# Accepted Manuscript


A block product preconditioner for saddle point problems

Li-Dan Liao, Guo-Feng Zhang, Mu-Zheng Zhu

PII: S0377-0427(18)30716-7
DOI: https://doi.org/10.1016/j.cam.2018.11.026
Reference: CAM 12034

To appear in: *Journal of Computational and Applied Mathematics*

Received date : 28 June 2016
Revised date : 3 July 2018

Please cite this article as: L.-D. Liao, G.-F. Zhang and M.-Z. Zhu, A block product preconditioner for saddle point problems, *Journal of Computational and Applied Mathematics* (2018), https://doi.org/10.1016/j.cam.2018.11.026

# A block product preconditioner for saddle point problems☆

Li-Dan Liao[a,b], Guo-Feng Zhang*[,a], Mu-Zheng Zhu[c]

[a]School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, P.R. China
[b]Department of Mathematics, Nanchang University, Nanchang 330031, P.R. China
[c]School of Mathematics and Statistics, Hexi University, ZhangYe 734000, P.R. China

**Abstract**

In this paper, a block product (BP) preconditioner is established for saddle point problems. Spectral properties of the BP preconditioned matrix are investigated. A strategy for practical choice of quasi-optimal parameter is given. Numerical results on saddle point linear systems arising from Stokes problems and weighted least square problems show that the proposed BP preconditioner is more economic to implement within Krylov subspace acceleration than some extensively studied preconditioners.

*Key words:* saddle point problems, preconditioner, spectral properties, optimal parameter, Krylov subspace method
*2000 MSC:* 65F10, 65F50, 65N22, 65F35

## 1. Introduction

Consider the solution to the following saddle point problems

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

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, $B \in \mathbb{R}^{m\times n}$ is of full row-rank, $D \in \mathbb{R}^{m\times m}$ is symmetric positive semi-definite, $x, f \in \mathbb{R}^n, y, g \in \mathbb{R}^m$, with $n \geq m$. The above assumptions guarantee the existence and uniqueness of the solution of linear system (1.1), also see [14] for detailed discussions. The linear system (1.1) arises in a wide variety of scientific computing and engineering applications. For example, computational fluid dynamics and mixed finite element approximation of elliptic PDEs, optimal control, weighted and equality constrained least squares estimation, structural analysis, electrical networks, inversion of geophysical data, computer graphics and so forth[1, 7–9, 12, 14, 20].

The accurate and efficient solution of linear system (1.1) has been an active research area in computational mathematics for many decades. Due to the large and sparse structure of coefficient matrix $\mathcal{A}$, there is a rapidly growing interest in iterative methods for solving linear system (1.1) in the past few years. So far, a large amount of work has been devoted to developing efficient iterative methods, including Uzawa-type methods [7, 9, 15, 21, 36], preconditioned Krylov subspace iteration methods [6, 14, 16, 17, 23], Hermitian and skew-Hermitian splitting (HSS) method as well as its accelerated variants [2–5, 8, 25], restrictively preconditioned conjugate gradient methods [10, 11, 35] and so on. In particular, Krylov subspace methods with appropriate preconditioners are considered to be more efficient in general. An important criterion for an efficient preconditioner is that it can be easily implemented, which further make the total computational cost to be optimal. In light of the special structure of

☆This work was supported by the National Natural Science Foundation of China(11771193, 11661033) and the Fundamental Research Funds for the Central Universities(lzujbky-2017-it56).
*Corresponding author.
*Email addresses:* liaold15@lzu.edu.com (Li-Dan Liao), gf_zhang@lzu.edu.cn (Guo-Feng Zhang)

equations (1.1), many efficient and robust preconditioners have been reported [1, 12, 16–18, 26, 28, 29, 31, 32]. For an overview of numerical methods and useful preconditioners for the saddle point problems, see [14].

It should be mentioned that the HSS method [3] is very useful for solving non-Hermitian positive definite systems. Benzi and Gloub proposed the following HSS preconditioner [12]

$$\mathscr{P}_{\rm HSS}=\frac{1}{2\alpha}\begin{bmatrix}\alpha I+A & 0\\ 0 & \alpha I+D\end{bmatrix}\begin{bmatrix}\alpha I & B^T\\ -B & \alpha I\end{bmatrix}$$

for the generalized saddle point problem (1.1), where $\alpha>0$ is a given constant, $I$ denotes the identity matrix. In order to further speed up the convergence rate of the HSS preconditioner, Bai et al. [18] proposed a SHSS preconditioner, which is described as

$$\mathscr{P}_{\rm SHSS}=\frac{1}{\alpha}\begin{bmatrix}A & 0\\ 0 & \alpha I\end{bmatrix}\begin{bmatrix}\alpha I & B^T\\ -B & D\end{bmatrix}. \tag{1.2}$$

The authors proved that the SHSS preconditioner $\mathscr{P}_{\rm SHSS}$ is a better approximation to the matrix $\mathscr{A}$, which is easier to implement than the preconditioner $\mathscr{P}_{\rm HSS}$ [18].

In this paper, based on a block-triangular product approximation to the coefficient matrix $\mathscr{A}$ in (1.1), we propose a block product (BP) preconditioner. This preconditioner also results in a BP splitting iteration method. The convergence properties of the BP splitting iteration are analyzed. A practical selection for the quasi-optimal parameters is given. Spectral properties of the BP preconditioned matrix and the finite-step termination properties of the preconditioned Krylov subspace method are described. Several numerical examples are performed to illustrate the effectiveness of the proposed preconditioners.

The paper is organized as follows. In Section 2, we present the BP preconditioner and its implementation for solving the generalized saddle point problems (1.1). In Section 3, the convergence analysis is given and the quasi-optimal parameter is discussed. In Section 4, the spectral properties of the BP preconditioned matrix are analyzed. In Section 5, numerical experiments are presented to show the effectiveness of the BP preconditioner and the feasibility of the strategy for parameter selection. Finally, some brief concluding remarks are given in Section 6.

## 2. Block Product preconditioner

In this section, we present a block product (BP) preconditioner for solving the linear system (1.1). According to (1.2), the SHSS preconditioner can be rewritten as

$$\mathscr{P}_{\rm SHSS}=\begin{bmatrix}A & 0\\ -B & I\end{bmatrix}\begin{bmatrix}I & \frac{1}{\alpha}B^T\\ 0 & D+\frac{1}{\alpha}BB^T\end{bmatrix}=\begin{bmatrix}A & \frac{1}{\alpha}AB^T\\ -B & D\end{bmatrix}. \tag{2.1}$$

As we know that, in many cases, $D$ usually owns some special properties, such as $D$ is a diagonal matrix or $D$ is a scalar matrix [30–33]. Since the matrix $\beta I+D$ can still maintain the special structure of $D$, while $D+\frac{1}{\alpha}BB^T$ may destroy its special structure, so that the computation of the solution on the matrix $D+\beta I$ will be more expensive than that of $D+\frac{1}{\alpha}BB^T$. With these in mind, we consider using $\beta I+D$ to replace $D+\frac{1}{\alpha}BB^T$ in (2.1), which results in the following block product (BP) preconditioner

$$\mathscr{P}_{\rm BP}(\alpha,\beta)=\begin{bmatrix}A & 0\\ -B & I\end{bmatrix}\begin{bmatrix}I & \frac{1}{\alpha}B^T\\ 0 & \beta I+D\end{bmatrix}=\begin{bmatrix}A & \frac{1}{\alpha}AB^T\\ -B & \beta I+D-\frac{1}{\alpha}BB^T\end{bmatrix}, \tag{2.2}$$

where $\alpha$ and $\beta$ are two positive parameters.

In fact, $\mathscr{P}_{\rm BP}(\alpha,\beta)$ can result in the following matrix splitting, called the BP splitting,

$$\mathscr{A}=\mathscr{P}_{\rm BP}(\alpha,\beta)-\mathscr{R}_{\rm BP}(\alpha,\beta) \tag{2.3}$$

with

$$\mathscr{R}_{\rm BP}(\alpha,\beta):=\mathscr{P}_{\rm BP}(\alpha,\beta)-\mathscr{A}=\begin{bmatrix}0 & \frac{1}{\alpha}AB^T-B^T\\ 0 & \beta I-\frac{1}{\alpha}BB^T\end{bmatrix}.$$

Based on the BP splitting (2.3), we can construct the BP splitting iteration method

$$\begin{bmatrix} x^{k+1} \\ y^{k+1} \end{bmatrix} = \mathscr{T}(\alpha,\beta)\begin{bmatrix} x^{k} \\ y^{k} \end{bmatrix} + \mathscr{P}_{\mathrm{BP}}(\alpha,\beta)^{-1}\begin{bmatrix} f \\ g \end{bmatrix}, \tag{2.4}$$

where

$$\mathscr{T}(\alpha,\beta) = \mathscr{P}_{\mathrm{BP}}(\alpha,\beta)^{-1}\mathscr{R}_{\mathrm{BP}}(\alpha,\beta) = \begin{bmatrix} A & \frac{1}{\alpha}AB^T \\ -B & \beta I + D - \frac{1}{\alpha}BB^T \end{bmatrix}^{-1}\begin{bmatrix} 0 & \frac{1}{\alpha}AB^T - A \\ 0 & \beta I - \frac{1}{\alpha}BB^T \end{bmatrix} \tag{2.5}$$

At each step in the application of the $\mathscr{P}_{\mathrm{BP}}(\alpha,\beta)$ preconditioner with a Krylov subspace method, it needs to solve a sequence of generalized residual equations $\mathscr{P}_{\mathrm{BP}}z = r$, where $r = (r_1^T, r_2^T)^T \in \mathbb{R}^{m+n}$ is a given residual vector, $z = (z_1^T, z_2^T)^T \in \mathbb{R}^{m+n}$ represents generalized residual vector, with $z_1, r_1 \in \mathbb{R}^n$ and $z_2, r_2 \in \mathbb{R}^m$.

It is easy to verify that

$$\begin{bmatrix} A & \frac{1}{\alpha}AB^T \\ -B & \beta I + D - \frac{1}{\alpha}BB^T \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ -B & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \beta I + D \end{bmatrix}\begin{bmatrix} I & \frac{1}{\alpha}B^T \\ 0 & I \end{bmatrix}. \tag{2.6}$$

Thus, according to (2.6), we can solve $\mathscr{P}_{\mathrm{BP}}z = r$ for the preconditioner $\mathscr{P}_{\mathrm{BP}}$ by the following algorithm.

**Algorithm 2.1.**

**(1)** *solve* $Au_1 = r_1$;

**(2)** *compute* $u_2 := r_2 + Bu_1$;

**(3)** *solve* $(\beta I + D)z_2 = u_2$;

**(4)** *compute* $z_1 := u_1 - \frac{1}{\alpha}B^T z_2$;

From Algorithm 2.1, we know that two linear sub-systems with coefficient matrices $A$ and $\beta I + D$ need to be solved at each iteration step. As discussed in many papers [2–4, 9, 18, 19, 27], when the matrix sizes are large, the preconditioned conjugate gradient (PCG) method can be applied to solve the afore-mentioned two linear sub-systems, as both $A$ and $\beta I + D$ are symmetric and positive definite.

In analogy with the BP preconditioner, the implementation of the SHSS [18] and HSS [12] preconditioners with a Krylov subspace method can be described as follows:

**Algorithm 2.2.** *We solve* $\mathscr{P}_{HSS}z = r$ *for the preconditioner* $\mathscr{P}_{HSS}$ *by the following steps:*

**(1)** *solve* $(\alpha I + A)u_1 = r_1$;

**(2)** *compute* $(\alpha I + D)u_2 := r_2 + Bu_1$;

**(3)** *solve* $(\frac{1}{\alpha}BB^T + \alpha I)z_2 = u_2$;

**(4)** *compute* $z_1 := \frac{1}{\alpha}(u_1 - B^T z_2)$.

**Algorithm 2.3.** *We solve* $\mathscr{P}_{SHSS}z = r$ *for the preconditioner* $\mathscr{P}_{SHSS}$ *by the following steps:*

**(1)** *solve* $Au_1 = r_1$;

**(2)** *compute* $u_2 := r_2 + Bu_1$;

**(3)** *solve* $(\frac{1}{\alpha}BB^T + \beta I)z_2 = u_2$;

**(4)** *compute* $z_1 := u_1 - \frac{1}{\alpha}B^T z_2$;

From the above algorithms, we see that the computational cost of applying the BP preconditioner to accelerate a Krylov subspace method for the linear system (1.1) are less than the SHSS and HSS preconditioners. Algorithm 2.1 is required to solve two linear subsystems with coefficient matrices $A$ and $\beta I + D$, while Algorithm 2.3 need to solve the linear subsystems with coefficient matrices $A$ and $\frac{1}{\alpha} BB^T + D$. In many cases, $D$ has very sparse structure, such as $D = \nu I$ or $D$ is a diagonal matrix [30–32], this means that the matrix $\beta I + D$ is easier to implement than the matrix $\frac{1}{\alpha} BB^T + D$. Thus, the computational cost of applying the BP preconditioner is cheaper than the computational cost of the SHSS preconditioner. For the HSS preconditioner, we can find that three sub-linear systems with coefficient matrices $\alpha I + A$, $\alpha I + D$ and $\frac{1}{\alpha} BB^T + \alpha I$ need to be solved at each step. Thus, the BP preconditioner is much easier to implement than the HSS preconditioner.

## 3. Convergence analysis of the BP iteration method

We now discuss the convergence properties of the BP iteration method for solving the linear system (1.1) and then analyze the choice of quasi-optimal parameter.

Let $\rho(\mathcal{T}(\alpha,\beta))$ denote the spectral radius of the iterative matrix $\mathcal{T}(\alpha,\beta)$. Then the BP iteration scheme (2.4) is convergent if and only if $\rho(\mathcal{T}(\alpha,\beta)) < 1$, see [5, 7, 9]. The following lemma is useful to analyze the spectral properties of iteration matrix.

**Lemma 3.1.** *Let $0 < a < b$ be two positive scalars. Define the function $g(\lambda,\beta)$ as*

$$g(\lambda,\beta) := \frac{\beta - \lambda}{\beta} \quad \text{with} \quad \beta > 0 \quad \text{and} \quad \lambda > 0.$$

*Then we have*

$$\min_{\beta>0} \max_{a\le\lambda\le b} |g(\lambda,\beta)| = \frac{b-a}{b+a}, \tag{3.1}$$

*where the equality holds at $\beta = \frac{a+b}{2}$.*

*Proof.* Obviously,

$$\max_{a\le\lambda\le b} g(\lambda,\beta) = \frac{\beta - a}{\beta} \quad \text{and} \quad \min_{a\le\lambda\le b} g(\lambda,\beta) = \frac{\beta - b}{\beta}.$$

Then we have

$$\max_{a\le\lambda\le b} |g(\lambda,\beta)| = \max\left\{\frac{\beta-a}{\beta}, \frac{b-\beta}{\beta}\right\}.$$

Let

$$f(\beta) := \max_{a\le\lambda\le b} |g(\lambda,\beta)|. \tag{3.2}$$

By the Chebyshev approximation theorem, the minimizer $\beta$ of the function $f(\beta)$ is the unique positive root of the following algebraic equation

$$\frac{\beta - a}{\beta} = \frac{b-\beta}{\beta}.$$

By solving this equation we obtain $\beta = \frac{a+b}{2}$. Substituting this $\beta$ into the expression of $f(\beta)$ in (3.2), we easily obtain (3.1). □

**Theorem 3.2.** *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite, $B \in \mathbb{R}^{m\times n}$ be of full row rank, $\alpha, \beta$ be positive parameters and $\mathcal{T}(\alpha,\beta)$ be defined in (2.5). Then $\rho(\mathcal{T}(\alpha,\beta)) < 1$ if*

$$\beta > \frac{\eta_1}{2} \quad \text{and} \quad \alpha \neq 0,$$

*where $\eta_1$ is the maximum eigenvalue of the matrix $BA^{-1}B^T$.*

*Proof.* Denote $\widetilde{D} := \beta I + D$. From (2.6), we get

$$\mathscr{P}_{\mathrm{BP}}(\alpha,\beta)^{-1} = \begin{bmatrix} I & -\frac{1}{\alpha}B^T \\ 0 & I \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \beta I + D \end{bmatrix}^{-1}\begin{bmatrix} I & 0 \\ B & I \end{bmatrix}\begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} - \frac{1}{\alpha}B^T\widetilde{D}^{-1}BA^{-1} & -\frac{1}{\alpha}B^T\widetilde{D}^{-1} \\ \widetilde{D}^{-1}BA^{-1} & \widetilde{D}^{-1} \end{bmatrix}. \tag{3.3}$$

From (2.5) and (3.3), we have

$$\mathscr{T}(\alpha,\beta) = \begin{bmatrix} 0 & \frac{1}{\alpha}B^T - A^{-1}B^T + \frac{1}{\alpha}B^T\widetilde{D}^{-1}BA^{-1}B^T - \frac{\beta}{\alpha}B^T\widetilde{D}^{-1} \\ 0 & \widetilde{D}^{-1}(\beta I - BA^{-1}B^T) \end{bmatrix}. \tag{3.4}$$

From (3.4), it is clear that $\mathscr{T}(\alpha,\beta)$ has $n$ eigenvalues at 0 and $m$ eigenvalues which are the same as those of $\widetilde{D}^{-1}(\beta I - BA^{-1}B^T)$.

Define $G := \widetilde{D}^{-1}(\beta I - BA^{-1}B^T)$. Then it is easy to see that

$$\rho(\mathscr{T}(\alpha,\beta)) = \rho(G)) \le \|G\|_2 \le \left\|\widetilde{D}^{-1}\right\|_2 \left\|\beta I - BA^{-1}B^T\right\|_2. \tag{3.5}$$

As the matrices $D$ and $A$ are symmetric, $\widetilde{D}^{-1}$ and $\beta I - BA^{-1}B^T$ are symmetric too. Thus

$$\|\widetilde{D}^{-1}\|_2 = \max_{1\le i\le m}\left|\lambda_i(\widetilde{D}^{-1})\right| = \max_{1\le i\le m}\left|\frac{1}{\beta + \lambda_i(D)}\right| \le \frac{1}{\beta}. \tag{3.6}$$

$$\|\beta I - BA^{-1}B^T\|_2 = \max_{1\le i\le m}\left|\lambda_i(\beta I - BA^{-1}B^T)\right| = \max_{1\le i\le m}\left|\beta - \lambda_i(BA^{-1}B^T)\right|. \tag{3.7}$$

Based on (3.5), (3.6) and (3.7), we get

$$\rho(\mathscr{T}(\alpha,\beta)) \le \sigma(\beta) = \max_{1\le i\le m}\left|\frac{\beta - \lambda_i(BA^{-1}B^T)}{\beta}\right| \tag{3.8}$$

Applying straightforward derivations, we get $\beta > \frac{\eta_1}{2} \Rightarrow \sigma(\beta) < 1$, which means

$$\beta > \frac{\eta_1}{2} \Rightarrow \rho(\mathscr{T}(\alpha,\beta)) < 1.$$

□

From the results of Theorem 3.2, we find that $\sigma(\beta)$ is an upper bound of $\rho(\mathscr{T}(\alpha,\beta))$. Therefore, we can obtain the following quasi-optimal parameter:

**Theorem 3.3.** *Assume that the conditions of Theorem 3.2 hold. Let $\eta_1$ and $\eta_m$ be the maximum and minimum eigenvalues of the matrix $BA^{-1}B^T$, respectively, $\sigma(\beta)$ be defined in (3.8). Then the quasi-optimal parameter $\beta_{qopt}$, which minimizes the spectral factor $\sigma(\beta)$ is given by*

$$\beta_{qopt} = \arg\min_{\beta}\sigma(\beta) = \frac{\eta_1 + \eta_m}{2}.$$

*The corresponding quasi-optimal spectral factor is*

$$\sigma_{qopt}(\beta) = \frac{\eta_1 - \eta_m}{\eta_1 + \eta_m}.$$

*Proof.* From Lemma 2.1, the conclusion is easy to be obtained. □

**Corollary 3.4.** *Under the assumptions of Theorem 3.3, if $D = 0$, then the BP iterative method (2.4) is convergent if the following inequality is satisfied*

$$\beta > \frac{\eta_1}{2}, \quad \forall \alpha > 0.$$

*The optimal parameter $\beta_{opt}$, which minimizes the spectral radius of the BP iteration matrix (2.4), is given by*

$$\beta_{opt} = \frac{\eta_1 + \eta_m}{2}.$$

*The corresponding optimal spectral radius is*

$$\rho_{opt}(\mathscr{T}(\alpha,\beta)) = \frac{\eta_1 - \eta_m}{\eta_1 + \eta_m}.$$

*Proof.* As $D = 0$, then $\widetilde{D} = \beta I$, which means $G = \frac{1}{\beta}(\beta I - BA^{-1}B^T)$. From (3.4), we know that

$$\rho(\mathscr{T}(\alpha,\beta)) = \rho(G)) = \max_{1\le i\le m} |1 - \frac{1}{\beta}\eta_i| = \max\left\{\frac{\beta - \eta_m}{\beta}, \ \frac{\eta_1 - \beta}{\beta}\right\}.$$

Thus,

$$\rho(\mathscr{T}(\alpha,\beta)) < 1 \Leftrightarrow \begin{cases} \frac{\beta-\eta_m}{\beta} < 1, \\ \frac{\eta_1-\beta}{\beta} < 1. \end{cases}$$

Simplify these inequalities, we obtain

$$\rho(\mathscr{T}(\alpha,\beta)) < 1 \Leftrightarrow \beta > \frac{\eta_1}{2}.$$

By Lemma 3.1, we can get the results above.

□

**Remark 3.5.** *From the results of Theorem 3.3, we see that $\rho_{opt}$ is related to the eigenvalues of $BA^{-1}B^T$, it may be difficult to compute such value. Therefore, in the next, we will use the techniques similar to those in [19, 24, 27] to discuss a choice strategy of the quasi-optimal parameters $\alpha$ and $\beta$.*

*As the difference between $A$ and $\mathscr{P}_{BP}(\alpha,\beta)$ is*

$$\mathscr{R}_{BP}(\alpha,\beta) := \mathscr{P}_{BP}(\alpha,\beta) - \mathscr{A} = \begin{bmatrix} 0 & \frac{1}{\alpha}AB^T - B^T \\ 0 & \beta I - \frac{1}{\alpha}BB^T \end{bmatrix}.$$

*Then we define*

$$\alpha_{\boldsymbol{prac}} = \arg\min_{\alpha} \|\alpha I - A\|_F^2, \qquad \beta_{\boldsymbol{prac}} = \arg\min_{\beta} \left\|\beta I - \frac{1}{\alpha}BB^T\right\|_F^2. \tag{3.9}$$

*From (3.9), we can get the following practical parameters*

$$\alpha_{\boldsymbol{prac}} = \frac{trace(A)}{n}, \qquad \beta_{\boldsymbol{prac}} = \frac{n\cdot\left\|B^T\right\|_F^2}{m\cdot trace(A)}, \tag{3.10}$$

*which make $\mathscr{P}_{BP}(\alpha,\beta)$ as close to $\mathscr{A}$ as possible.*

From (3.10), we can see that the selection of accelerated parameters $\alpha$ and $\beta$ is very economic, because it avoids calculating the eigenvalues or singular values of matrix.

It is deserve to mentioning that Cao et al. [18] have given a choice for the parameter $\alpha$:

$$\alpha = \frac{\|BB^T\|_2}{\|D\|_2}$$

Firstly, it is difficult to compute the value of $\alpha$ when the problem size is large. Secondly, when $D$ is a zero matrix, it will be not feasible. Therefore, by the algebraic estimation technique used in [24], we will find a suitable parameter

$$\alpha = \frac{\text{trace}(A)}{n} \tag{3.11}$$

to approximate $\alpha$, which is proved to be very efficient in Section 5.

## 4. Spectral properties of the BP preconditioned matrix

It is known that, when a preconditioner is applied to Krylov subspace method, the convergence rate of the corresponding preconditioned iterative method depends on both the spectrum and eigenvector distributions of the preconditioned matrix [18, 34]. The following theorem describes the eigenvalue distributions of the preconditioned matrix $\mathscr{P}_{\rm BP}(\alpha,\beta)^{-1}\mathscr{A}$.

**Theorem 4.1.** *Assume that the conditions of Theorem 3.3 hold. Then the preconditioned matrix $\mathscr{P}_{BP}(\alpha,\beta)^{-1}\mathscr{A}$ has an unit eigenvalue of algebraic multiplicity at least $n$, and its remaining eigenvalues are all real and lie in the following interval:*

$$\left[\frac{2\eta_m}{\eta_1+\eta_m},\frac{2\eta_1}{\eta_1+\eta_m}\right].$$

*Proof.* From (3.4), we have

$$\begin{aligned}\mathscr{P}_{\rm BP}(\alpha,\beta)^{-1}\mathscr{A} &= I-\mathscr{T}(\alpha,\beta)\\ &=\begin{bmatrix} I & A^{-1}B^T-\frac{1}{\alpha}B^T\widetilde{D}^{-1}(D+BA^{-1}B^T)\\ 0 & \widetilde{D}^{-1}(D+BA^{-1}B^T)\end{bmatrix}.\end{aligned}\tag{4.1}$$

Thus, from (4.1), it obtains that the preconditioned matrix $\mathscr{P}_{\rm BP}(\alpha,\beta)^{-1}\mathscr{A}$ has eigenvalue 1 with multiplicity at least $n$, the remaining eigenvalues are the same as those of $\widetilde{D}^{-1}(D+BA^{-1}B^T)$. As both $D=\beta I+D$ and $D+BA^{-1}B^T$ are symmetric positive definite, we know that all of the eigenvalues of $\widetilde{D}^{-1}(D+BA^{-1}B^T)$ are real and positive.

Furthermore, we have

$$\mathrm{sp}\left(\mathscr{P}_{\rm BP}(\alpha,\beta)^{-1}\mathscr{A}\right)=\mathrm{sp}\left(\widetilde{D}^{-1}(D+BA^{-1}B^T)\right)\subseteq\left[1-\rho\left(\mathscr{T}(\alpha,\beta)\right),1+\rho\left(\mathscr{T}(\alpha,\beta)\right)\right],$$

where sp(•) denotes the spectrum of a matrix. By the results of Theorem 3.3, we know that

$$\rho\left(\mathscr{T}(\alpha,\beta)\right)\le\frac{\eta_1-\eta_m}{\eta_1+\eta_m},$$

this also implies that

$$\mathrm{sp}\left(\mathscr{P}_{\rm BP}(\alpha,\beta)^{-1}\mathscr{A}\right)\subseteq\left[\frac{2\eta_m}{\eta_1+\eta_m},\frac{2\eta_1}{\eta_1+\eta_m}\right].$$

Thus, we complete the proof. □

Then, we discuss the eigenvector distributions and the upper bound of the degree of the minimal polynomial of the preconditioned matrix $\mathscr{P}_{\rm BP}(\alpha,\beta)^{-1}\mathscr{A}$. As the proofs are similar to those in [18, 19, 27], we omit them here.

**Theorem 4.2.** *Let $\mathscr{P}_{BP}(\alpha,\beta)$ be defined in (2.2). Then the preconditioned matrix $\mathscr{P}_{BP}(\alpha,\beta)^{-1}\mathscr{A}$ has*

**(1)** *$n$ linearly independent eigenvectors of the form $[u_l^T\ \ 0^T]^T$ $(l=1,2,\cdots,n)$, that correspond to the unite eigenvalue, where $u_l$ $(l=1,2,\cdots,n)$ denote arbitrary linearly independent vectors;*

**(2)** *$i$ $(1\le i\le m)$ eigenvectors of the form $[(u_l^1)^T\ (v_l^1)^T]^T$, with $v_l^1\neq 0$, $(A-\alpha I)B^Tv_l^1=0$, $i=\dim\{null\,(A-\alpha I)\cap range\,(B^T)\}$, that correspond to the eigenvalues 1, where $u_l^1$ are arbitrary vectors;*

**(3)** *$j$ $(1\le j\le m)$ eigenvectors of the form $[(u_l^2)^T\ \ (v_l^2)^T]^T$ with $v_l^2\neq 0$, $(D+BA^{-1}B^T)v_l^2=\lambda_l(\beta I+D)v_l^2$, $u_l^2=\frac{1}{\alpha(1-\lambda_l)}A^{-1}(\lambda_lA+(1-\alpha)I)B^Tv_l^2$, that correspond to the non-unite eigenvalues $\lambda$.*

**Theorem 4.3.** *Under the assumptions of Theorem 4.1. If $1\le k\le m$ is the degree of the minimal polynomial of the matrix $G_2:=(\beta I+D)^{-1}(D+BA^{-1}B^T)$. Then the degree of the minimal polynomial of the preconditioned matrix $\mathscr{P}_{BP}(\alpha,\beta)^{-1}\mathscr{A}$ is at most $k+1$.*

**Remark 4.4.** *Theorem 4.3 indicates that the dimension of the Krylov subspace $\mathscr{K}(\mathscr{P}_{BP}^{-1}\mathscr{A},b)$ is at most $k+1$, which also implies that the GMRES method with the BP preconditioner to solve the linear system (1.1) will terminate in at most $k+1$ steps [34]. While in [18], Cao et al. proved that the GMRES method with the SHSS preconditioner to solve the linear system (1.1) will converge to the exact solution of the linear system with the coefficient matrix $\mathscr{A}$ with at most $m+1$ iterations, this indicates that our results are better than theirs.*

## 5. Numerical results

In this section, we will give some numerical experiments to illustrate the effectiveness of the BP preconditioner. The experiments also aim at identifying the efficiency of the quasi-optimal parameters $\alpha$ and $\beta$. All the tests are performed in MATLAB R2013a with machine precision $10^{-16}$ on a personal computer with 3.2 GHz CPU (Intel(R) Core(TM)2 i5-3470). The initial guess is chosen as $u^{(0)} = 0$, and terminated once the current iterate solution $u^{(k)}$ satisfies

$$\mathrm{ERR} = \frac{\|b - \mathcal{A}u^{(k)}\|_2}{\|b\|_2} \leq 10^{-6}$$

or the number of the prescribed iteration $k_{\max} = 500$ are exceeded. In the all following tables, " – " means that $IT \geq 500$ or the $ERR > 10^{-6}$. In the following, we show the advantage of the BP preconditioner over some existing preconditioners from aspects of number of iteration steps (denoted by "IT"), elapsed CPU times in seconds (denoted by "CPU") .

### 5.1. *Stokes problems*

**Example 1.** *([5, 12, 14]) Consider the following two-dimensional Stokes problem:*

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

*where the boundary and the normalization conditions $u = 0$ on $\partial\Omega$ and $\int_\Omega p(x)\,\mathrm{d}x = 0$, $\Omega = [0,1]\times[0,1]$, $\partial\Omega$ is the boundary of $\Omega$. $\Delta$ is the componentwise Laplacian operator, $\nabla$ and $\nabla\cdot$ denote the gradient and divergence operators, $u$ and $p$ are two vectors, the velocity of the fluid and the pressure, respectively. By discretizing this problem with finite differences, we obtain the linear system (1.1), in which*

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

*and $T = \frac{1}{h^2}\cdot tridiag(-1,\ 2,\ -1) \in \mathbb{R}^{p\times p}$, $F = \frac{1}{h}\cdot tridiag(-1,\ 1,\ 0) \in \mathbb{R}^{p\times p}$ with $\otimes$ being the Kronecker product symbol and $h = \frac{1}{p+1}$ the discretization mesh-size. In this example, we let the right-hand side vector $b = (f^T, g^T)^T \in \mathbb{R}^{(m+n)\times 1}$ with $f = (1,1,\cdots,1)^T \in \mathbb{R}^{n\times 1}$ and $g = (1,1,\cdots,1)^T \in \mathbb{R}^{m\times 1}$. We note that $n = 2p^2$ and $m = p^2$. Hence, the total number of unknowns is $m+n = 3p^2$. As $D = 0$, the SHSS preconditioner is reduced to the RHSS preconditioner [19]. Therefore, in this example, we compare the RHSS preconditioner with the BP preconditioner to illustrate the effectiveness of the BP preconditioner. The numerical results for Example 1 are listed in Table 1.*

**Example 2.** *([4, 9, 18]) We still consider the two-dimensional Stokes problem (5.1). But the square domain we used is $\Omega = [-1,1]\times[-1,1]$ and with the non-zero horizontal velocity on the top part of the domain, namely $\frac{\partial u}{\partial x} = 1 - x^4$ on $[-1,1]\times\{1\}$. This test problem is a 'regularized' two-dimensional lid-driven cavity problem. We discretize the Stokes equations (5.1) by the $Q_1 - P_0$ (the stabilization parameter is 0.25) and $Q_2 - Q_1$ finite elements on some uniform grids, respectively. We use the IFISS software package [22] to generate linear systems for the meshes of size $64\times 64$, $128\times 128$, $256\times 256$, $512\times 512$. The corresponding numerical results are listed in Tables 2 and 3.*

In Examples 1-2, we apply the GMRES iteration method incorporated with the BP preconditioner $\mathcal{P}_{\mathrm{BP}}(\alpha,\beta)$, the SHSS preconditioner $\mathcal{P}_{\mathrm{SHS}}$ [18], the HSS preconditioner $\mathcal{P}_{\mathrm{HSS}}$ [12] or without preconditioner (denoted as $I$), respectively, to solve the linear system (1.1). The parameters $\alpha$ and $\beta$ are obtained by the formula (3.10) for the BP preconditioner. For the SHSS preconditioner, we compute the parameter $\alpha$ by the formula as given in (3.11). The parameters for the HSS preconditioner are the experimentally computed optimal ones that minimize the total number of iteration steps of the HSS-GMRES method. In addition, the coefficient matrix $A$ is approximated by the action of two AMG V-cycles with a 2-2 (presmoothing-postsmoothing) point damped Jacobi smoothing strategy, and we use PCG method to solve the linear sub-systems with respect to the Hermitian positive definite matrices $\beta I + D$, $\frac{1}{\alpha}BB^T + D$, $\alpha I + D$ and $\frac{1}{\alpha}BB^T + \alpha I$.

As shown in Tables 1-3, we can see significant improvements for the performance of GMRES with all preconditioners. Meanwhile, we observe that the iteration steps of the HSS preconditioned-GMRES method increase

rapidly with problem size and the SHSS and BP preconditioners lead to much better results than the HSS preconditioners. It appears to offer advantages in terms of both iteration steps and CPU time. Besides, we also observe that the parameter choice for the BP preconditioner given in the paper is also very feasible. Moreover, we observe that the iteration steps of the BP preconditioner almost keep constant, which shows that the BP preconditioned-GMRES method is mesh-size insensitive convergent.

We also depicted the eigenvalue distributions of the HSS, RHSS and BP preconditioned matrices and the original coefficient matrix $\mathcal{A}$ for Example 1, see Fig. 1. In addition, we have plotted the trends of the iteration steps vs $\alpha$, see Fig. 2. From Fig. 1, we see that the BP preconditioner is better than others since it has a more compact spectral distribution, which tends to result in a faster convergence rate. Observed from Fig. 2, we find that the IT of BP preconditioner changes little and it becomes stable when $\alpha > 400$, which implies that our preconditioner is not very sensitive to $\alpha$, this verifies the validity of our theoretical results.

Table 1: Numerical results of Example 1.

| Pre | | $n+m$ | | | |
|---|---|---|---|---|---|
| | | 12288 | 49152 | 196608 | 786432 |
| $\mathscr{P}_{\rm HSS}$ | $\alpha$ | 90 | 180 | 335 | 360 |
| | IT | 51 | 67 | 92 | 108 |
| | CPU | 2.5280 | 10.8504 | 307.7131 | 1144.4793 |
| $\mathscr{P}_{\rm RHSS}$ | $\alpha$ | 16900 | 66564 | 264196 | – |
| | IT | 49 | 70 | 97 | – |
| | CPU | 5.3710 | 20.0622 | 204.2688 | – |
| $\mathscr{P}_{\rm BP}(\alpha,\beta)$ | $\alpha$ | 16900 | 66564 | 264196 | 1052676 |
| | $\beta$ | 0.9922 | 0.9961 | 0.9980 | 0.9990 |
| | IT | 15 | 17 | 19 | 20 |
| | CPU | 1.8400 | 4.1576 | 38.7896 | 460.2774 |

Table 2: Numerical results for Example 2 discretized by the $Q_2 - P_0$ finite elements.

| Pre | grids(DOF) | $64\times 64$ (12546) | $128\times 128$(49666) | $256\times 256$(197634) | $512\times 512$ (788482) |
|---|---|---|---|---|---|
| $I$ | IT | – | – | – | – |
| | CPU | – | – | – | – |
| $\mathscr{P}_{\rm HSS}$ | $\alpha$ | 0.002 | 0.013 | 0.006 | 0.001 |
| | IT | 60 | 82 | 109 | 150 |
| | CPU | 2.4756 | 9.6513 | 74.8147 | 901.7666 |
| $\mathscr{P}_{\rm SHSS}$ | $\alpha$ | 3.5724 | 3.6117 | 3.6316 | – |
| | IT | 35 | 50 | 67 | – |
| | CPU | 4.1299 | 24.7710 | 225.9281 | – |
| $\mathscr{P}_{\rm BP}(\alpha,\beta)$ | $\alpha$ | 2.4708 | 2.5657 | 2.6154 | 2.6408 |
| | $\beta$ | 0.0030 | 0.0007 | 0.0002 | 0.0001 |
| | IT | 22 | 24 | 26 | 28 |
| | CPU | 1.7615 | 7.9486 | 59.9862 | 308.5899 |

### 5.2. *Weighted least squares problems*

**Example 3.** *([20, 30–33]) We consider an example arising from the weighted least square problem*

$$\min_{x\in\mathbb{R}} ||Bx-\tilde{b}||_2^2 \quad with \quad B=\begin{pmatrix} \Xi K \\ \sqrt{\nu} I \end{pmatrix} \quad and \quad \tilde{b}=\begin{pmatrix} \Xi f \\ 0 \end{pmatrix}, \tag{5.2}$$

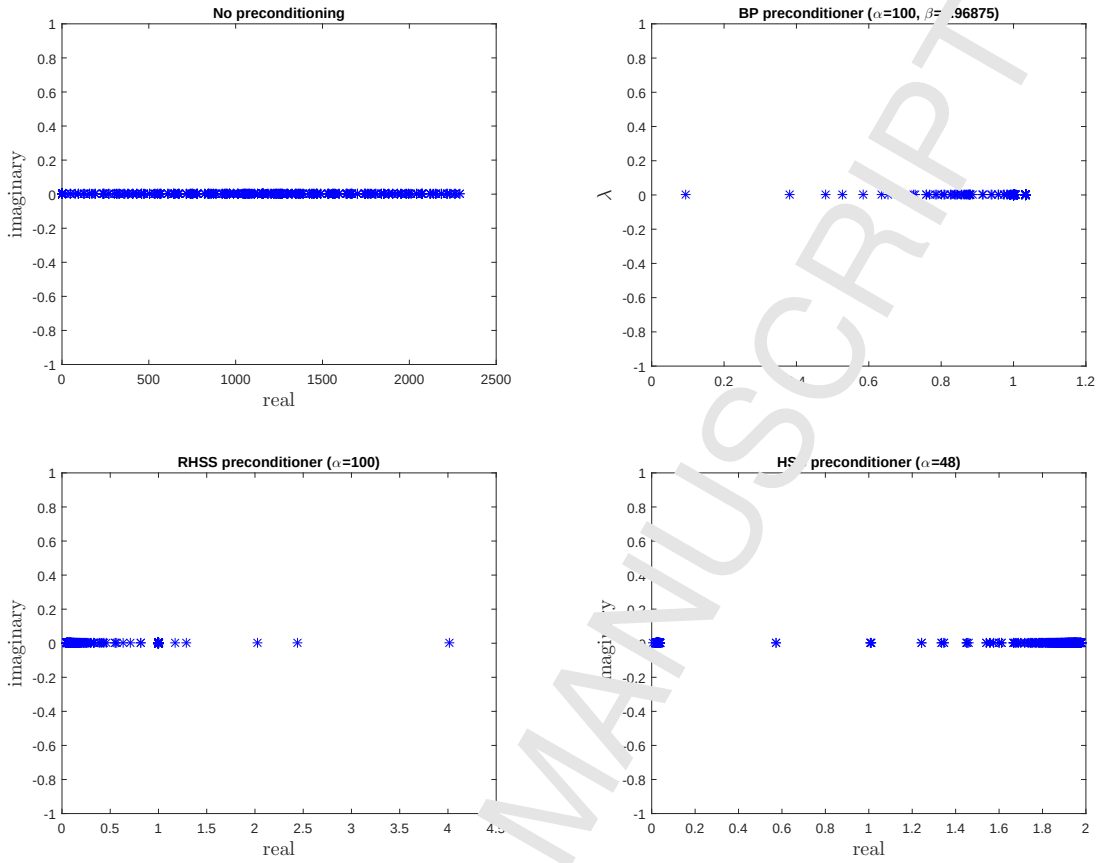


Figure 1: Eigenvalue distributions of the preconditioned matrices for Example 1 ($p = 16$).

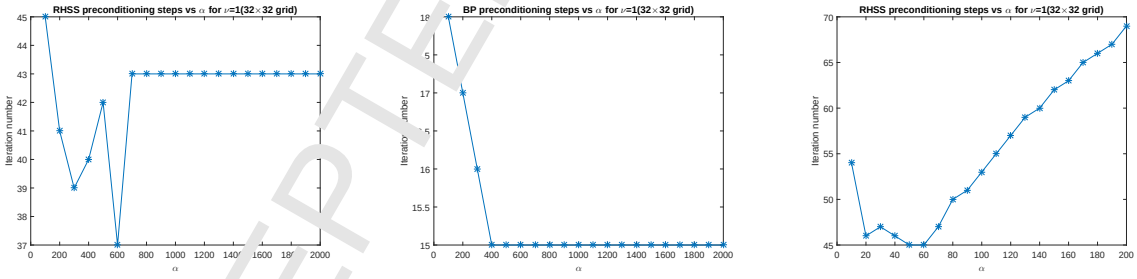


Figure 2: The iteration steps vs $\alpha$ for Example 1($p = 32$).

*where* $\Xi \in \mathbb{R}^{m\times m}$ $(m \geq n)$ *is a positive diagonal matrix,* $K \in \mathbb{R}^{m\times n}$ *is a Toeplitz matrix of full column rank,* $f \in \mathbb{R}^m$ *is a given vector and* $\nu > 0$ *is a regularization parameter. This problem is also tested in [30, 31]. Let* $M = \Xi^{(-2)}$ *and* $y = \Xi^2(f - Kx)$*, then solving the least squares problem (5.2) is equivalent to solving the saddle point system*

$$\begin{pmatrix} M & K \\ -K^T & \nu I \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \equiv b. \tag{5.3}$$

Table 3: Numerical results for Example 2 discretized by the $Q_2 - Q_1$ finite elements.

| Pre | grids(DOF) | $64\times64$(12546) | $128\times128$(49666) | $256\times256$(1[illegible]) | $512\times512$ (788482) |
|---|---|---|---|---|---|
| $I$ | IT | – | – | – | – |
| | CPU | – | – | - | – |
| $\mathscr{P}_{\rm HSS}$ | $\alpha$ | 0.12 | 0.03 | - | – |
| | IT | 145 | 474 | - | – |
| | CPU | 5.1203 | 56.6073 | - | – |
| $\mathscr{P}_{\rm RHSS}$ | $\alpha$ | 3.6930 | 3.8421 | 3.91[illegible]9 | 3.9597 |
| | IT | 28 | 41 | [illegible]8 | 73 |
| | CPU | 0.7970 | 3.4178 | 17[illegible]392 | 210.0296 |
| $\mathscr{P}_{\rm BP}(\alpha,\beta)$ | $\alpha$ | 3.6930 | 3.8421 | 3.9199 | 3.9597 |
| | $\beta$ | 0.0087 | 0.0023 | 0.0006 | 0.0002 |
| | IT | 54 | 60 | 64 | 67 |
| | CPU | 0.7332 | 2.9872 | 13.6346 | 173.3568 |

*As considered in [30, 31], $\nu$ is chosen to be $\nu = 0.001$. The Toeplitz matrix $K \in \mathbb{R}^{n\times n}$ a square matrix with its entries*

$$t_{ij} = \frac{1}{\sqrt{|i-j|+1}}, \tag{5.4}$$

*or*

$$t_{ij} = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-|i-j|^2}{2\sigma^2}} \quad (\sigma = 2). \tag{5.5}$$

*The matrix $K$ defined in (5.4) is well-conditioned, while $K$ defined in (5.5) is highly ill-conditioned. The matrix $M$ is a positive diagonal random matrix generated by the following MATLAB codes*

$p = 1;$
$rand('seed', 1)$
$s = rand(m, 1);$
$maxs = \max(s);$
$mins = \min(s);$
$a = (cond * p - p)/(maxs - mins * cond);$
$b = a * s + p;$
$M = diag(abs(sparse(b)))).$

*We scale its diagonal entries so that the condition number of $M$ is around $10^6$, i.e., we set $cond = 10^6$. The given vector $f$ is $(1, 1, \cdots, 1)^T$.*

Benzi and Ng [13] have considered the HSS preconditioner and the constraint preconditioner for the saddle point problem (5.3). Then Pan and Ng [30] have proposed the NHSS preconditioner, which can accelerate the convergence rate of the HSS preconditioner. Here the HSS preconditioner, the constraint preconditioner and the NHSS preconditioner are

$$\mathscr{P}_{\rm NHSS} = \frac{1}{2}\Sigma^{-1}(\Sigma + H)(\Sigma + S), \quad \mathscr{P}_C = \begin{pmatrix} \gamma I & K^T \\ K & -\nu I \end{pmatrix},$$

where

$$H = \begin{pmatrix} M & 0 \\ 0 & \nu I \end{pmatrix} \quad S = \begin{pmatrix} 0 & K^T \\ -K & 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \alpha I & 0 \\ 0 & \nu I \end{pmatrix}, \quad \gamma = \frac{m_{11} + m_{22} + \cdots + m_{nn}}{n},$$

Table 4: Numerical results of iteration methods of Example 3 for the well-conditioned $K$.

| | $\mathscr{P}_{\rm C}$ | | $\mathscr{P}_{\rm NHSS}$ | | $\mathscr{P}_{\rm BP}(\alpha,\beta)$ | | $\mathscr{P}_{\rm SHSS}$ | |
|---|---|---|---|---|---|---|---|---|
| $2n$ | IT | CPU | IT | CPU | IT | CPU | IT | CPU |
| 8192 | 28 | 1.1875 | 32 | 1.2969 | 14 | **0.1563** | 18 | 0.7344 |
| 32768 | 26 | 3.4375 | 33 | 4.3281 | 24 | **0.7969** | 33 | 4.1875 |
| 131072 | 25 | 20.3125 | 33 | 28.3594 | 34 | **10.9531** | 52 | 35.4531 |
| 524288 | 81 | 381.1719 | 70 | 309.7656 | 28 | **30.7188** | 52 | 118.3125 |
| 2097152 | 55 | 1046.7813 | 61 | 1282.0625 | 42 | **239.7500** | 64 | 1330.7969 |

Table 5: Numerical results of iteration methods of Example 3 for the ill-conditioned $K$.

| | $\mathscr{P}_{\rm C}$ | | $\mathscr{P}_{\rm NHSS}$ | | $\mathscr{P}_{\rm BP}(\alpha,\beta)$ | | $\mathscr{P}_{\rm SHSS}$ | |
|---|---|---|---|---|---|---|---|---|
| $2n$ | IT | CPU | IT | CPU | IT | CPU | IT | CPU |
| 8192 | 102 | 3.6719 | 20 | 0.5469 | 9 | **0.1250** | 9 | 0.2813 |
| 32768 | 103 | 10.3906 | 20 | 1.7344 | 17 | **0.4844** | 17 | 1.1250 |
| 131072 | 108 | 102.2500 | 22 | 12.4688 | 27 | **1.6719** | 27 | 12.2344 |
| 524288 | – | – | 18 | 40.1563 | 10 | **7.0313** | 10 | 13.6719 |
| 2097152 | – | – | 20 | 202.4375 | 26 | **120.7094** | 26 | 188.5625 |

$m_{ii}, i = 1,2,...,n$ are the main diagonal elements of $M$.

In this example, we compare the proposed preconditioner $\mathscr{P}_{\rm BP}(\alpha,\beta)$ with the preconditioners $\mathscr{P}_{NHSS}$, $\mathscr{P}_{\rm SHSS}$ and $\mathscr{P}_{\rm C}$. In the implementation of the preconditioned GMRES, the linear sub-systems with the coefficient matrices $\nu I+\frac{1}{\gamma}KK^T$ and $\nu I+\frac{1}{\alpha}KK^T$ are solved iteratively by the PCG method with the preconditioner $\nu I+\frac{1}{\gamma}CC^T$ and the preconditioner $\nu I+\frac{1}{\alpha}CC^T$, respectively, where the circulant preconditioner $C$ used here is T. Chans preconditioner. The inner stop criterion is set to be $\|r_k\|/\|r_0\| < 10^{-5}$. The parameter values of the BP and SHSS preconditioners are chosen by the same way as discussed in Section 5. Iteration counts and CPU times for Example 3 are displayed in Tables 4-5.

From Tables 4-5, we can see that when $K$ is well-conditioned, all of the preconditioners succeed in solving the problem for all $n$. For the well-conditioned $K$, the BP preconditioner is the best one with respect to IT and CPU in Table 4. For the ill-conditioned $K$, the BP preconditioner costs the same iteration steps as the SHSS preconditioner, but it costs much less CPU time than the SHSS preconditioner. The reason is that the BP preconditioner avoids solving the linear sub-systems with respect to the coefficient matrix $\nu I+\frac{1}{\alpha}KK^T$ within Krylov subspace acceleration, it only involves the calculation of Toeplitz matrix-vector product, which can be obtained by a fast algorithms. In addition, for the well-conditioned $K$, the constraint preconditioner $\mathscr{P}_{\rm C}$ show better performance than the NHSS preconditioner $\mathscr{P}_{\rm NHSS}$, however, it is inefficient for the ill-conditioned $K$. Therefore, our proposed preconditioner $\mathscr{P}_{\rm BP}(\alpha,\beta)$ are very efficient for both the well-conditioned $K$ and the ill-conditioned $K$.

## 6. Conclusions

In this paper, we have established and analyzed a new BP preconditioner for solving the saddle point problems. The conditions for guaranteeing the convergence of the BP iterative method are derived, and the spectral properties of preconditioned matrix are discussed. A feasible strategy to chose the quasi-optimal parameters is obtained. The implementation of the BP preconditioner is given, revealing that the algorithmic cost of the BP preconditioner is lower than that of the exist ones when they are applied to accelerate the GMRES method. Numerical experiments show that the GMRES method is greatly accelerated by utilizing the BP preconditioner with the quasi-optimal parameter.

**Acknowledgement**

We would like to express our sincere thanks to the reviewers for their useful comments and valuable suggestions, which greatly improved the original manuscript of this paper.

## References


[1] Z.-Z. Bai, Structured preconditioners for nonsingular matrices of block two-by-two structures, Math. Comput. 75 (2006) 791–815.
[2] Z.-Z. Bai, Optimal parameters in the HSS-like methods for saddle-point problems, Numer. Linear Algebra Appl. 16 (2009) 447–479.
[3] Z.-Z. Bai, G. Golub, M. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl. 24 (2003) 603–626.
[4] Z.-Z. Bai, G. Golub, Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems, IMA J. Numer. Anal. 27 (2007) 1–23.
[5] Z.-Z. Bai, G. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, Numer. Math. 98 (2004) 1–32.
[6] Z.-Z. Bai, M. Ng, On inexact preconditioners for nonsymmetric matrices, SIAM J. Sci. Comput. 26 (2005) 1710–1724.
[7] Z.-Z. Bai, Z.-Q. Wang, On parameterized inexact Uzawa methods for generalized saddle point problems, Linear Algebra Appl. 428 (2008) 2900–2932.
[8] Z.-Z. Bai, G. Golub, C.-K. Li, Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices, SIAM J. Sci. Comput. 28 (2006) 583–603.
[9] Z.-Z. Bai, B. Parlett, Z.-Q. Wang, On generalized successive over relaxation methods for augmented linear systems, Numer. Math. 102 (2005) 1–38.
[10] Z.-Z. Bai, Z.-Q. Wang, Restrictive preconditioners for conjugate gradient methods for symmetric positive definite linear systems, J. Comput. Appl. Math. 187 (2006) 202–226.
[11] Z.-Z. Bai, G.-Q. Li, Restrictively preconditioned conjugate gradient methods for systems of linear equations, IMA J. Numer. Anal. 23 (2003) 561–580.
[12] M. Benzi, G. Golub, A preconditioner for generalized saddle point problems, SIAM J. Matrix Anal. Appl. 26 (2004) 20–41.
[13] M. Benzi, M. K. Ng. Preconditioned iterative methods for weighted toeplitz least squares problems. SIAM J. Matrix Anal. Appl. 27(4) (2006) 1106–1124.
[14] M. Benzi, G. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[15] J. Bramble, J. Pasciak, A. Vassilev, Uzawa type algorithms for nonsymmetric saddle point problems, Math. Comput. 69 (2000) 667–690.
[16] Z.-H. Cao, Positive stable block triangular preconditioners for symmetric saddle point problems, Appl. Numer. Math. 57 (2007) 899–910.
[17] Z.-H. Cao, Constraint Schur complement preconditioners for nonsymmetric saddle point problems, Appl. Numer. Math. 59 (2009) 151–169.
[18] Y. Cao, Z.-R. Ren, Q. Shi, A simplified HSS preconditioner for generalized saddle point problems, BIT Numer. Math. 52(2) (2016), 423–439.
[19] Y. Cao, L.-Q. Yao, M.-Q. Jiang, Q. Niu, A relaxed HSS preconditioner for saddle point problems from meshfree discretization, J. Comput. Math. 31 (2013), 398–421.
[20] X. Chen, On preconditioned Uzawa methods and SOR methods for saddle-point problems, J. Comput. Appl. Math. 100 (1998) 207–224.
[21] F. Chen, Y.-L. Jiang, A generalization of the inexact parameterized Uzawa methods for saddle point problems, Appl. Math. Comput. 206 (2008) 765–771.
[22] H. C. Elman, A. Ramage, D. J. Silvester, IFISS: a Matlab toolbox for modelling incompressible flow, Acm Transactions, Math. Software, 33(2)(2010), 0098–3500.
[23] H. Elman, Preconditioning for the steady-state Navier-Stokes equations with low viscosity, SIAM J. Sci. Comput. 20 (1999) 1299–1316.
[24] Y.-M. Huang A practical formula for computing optimal parameters in the HSS iteration methods. J. Comput. Appl. Math., 255(255) (2014) 142–149.
[25] M.-Q. Jiang, Y. Cao, On local Hermitian and skew-Hermitian splitting iteration methods for generalized saddle point problems, J. Comput. Appl. Math. 231 (2009) 973–982.
[26] A. Klawonn, Block-triangular preconditioners for saddle point problems with a penalty term, SIAM J. Sci. Comput. 19(1)(1998) 172–184.
[27] Z.-Z. Liang, G.-F. Zhang, SIMPLE-like preconditioners for saddle point problems from the steady Navier-Stokes equations, J. Comput. Appl. Math. 302 (2016), 211–223.
[28] L.-D. Liao, G.-F. Zhang, New variant of the HSS iteration method for weighted Toeplitz regularized least-squares problems from image restoration, Comput. Math. Appl. 73(11) (2017) 2482-249949.
[29] V. Simoncini, Block triangular preconditioners for symmetric saddle-point problems, Appl. Numer. Math. 49(1) (2004) 63–80.
[30] M. K. Ng and J.-Y. Pan. Weighted Toeplitz regularized least squares computation for image restoration. *SIAM J. Sci. Comput.*, 36(1) (2014) 94–121.
[31] S.-Q. Shen, T.-Z. Huang, J. F. Zhang, Augmentation block triangular preconditioners for regularized saddle point problems, SIAM J. Matrix Anal. Appl. 33(3)(2012) 721–741.
[32] S.-Q. Shen, W.-D. Bao, Z. Jian, On augmentation block triangular preconditioners for regularized saddle point problems, Comput. Math. Appl. 69(8) (2015) 828–837.
[33] S.-Q. Shen, J. Lin, W.-D. Bao, T.-Z. Huang, On the eigenvalue distribution of preconditioned nonsymmetric saddle point matrices. Numer. Linear Algebra Appl. 21(4) (2014) 557–568.
[34] Y. Saad. Iterative methods for sparse linear systems. SIAM, 2003.
[35] J.-F. Yin, Z.-Z. Bai, The restrictively preconditioned conjugate gradient methods on normal residual for block two-by-two linear systems, J. Comput. Math. 26 (2008) 240–249.
[36] Y.-Y. Zhou, G.-F. Zhang, A generalization of parameterized inexact Uzawa method for generalized saddle point problems, Appl. Math. Comput. 21 (2008) 599–607.