
	1	148	0.2964	697	0.9516	1688	1.9500
	10	102	0.3900	3481	4.1964	1455	1.8096
	100	80	0.3276	–	–	814	0.9984
64×64	0.001	140	1.6380	140	1.6380	75	1.3416
	1	236	2.5276	2170	15.0229	3491	22.4017
	10	115	1.4352	1861	119.0444	2404	15.4129
	100	49	1.9344	–	–	1099	7.8313
128×128	0.001	701	25.1786	330	14.9605	141	8.4453
	1	119	9.0793	7795	258.5561	6468	241.1931
	10	64	6.2088	–	–	4074	128.9504
	100	48	6.0840	–	–	1271	46.8315

(ii) Tables 1, 2, 3 indicate that using the VDPSS preconditioner often leads to much better performance than using the DPSS and RDPSS preconditioners in terms of iteration counts and CPU time. Specially in the case of some large values of α, the VDPSS preconditioner has significant reduction in the iteration counts, while the other two preconditioners are difficult to implement efficiently, even the RDPSS preconditioner sometimes fails to converge. But it also comes to our attention that the new preconditioner is not competitive in the case of small α. Apart from that, the new preconditioner seems to be difficult in dealing with low viscosity, that is, the iteration counts increase with the decreasing of the kinematic viscosity.

4.2 Discretizing with Q2-P1 finite element on stretched grids

To further investigate the VDPSS preconditioner, we also consider discretizing the Oseen problem with Q2-P1 finite element on stretched grids. Similar to the uniform grids, we also test four values of the parameter α, that is, $\alpha = 0.001$, $\alpha = 1$, $\alpha = 10$ and $\alpha = 100$ with the kinematic viscosity $\nu = 0.01$, $\nu = 0.1$ and $\nu = 1$.

Figure 2 depicts the eigenvalue distribution of the preconditioned matrices $\mathscr{P}_{\text{DPSS}}^{-1}\mathscr{A}$ and $\mathscr{P}_{\text{VDPSS}}^{-1}\mathscr{A}$ for $\alpha = 0.001$ $\alpha = 1$ and $\alpha = 100$ on 32×32 stretched grids with $\nu = 0.1$. Tables 4, 5, 6 demonstrate the numerical results for $\nu = 1$, $\nu = 0.1$ and $\nu = 0.01$ on the stretched grids, where the default stretch factors are provided by IFISS.

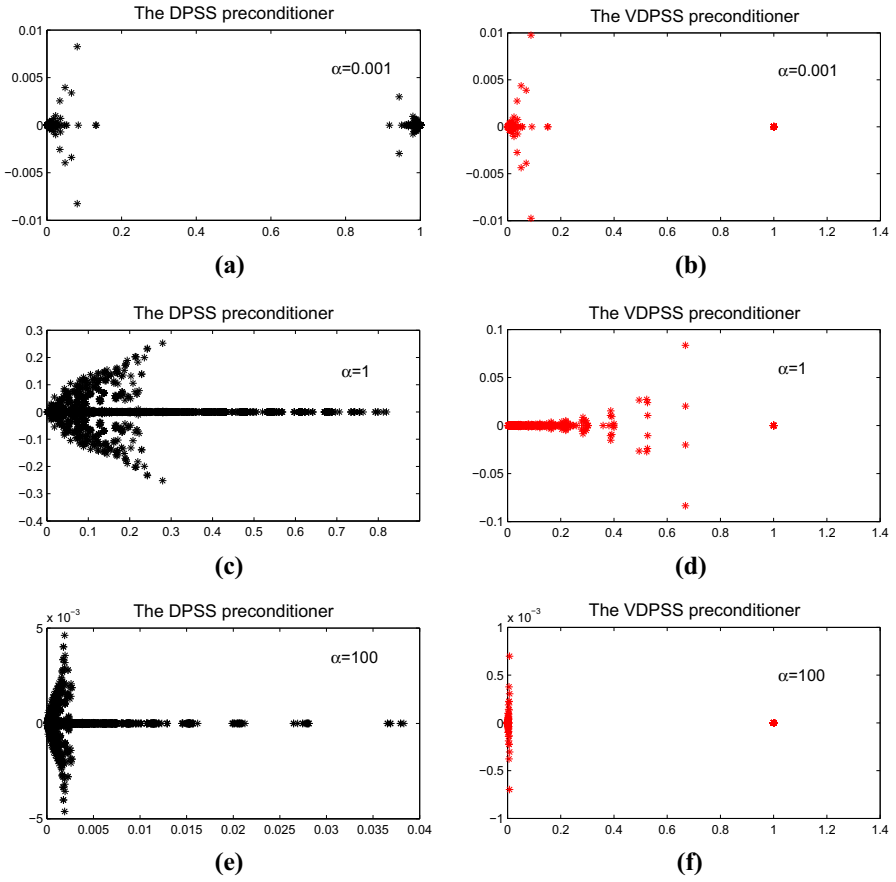

Fig. 2 Eigenvalue distribution of the preconditioned matrix (Q2P1FEM, $\nu = 0.1$, 32×32 stretched grid)

The conclusion obtained from Fig. 2 and Tables 4, 5, 6 is similar to that given in Sect. 4.1. The VDPSS preconditioner leads to much better performance than the DPSS and RDPSS preconditioners in the case of some large values of α. And the new preconditioner also has difficulty in dealing with low-viscosity problems on stretched grids.

5 Conclusion

In this paper, we present a variant of the deteriorated positive-definite and skew-Hermitian splitting (VDPSS) preconditioner for nonsymmetric saddle point problem (1.1). The proposed preconditioner is based on the deteriorated positive-definite and skew-Hermitian splitting (DPSS) preconditioner, but is much closer to the coefficient matrix. The eigenvalue distribution and the acceleration effect (as a preconditioner) on GMRES(30) for solving the Oseen problems have been investigated. Numerical experiments have shown that the new preconditioner is effective. However, how to

Table 4 Numerical results for Oseen equation with different stretched grids (Q2P1FEM, $\nu = 1$)

Grid	α	VDPSS		RDPSS		DPSS	
		Iter	CPU	Iter	CPU	Iter	CPU
16×16	0.001	15	0.0312	15	0.0312	25	0.0624
	1	25	0.0312	29	0.0312	460	0.2496
	10	34	0.2184	37	0.2312	421	0.6396
	100	28	0.0468	59	0.0936	429	0.2652
32×32	0.001	23	0.2340	23	0.2340	51	0.3120
	1	87	0.4212	72	0.2652	592	0.8736
	10	66	0.3744	89	0.4680	1942	2.5272
	100	35	0.2028	216	0.4368	1674	2.0748
64×64	0.001	64	1.4820	40	1.7472	115	1.8092
	1	136	1.5288	138	2.5534	1701	11.2477
	10	73	1.2636	209	2.6832	4810	30.4514
	100	23	0.9828	548	5.2260	3208	20.5609
128×128	0.001	60	8.1085	61	8.1909	120	11.4193
	1	150	10.7329	250	23.3377	4502	139.0125
	10	49	5.9434	507	35.9582	2897	95.8002
	100	5	5.0544	1435	85.3949	1378	47.6427

Table 5 Numerical results for Oseen equation with different stretched grids (Q2P1FEM, $\nu = 0.1$)

Grid	α	VDPSS		RDPSS		DPSS	
		Iter	CPU	Iter	CPU	Iter	CPU
16×16	0.001	21	0.0156	21	0.0156	38	0.0936
	1	39	0.0936	37	0.0936	329	0.2652
	10	87	0.0936	58	0.0312	676	0.3496
	100	37	0.0780	96	0.1248	188	0.1404
32×32	0.001	30	0.2496	30	0.2496	58	0.6240
	1	224	0.5928	89	0.4992	1636	2.1372
	10	95	0.3432	175	0.7020	1125	1.2948
	100	64	0.3432	478	0.7644	614	0.9204
64×64	0.001	100	1.6068	56	1.3728	152	2.2152
	1	208	2.2932	210	2.4648	2403	17.2693
	10	73	1.8408	209	2.5116	4810	29.7494
	100	73	1.4196	3640	25.9274	1623	12.2305
128×128	0.001	281	18.4865	90	7..8469	509	26.5826
	1	291	12.7921	527	23.1973	6487	200.3677
	10	153	8.7985	1646	60.9964	8047	252.3316
	100	52	5.7408	13,427	453.2921	2394	77.1893

Table 6 Numerical results for Oseen equation with different stretched grids (Q2P1FEM, $\nu = 0.01$)

Grid	α	VDPSS		RDPSS		DPSS	
		Iter	CPU	Iter	CPU	Iter	CPU
16×16	0.001	41	0.0312	41	0.0312	73	0.0780
	1	189	0.1716	96	0.0780	626	0.2652
	10	136	0.1248	330	0.1404	911	0.4056
	100	80	0.0936	297	0.2048	639	0.2340
32×32	0.001	62	0.3120	63	0.3120	81	0.4680
	1	874	1.6068	300	0.8424	1808	2.3244
	10	186	0.3432	1912	2.4808	2020	2.8556
	100	94	0.7644	24075	26.0366	1297	1.4976
64×64	0.001	208	2.4336	109	2.0124	123	1.8720
	1	447	3.5724	1255	8.5489	4221	26.6294
	10	411	3.3852	9433	56.6104	4106	25.6154
	100	272	2.4180	–	–	2177	14.0713
128×128	0.001	5335	229.4464	206	11.6689	1549	70.5281
	1	734	25.6934	3874	137.4152	7833	240.1635
	10	568	21.9181	–	–	8265	257.8385
	100	339	14.8513	–	–	3641	115.9711

deal with low-viscosity problems still requires further in-depth studies. Meanwhile, future work should also focus on the choice of the optimal parameter α.

References

1. Arrow, K.J., Hurwicz, L., Uzawa, H.: Studies in Linear and Non-Linear Programming. Stanford University Press, Stanford (1958)
2. Bai, Z.-Z.: Structured preconditioners for nonsingular matrices of block two-by-two structures. Math. Comput. **75**, 791–815 (2006)
3. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput. **26**, 844–863 (2005)
4. Bai, Z.-Z.: A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations. Adv. Comput. Math. **10**, 169–186 (1999)
5. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**, 603–626 (2003)
6. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**, 1–32 (2004)
7. Bai, Z.-Z., Ng, M.K.: On inexact preconditioners for nonsymmetric matrices. SIAM J. Sci. Comput. **26**, 1710–1724 (2005)
8. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**, 1–23 (2007)
9. Bai, Z.-Z.: Optimal parameters in the HSS-like methods for saddle-point problems. Numer. Linear Algebra Appl. **16**, 447–479 (2009)
10. Bai, Z.-Z., Golub, G.H., Li, C.-K.: Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. Math. Comput. **76**, 287–298 (2007)

11. Bai, Z.-Z., Parlett, B.N., Wang, Z.-Q.: On generalized successive overrelaxation methods for augmented linear systems. Numer. Math. **102**, 1–38 (2005)
12. Bai, Z.-Z., Wang, Z.-Q.: On parameterized inexact Uzawa methods for generalized saddle point problems. Linear Algebra Appl. **428**, 2900–2932 (2008)
13. Bai, Z.-Z.: Motivations and realizations of Krylov subspace methods for large sparse linear systems. J. Comput. Appl. Math. **283**, 71–78 (2015)
14. Bai, Z.-Z., Ng, M.K., Wang, Z.-Q.: Constraint preconditioners for symmetric indefinite matrices. SIAM J. Matrix Anal. Appl. **31**, 410–433 (2009)
15. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
16. Benzi, M., Guo, X.-P.: A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations. Appl. Numer. Math. **61**, 66–76 (2011)
17. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 20–41 (2004)
18. Cao, Z.-H.: Positive stable block triangular preconditioners for symmetric saddle point problems. Appl. Numer. Math. **57**, 899–910 (2007)
19. Cao, Y., Yao, L.-Q., Jiang, M.-Q.: A modified dimensional split preconditioner for generalized saddle point problems. J. Comput. Appl. Math. **250**, 70–82 (2013)
20. 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. **273**, 41–60 (2015)
21. Dollar, H.S., Wathen, A.J.: Approximate factorization constraint preconditioners for saddle-point matrics. SIAM J. Sci. Comput. **27**, 1555–1572 (2006)
22. Dollar, H.S.: Constraint-style preconditioners for regularized saddle-point problems. SIAM J. Matrix Anal. Appl. **29**, 672–684 (2007)
23. Elman, H.C., Ramage, A., Silvester, D.J.: IFISS: a Matlab toolbox for modelling incompressible flow. ACM Trans. Math. Software **33**, Article 14 (2007)
24. Golub, G.H., Wu, X., Yuan, J.-Y.: SOR-like methods for augmented systems. BIT **41**, 71–85 (2001)
25. Ipsen, I.C.F.: A note on preconditioning nonsysmmetric matrices. SIAM J. Sci. Comput. **23**, 1050–1051 (2001)
26. Keller, C., Gould, N.I.M., Wathen, A.J.: Constraint preconditioning for indefinite linear systems. SIAM J. Matrix Anal. Appl. **21**, 1300–1317 (2000)
27. Murphy, M.F., Golub, G.H., Wathen, A.J.: A note on preconditioning for indefinite linear systems. SIAM J. Sci. Comput. **21**, 1969–1972 (2000)
28. Pan, J.-Y., Ng, M.K., Bai, Z.-Z.: New preconditioners for saddle point problems. Appl. Math. Comput. **172**, 762–771 (2006)
29. Sturler, E.D., Liesen, J.: Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. SIAM J. Sci. Comput. **26**, 1598–1619 (2005)
30. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
31. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986)
32. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM J. Sci. Comput. **14**, 461–469 (1993)
33. Zhang, J.-L., Gu, C.-Q., Zhang, K.: A relaxed positive-definite and skew-Hermitian splitting preconditioner for saddle point problems. Appl. Math. Comput. **249**, 468–479 (2014)