



Generalized Shift-Splitting Preconditioner for Saddle Point Problems with Block Three-by-Three Structure

Linna Wang, Ke Zhang*

Department of Mathematics, Shanghai Maritime University, Shanghai, China

Email: 18916137726@163.com, *xznuzk123@126.com

How to cite this paper: Wang, L.N. and Zhang, K. (2019) Generalized Shift-Splitting Preconditioner for Saddle Point Problems with Block Three-by-Three Structure. *Open Access Library Journal*, 6: e5968. <https://doi.org/10.4236/oalib.1105968>

Received: November 29, 2019

Accepted: December 22, 2019

Published: December 25, 2019

Copyright © 2019 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

We propose a generalized shift-splitting iteration method for saddle point problems with block three-by-three structure. As a new iteration method, the method converges to the unique solution of the saddle point problem unconditionally. When exploited as a preconditioner, the spectral distribution of the preconditioned matrix is investigated. Numerical experiments show that the new variant is efficient in speeding up GMRES for solving the block three-by-three saddle point problem.

Subject Areas

Numerical Mathematics

Keywords

Saddle Point Problem, Generalized Shift-Splitting, Krylov Subspace, GMRES

1. Introduction

We consider the saddle point problems with the following block three-by-three structure

$$\begin{bmatrix} A & B^T & 0 \\ -B & 0 & -C^T \\ 0 & C & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f \\ g \\ h \end{bmatrix} \text{ or } \mathcal{A}u = b, \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{s \times m}$ are matrices of full row rank, $f \in \mathbb{R}^n$, $g \in \mathbb{R}^m$ and $h \in \mathbb{R}^s$ are given vectors. The linear system of equations (1) has many practical applications in

scientific computing and engineering fields, including the constrained quadratic optimization problem, the constrained least squares problem and the computational fluid dynamics [1].

In practice, the coefficient matrix A in (1) is generally sparse and of large size. Therefore, iterative methods are usually preferred for their low memory requirement per iteration. Among these candidates, the Uzawa-type methods [2], the matrix splitting methods [3] and the Krylov subspace methods [4] have received much attention during the past decades. For instance, the Uzawa method [2] is a classical approach which is known for its low computational overhead and easy implementation. To improve the numerical performance, some Uzawa-type variants have been proposed recently which include the inexact Uzawa [5] [6] [7], Uzawa-SOR [8] and Uzawa-PSS [9], etc. The other way for solving the saddle point problems is to exploit the matrix splitting. By using the Hermitian and skew-Hermitian splitting (HSS), Bai *et al.* [3] propose the HSS iteration method. Following this reasoning, some efficient variants such as the accelerated Hermitian and skew-Hermitian splitting (AHSS) [10], the modified Hermitian and skew-Hermitian splitting (MHSS) [11] and the relaxed Hermitian and skew-Hermitian splitting (RHSS) [12] have been proposed henceforth. Recently, the sophisticated Krylov subspace methods have been used to solve the saddle point problems. Due to the structure of the saddle point matrices, however, a slow convergence or even stagnation is often witnessed during the actual computation. Such drawback can be alleviated by employing an appropriate preconditioner. In the context of saddle point problems, many efficient preconditioners, to name just a few, the block diagonal preconditioners, the constrained preconditioners, the HSS-based preconditioners, have been developed; see [13] [14] [15] [16] and the more recent monograph [17] for detail.

In [18], Bai *et al.* propose a class of shift-splitting iteration method for solving non-Hermitian positive definite linear system $Au = b$. It can be recast in the following way. Suppose that $A \in \mathbb{R}^{n \times n}$ is a non-Hermitian positive definite matrix. Then the shift-splitting of the coefficient matrix A is given by

$$A = M(\alpha) - N(\alpha) = \frac{1}{2}(\alpha I + A) - \frac{1}{2}(\alpha I - A),$$

which leads naturally to the shift-splitting iteration scheme

$$u^{k+1} = M^{-1}(\alpha)N(\alpha)u^k + M^{-1}(\alpha)b, \quad k = 0, 1, 2, \dots$$

Theoretical analysis shows that shift-splitting iteration is endowed with an unconditional convergence. By using this sound convergence property, Cao *et al.* come up with the shift-splitting to solve the saddle point problems and propose a shift-splitting preconditioner and a local shift-splitting variant [19]. Numerical experiments show that the proposed two shift-splitting-type preconditioners are superior to the HSS preconditioner in terms of CPU time and iteration steps. The shift-splitting technique has also been generalized by incorporating two relaxation parameters [20]. Furthermore, Salkuyeh *et al.* extend it to solve the ge-

neralized saddle point problems and propose a modified generalized shift-splitting preconditioner [21].

By taking into account the special structure of (1), Cao [22] proposes shift-splitting preconditioners for solving (1). Motivated by [20] and [22], we devise a new generalized shift-splitting (GSS) iteration scheme for the structured linear system (1). A corresponding GSS preconditioner is induced to accelerate the convergence of GMRES. Theoretical results on the convergence of GSS iteration method and the eigenvalue distribution of the preconditioned matrix are also given.

The organization of this paper is as follows. In Section 2, we introduce the generalized shift-splitting iteration method for solving the block three-by-three saddle point problem and look into its convergence property. In Section 3, we obtain a GSS preconditioner derived from the GSS iteration and analyze the spectrum of the preconditioned matrix. In Section 4, some numerical experiments are carried out to illustrate the effectiveness of GSS by comparing it with other shift-splitting preconditioners given in [22]. Finally, some conclusions are given in Section 5.

2. Generalized Shift-Splitting Iteration Method and Its Convergence

The proposed generalized shift-splitting (GSS) iteration is based on the work of [22]. Therefore, it is reasonable for us to recap briefly the shift splitting used in [22] before introducing the new iteration method in this section.

In its simplest form, the shift splitting (SS) of the coefficient matrix \mathcal{A} in (1) goes as follows

$$\begin{aligned} \mathcal{A} &= \frac{1}{2}(\alpha I + \mathcal{A}) - \frac{1}{2}(\alpha I - \mathcal{A}) \\ &= \frac{1}{2} \begin{bmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & -C^T \\ 0 & C & \alpha I \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \alpha I - A & -B^T & 0 \\ B & \alpha I & C^T \\ 0 & -C & \alpha I \end{bmatrix}, \end{aligned} \quad (2)$$

where I is the identity matrix with conforming size. The parameter α in (2) is used to monitor the convergence of the SS iteration method and hence the spectrum of the preconditioned matrix. In other words, α plays the same role in adjusting the two block diagonal matrices in (2), *i.e.*,

$$\begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \in \mathbb{R}^{(m+n) \times (m+n)} \quad \text{and} \quad 0 \in \mathbb{R}^{s \times s}. \quad (3)$$

However, the reason for choosing a single parameter to control simultaneously these two block diagonal matrices above remains elusive. Instead, it seems more appealing to investigate the convergence if the two block diagonal matrices are affected by two different parameters separately. This is the main motivation of our work. To this end, we refine the splitting (2) by embedding an extra parameter β in the (3,3)-block, which yields the following generalized shift-

splitting

$$\begin{aligned} \mathcal{A} &= M(\alpha, \beta) - N(\alpha, \beta) = \frac{1}{2}(\Omega + \mathcal{A}) - \frac{1}{2}(\Omega - \mathcal{A}) \\ &= \frac{1}{2} \begin{bmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & -C^T \\ 0 & C & \beta I \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \alpha I - A & -B^T & 0 \\ B & \alpha I & C^T \\ 0 & -C & \beta I \end{bmatrix}. \end{aligned} \quad (4)$$

Here $\alpha > 0$, $\beta > 0$ and $\Omega = \begin{bmatrix} \alpha I & & \\ & \alpha I & \\ & & \beta I \end{bmatrix}$. Then it is trivial to find that

the generalized shift-splitting (4) reduces to the shift-splitting (2) if $\alpha = \beta$.

Using the splitting (4), we obtain the following stationary iteration

$$u^{k+1} = \Gamma_{\text{GSS}} u^k + c, \quad (5)$$

where $\Gamma_{\text{GSS}} = M^{-1}(\alpha, \beta)N(\alpha, \beta)$, $c = M^{-1}(\alpha, \beta)b$ and $k = 0, 1, \dots$. To put it another way, the generalized shift-splitting method (GSS) for the block three-by-three saddle point problem (1) that can be stated as follows. Given an initial vector u^0 , compute

$$\frac{1}{2} \begin{bmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & -C^T \\ 0 & C & \beta I \end{bmatrix} u^{k+1} = \frac{1}{2} \begin{bmatrix} \alpha I - A & -B^T & 0 \\ B & \alpha I & C^T \\ 0 & -C & \beta I \end{bmatrix} u^k + \begin{bmatrix} f \\ g \\ h \end{bmatrix} \quad (6)$$

until u^k converges, where $k = 0, 1, \dots$.

It is known that the GSS iteration method (5) enjoys the unconditional convergence if the spectral radius of the iteration matrix Γ_{GSS} is less than 1. The following results are adapted from [22] and [23] which are instrumental in proving such sound convergence property.

Lemma 1. Suppose that $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{s \times m}$ are of full row rank. Let λ be an eigenvalue of the saddle point matrix \mathcal{A} , then

$$\text{Re}(\lambda) > 0.$$

Proof. The proof can be found in ([22], Lemma 2.2).

Lemma 2. Let $W \in \mathbb{R}^{n \times n}$ be a non-Hermitian positive semidefinite matrix and $E(W, \delta) = (\delta I + W)^{-1}(\delta I - W)$ be the extrapolated Cayley transform with $\delta > 0$. Then

- 1) the spectral radius of $E(W, \delta)$ is bounded by 1, i.e., $\rho(E(W, \delta)) \leq 1$;
- 2) $\rho(E(W, \delta)) = 1$ if and only if the matrix W has a (reducing) eigenvalue of the form $i\xi$ with $\xi \in \mathbb{R}$ and i the imaginary unit.

Proof. See ([23], Theorem 2.1) for detail.

The following theorem on the convergence of GSS iteration method comes out as a result of Lemma 1 and Lemma 2.

Theorem 1. Suppose that $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{s \times m}$ are matrices of full row rank. Let $\alpha > 0$ and $\beta > 0$. Then the spectral radius of Γ_{GSS} defined in (5) satisfies

$$\rho(\Gamma_{GSS}) < 1,$$

i.e., the GSS iteration method converges to the unique solution of (1) unconditionally.

Proof. By (4) and (5), we rewrite the iteration matrix Γ_{GSS} as

$$\Gamma_{GSS} = (\Omega + \mathcal{A})^{-1}(\Omega - \mathcal{A}) = \Omega^{-\frac{1}{2}} \left(I + \Omega^{-\frac{1}{2}} \mathcal{A} \Omega^{-\frac{1}{2}} \right)^{-1} \left(I - \Omega^{-\frac{1}{2}} \mathcal{A} \Omega^{-\frac{1}{2}} \right) \Omega^{\frac{1}{2}}.$$

Let $\hat{\mathcal{A}} = \Omega^{-\frac{1}{2}} \mathcal{A} \Omega^{-\frac{1}{2}}$. It is easy to see that Γ_{GSS} is similar to the matrix $\hat{\Gamma}_{GSS} = (I + \hat{\mathcal{A}})^{-1} (I - \hat{\mathcal{A}})$. Therefore, it suffices to prove $\rho(\hat{\Gamma}_{GSS}) < 1$. By the construction of $\hat{\mathcal{A}} = \Omega^{-\frac{1}{2}} \mathcal{A} \Omega^{-\frac{1}{2}}$, we have

$$\begin{aligned} \hat{\mathcal{A}} &= \begin{bmatrix} \alpha I & & \\ & \alpha I & \\ & & \beta I \end{bmatrix}^{-\frac{1}{2}} \begin{bmatrix} A & B^T & 0 \\ -B & 0 & -C^T \\ 0 & C & 0 \end{bmatrix} \begin{bmatrix} \alpha I & & \\ & \alpha I & \\ & & \beta I \end{bmatrix}^{\frac{1}{2}} \\ &= \begin{bmatrix} \alpha^{-1} A & \alpha^{-1} B^T & 0 \\ -\alpha^{-1} B & 0 & -(\alpha\beta)^{-\frac{1}{2}} C^T \\ 0 & (\alpha\beta)^{-\frac{1}{2}} C & 0 \end{bmatrix}. \end{aligned}$$

Since α and β are positive, then $\alpha^{-1}A$ is a symmetric positive definite matrix, $\alpha^{-1}B$ and $(\alpha\beta)^{-\frac{1}{2}}C$ are matrices of full row rank. Thus $\hat{\mathcal{A}}$ is a nonsymmetric positive semi-definite matrix. Moreover, it follows from Lemma 1 that $\operatorname{Re}(\lambda(\hat{\mathcal{A}})) > 0$. Therefore, it can be deduced from Lemma 2 that $\rho(\hat{\Gamma}_{GSS}) \leq 1$. The equality $\rho(\hat{\Gamma}_{GSS}) = 1$ holds if and only if $\hat{\mathcal{A}}$ has a (reducing) eigenvalue of the form $i\xi$, which is impossible because $\operatorname{Re}(\lambda(\hat{\mathcal{A}})) > 0$. It turns out that $\rho(\hat{\Gamma}_{GSS}) < 1$. As a result, $\rho(\Gamma_{GSS}) < 1$ which completes the proof.

3. GSS Preconditioner and Its Properties

Though the GSS iteration method is proved to converge unconditionally, its convergence can be rather slow when compared with the involved Krylov subspace methods [24]. In the context of linear system solvers, it is more interesting to exploit the generalized shift-splitting as a preconditioner to accelerate the Krylov subspace methods like GMRES.

In this section, we present the following generalized shift-splitting preconditioner (GSS)

$$\mathcal{P}_{GSS} = \frac{1}{2} \begin{bmatrix} \alpha I + A & B^T & 0 \\ -B & \alpha I & -C^T \\ 0 & C & \beta I \end{bmatrix}. \quad (7)$$

When applying a preconditioner, the convergence of the preconditioned linear system is closely related with the eigenvalue distribution. The following finding shows that the eigenvalues of GSS-preconditioned matrix $\mathcal{A}\mathcal{P}_{GSS}^{-1}$ cluster in a circle centered at the point $(1, 0)$ with radius less than 1. We adopt the

right preconditioning here such that the residual of the original saddle point problem is the same as that of its right-preconditioned counterpart.

Theorem 2. *Suppose that $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $B \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{s \times m}$ are matrices of full row rank. Let θ be an eigenvalue of $\mathcal{A}\mathcal{P}_{GSS}^{-1}$, then*

$$|1 - \theta| < 1.$$

Proof. Since the preconditioned matrix $\mathcal{A}\mathcal{P}_{GSS}^{-1}$ is similar to $\mathcal{P}_{GSS}^{-1}\mathcal{A}$, then it is only necessary to investigate the spectrum of $\mathcal{P}_{GSS}^{-1}\mathcal{A}$. Since $\mathcal{P}_{GSS} = M(\alpha, \beta)$, then it follows from (4) that

$$\begin{aligned} \mathcal{P}_{GSS}^{-1}\mathcal{A} &= \mathcal{P}_{GSS}^{-1}(M(\alpha, \beta) - N(\alpha, \beta)) \\ &= M(\alpha, \beta)^{-1}(M(\alpha, \beta) - N(\alpha, \beta)) \\ &= I - \Gamma_{GSS}. \end{aligned}$$

From Theorem 1, we know that $|1 - \theta| < 1$, where θ is an eigenvalue of $\mathcal{P}_{GSS}^{-1}\mathcal{A}$.

Before ending this section, we give some implementation details when applying the GSS preconditioner. In actual computations, a matrix-vector product $\mathcal{P}_{GSS}^{-1}r$ is often needed which can be solved from the linear system $\mathcal{P}_{GSS}z = r$. Note that the preconditioner \mathcal{P}_{GSS} can be decomposed as

$$\mathcal{P}_{GSS} = \frac{1}{2} \begin{bmatrix} I & B^T \left(\alpha I + \frac{1}{\beta} C^T C \right)^{-1} & 0 \\ 0 & I & -\frac{1}{\beta} C^T \\ 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} \alpha I + A + B^T \left(\alpha I + \frac{1}{\beta} C^T C \right)^{-1} B & 0 & 0 \\ 0 & \alpha I + \frac{1}{\beta} C^T C & 0 \\ 0 & 0 & \beta I \end{bmatrix} \quad (8)$$

$$\cdot \begin{bmatrix} I & 0 & 0 \\ -\left(\alpha I + \frac{1}{\beta} C^T C \right)^{-1} B & I & 0 \\ 0 & \frac{1}{\beta} C & I \end{bmatrix}.$$

Let $r = [r_1^T, r_2^T, r_3^T]^T$ and $z = [z_1^T, z_2^T, z_3^T]^T$, where $r_1, z_1 \in \mathbb{R}^n$, $r_2, z_2 \in \mathbb{R}^m$ and $r_3, z_3 \in \mathbb{R}^s$. By using the decomposition (8), we obtain the following sub-routine for solving $z = \mathcal{P}_{GSS}^{-1}r$.

- 1) Solve w_1 from $\left(\alpha I + \frac{1}{\beta} C^T C \right) w_1 = 2r_2 + \frac{2}{\beta} C^T r_3$;

- 2) Solve z_1 from $\left(\alpha I + A + B^T \left(\alpha I + \frac{1}{\beta} C^T C \right)^{-1} B \right) z_1 = 2r_1 - B^T w_1$;
- 3) Solve w_2 from $\left(\alpha I + \frac{1}{\beta} C^T C \right) w_2 = B z_1$;
- 4) Update $z_2 = w_1 + w_2$;
- 5) Update $z_3 = \frac{1}{\beta} (2r_3 - C z_2)$.

In computing $\mathcal{P}_{GSS}^{-1} r$, the computational overhead by GSS is the same as that by SS [22]. As indicated in the numerical examples, however, an appropriate choice of (α, β) generally yields better numerical performance (concerning CPU time and iteration steps) than using a single parameter α (as in SS). In this sense, the GSS can be regarded as a more promising variant than its predecessor SS.

4. Numerical Experiments

In this section, we give some numerical experiments to illustrate the effectiveness of the new preconditioner GSS in terms of the number of iteration steps (iter), CPU time in seconds (time) and the relative residual norm (res). For completeness, we display both the number of restarts and the number of inner steps in the last restart; see, for instance, **Table 3**. The unconditioned GMRES(k), the shift-splitting-preconditioned GMRES(k) (SS) [22] and the relaxed shift-splitting-preconditioned GMRES(k) (RSS) [22] are employed for comparison, where k is the restarting frequency. In what follows, the right preconditioning is used for SS, RSS and GSS, the reason of which is given in Section 3. We set the restarting frequency k to be 5 and the initial guess u^0 to be the zero vector. The right-hand side vector b in (1) is given by $10 \cdot \text{rand}(m+n+s, 1)$. All algorithms mentioned above terminate if $\|b - Au^i\|_2 / \|b\|_2 < 10^{-6}$ within 1500 restarts, where u^i is the i th iterate. Numerical experiments are carried out on a laptop using MATLAB 2014b with Intel Core (8G RAM) under Windows 10 system.

The testing problem is adapted from [25]. It is a block three-by-three saddle point problem with A , B and C of the following structure

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

$$B = [I \otimes F \quad F \otimes I] \in \mathbb{R}^{p^2 \times 2p^2},$$

$$C = E \otimes F \in \mathbb{R}^{p^2 \times p^2},$$

where $T = \frac{1}{h^2} \cdot \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{p \times p}$, $F = \frac{1}{h} \cdot \text{tridiag}(0, 1, -1) \in \mathbb{R}^{p \times p}$,

$E = \text{diag}(1, p+1, \dots, p^2 - p + 1)$ with \otimes being the Kronecker product and

$h = \frac{1}{p+1}$ the grid size.

4.1. Choice of the Parameter Pair (α, β)

As mentioned in Section 2, the main motivation for embracing an extra parameter β in GSS is to strike a balance in controlling the sub-matrices in (3). A proper choice of α and β will make the GSS preconditioner more appealing. As shown in (7), the GSS preconditioner \mathcal{P}_{GSS} becomes closer to (half of) the saddle point matrix \mathcal{A} as α and β decrease. This implies that the choice $\alpha = \beta = 0$ seems to be the optimal one. However, we do not suggest using extremely small values of α and β for numerical concerns. Instead, we use the experimental (optimal) values of α and β in the following experiments.

To this end, we first tabulate the CPU time of the GSS preconditioner with varying α and β for $p=16$ and $p=32$ in **Table 1** and **Table 2**, respectively. The parameter pair $(\alpha, \beta) = (0.01, 0.001)$ yields the shortest CPU time in both tables. To further verify this finding, we carry out more experiments in **Figure 1** and **Figure 2** by depicting the number of total iteration steps¹ and CPU time of GSS with varying β when α is chosen to be 0.01, 0.1 and 1, respectively. Some remarks are in order. Loosely speaking, for a fixed value of α , the number of total iteration steps and CPU time increase as β goes up although there are some zigzags; see **Figure 1** and **Figure 2**. Moreover, a similar pattern is also observed for β when the value of β is fixed while α varies; for instance, when $\beta = 50$, the total iteration steps increase as α changes from 0.01 to 1, which reads readily by comparing the subplots (a), (c) and (e) in **Figure 1**. This justifies our choice of using $(\alpha, \beta) = (0.01, 0.001)$ as the experimental optimal value.

Based on the above experiments, we use $(\alpha, \beta) = (0.01, 0.001)$ as the experimental optimal choice for α and β in GSS. As for the optimal value of α

Table 1. CPU time of the GSS preconditioner with different values of α and β ($p=16$).

time β	α					
	0.001	0.01	0.1	1	5	10
0.001	0.0391	0.0306	0.0535	0.0704	0.0921	0.1283
0.01	0.0336	0.0550	0.0512	0.0854	0.1321	0.1473
0.1	0.0506	0.0515	0.0675	0.1122	0.1507	0.2250
1	0.0739	0.0612	0.0886	0.1484	0.2838	0.4180
5	0.1056	0.1041	0.1393	0.2813	0.7082	1.3993
10	0.1443	0.1560	0.1744	0.4298	0.9397	2.1133

¹By “total iteration steps”, we mean the total number of matrix-vector products associated with $\mathcal{A}\mathcal{P}_{GSS}^{-1}$; for instance, if “iter”, as in **Table 3**, is denoted by $10(3)$, then the number of total iteration steps will be $(10-1) \times 5 + 3 = 48$. In general, if “iter” is denoted by $i(j)$, then the number of total iteration steps will be $(i-1) \times 5 + j$.

in SS and RSS, we follow the choice $\alpha = 0.01$ used in [22] and regard it as the empirical optimal value.

4.2. Comparison with Other SS-Type Methods

In this subsection, we compare the numerical performance of GSS with SS and

Table 2. CPU time of the GSS preconditioner with different values of α and β ($p = 32$).

β	α					
	0.001	0.01	0.1	1	5	10
0.001	1.4348	1.3374	1.8127	2.5649	3.1475	4.3422
0.01	1.3786	2.1595	1.8248	2.7657	3.7720	4.5090
0.1	2.3747	2.0354	2.4876	3.1519	5.4760	7.0319
1	2.5129	2.5942	3.2291	5.2169	10.1217	15.6771
5	3.7049	3.8040	5.1990	9.5487	24.7657	47.8987
10	4.4818	4.4699	5.6668	13.9554	35.1132	74.5482

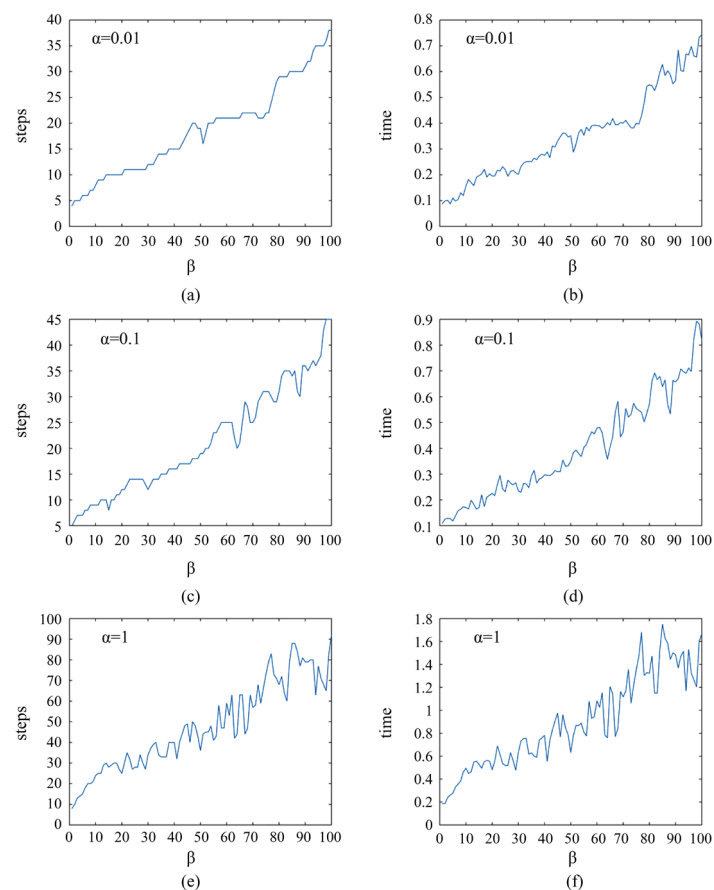


Figure 1. Total iteration steps (left) and CPU time (right) of the GSS preconditioner for different values of α and β ($p = 16$).

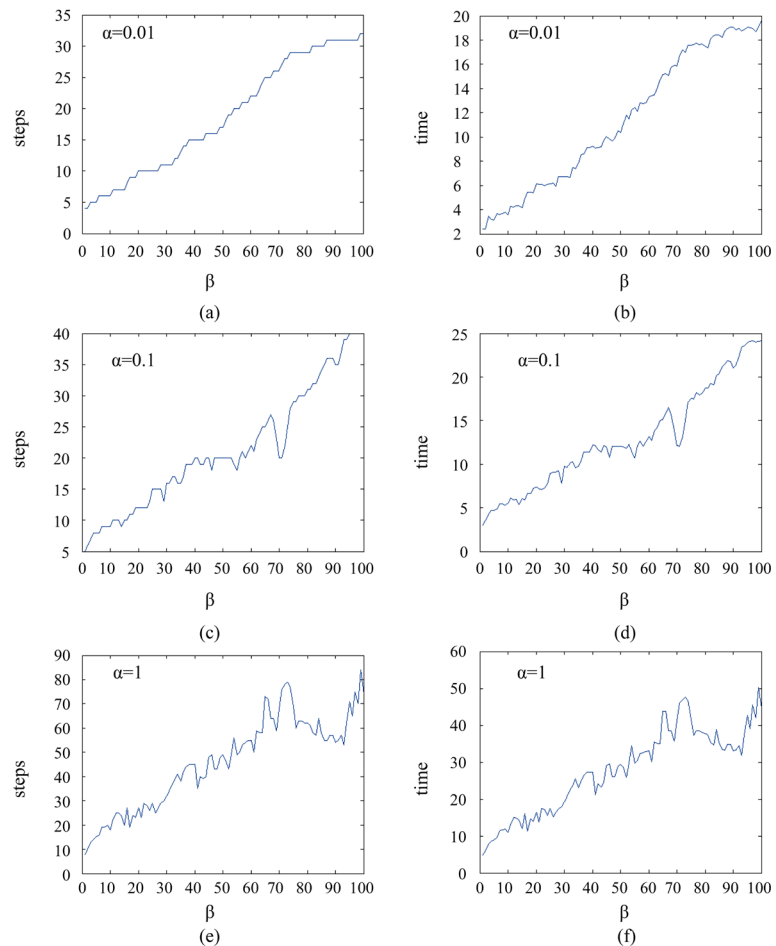


Figure 2. Total iteration steps (left) and CPU time (right) of the GSS preconditioner for different values of α and β ($p = 32$).

RSS regarding CPU time and iteration steps. It should be noted that all sub-linear equations involved in computing matrix-vector product $\mathcal{P}^{-1}r$ are solved by direct methods.

It is known that the spectrum imposes a great effect on the convergence of the preconditioned linear system. To begin with the discussion, we plot respectively the spectral distributions of the original saddle point matrix \mathcal{A} , the GSS-preconditioned matrix $\mathcal{A}\mathcal{P}_{GSS}^{-1}$, the SS-preconditioned matrix $\mathcal{A}\mathcal{P}_{SS}^{-1}$, and the RSS-preconditioned matrix $\mathcal{A}\mathcal{P}_{RSS}^{-1}$ in **Figure 3**. As illustrated, the eigenvalues of the original saddle point matrix \mathcal{A} are well separated, which makes the unpreconditioned GMRES method cumbersome. After employing the SS-type preconditioners, however, the eigenvalues of the preconditioned matrices become much more clustered. In particular, the eigenvalues of the GSS-preconditioned matrix are more clustered than those of SS and RSS; see **Figure 3**. Thus it can be expected that GSS will bring about an improvement over SS and RSS in solving (1). This intuition is echoed by **Table 3**, where the saddle point problem (1) of four different sizes are considered. In [22], it is stated that the iteration steps of both the SS and RSS preconditioned iteration methods remain constant

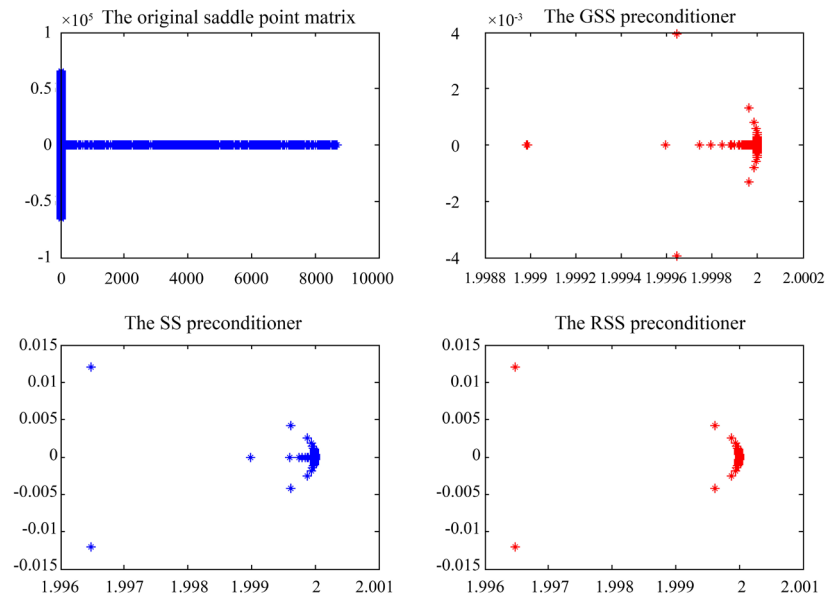


Figure 3. Eigenvalue distributions of the unpreconditioned saddle point matrix and of three SS-type preconditioned matrices ($p = 32$).

Table 3. Numerical results for three preconditioned GMRES methods with different grid sizes.

size	SS			RSS			GSS		
	iter	time	res	iter	time	res	iter	time	res
$p = 16$	1(3)	0.0744	9.3431e-09	1(3)	0.0651	5.1792e-09	1(2)	0.0415	4.5483e-07
$p = 32$	1(3)	1.9894	8.5844e-09	1(3)	1.8084	4.4832e-09	1(2)	1.2381	3.4408e-07
$p = 48$	1(3)	16.5567	7.8697e-09	1(3)	16.2467	2.7591e-09	1(2)	10.9178	3.1121e-07
$p = 64$	1(3)	88.9495	7.6631e-09	1(3)	84.8754	2.641e-09	1(2)	57.3213	2.7896e-07

even if the problem size $4p^2$ soars. Fortunately, this desirable property has also been carried over to GSS. Furthermore, GSS outperforms SS and RSS in that it requires the shortest CPU time and the least number of iteration steps in all test problems, as observed in **Table 3**. In light of this, we conclude that GSS is a competitive shift-splitting variant for solving the structured saddle point problem (1).

5. Conclusion

The shift-splitting iteration method/preconditioner for solving the block three-by-three saddle point problem (1) uses a single parameter to control the convergence. Being aware of the different roles of the block diagonal matrices in the saddle point matrix, we introduce an additional parameter to monitor the $(3,3)$ -block, which yields a generalized shift-splitting iteration method and a corresponding auxiliary preconditioner. By looking into the spectrum of the iteration matrix, we prove the unconditional convergence of the proposed method. We also apply the induced generalized shift-splitting preconditioner to speed up the

convergence of GMRES. It is proved that the preconditioned matrix using the new preconditioner has a well-clustered spectrum which is verified by the numerical examples. Future works include finding the theoretically optimal choice of the parameters in the new scheme and develop other efficient shift-splitting preconditioners.

Acknowledgements

This work is supported by the National Natural Science Foundation (grant number: 11601323) and the Key Discipline Fund (2018) of College of Arts and Sciences in Shanghai Maritime University. The authors would like to thank the referees for their constructive suggestions.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Benzi, M., Golub, G.H. and Liesen, J. (2005) Numerical Solution of Saddle Point Problems. *Acta Numerica*, **14**, 1-137. <https://doi.org/10.1017/S0962492904000212>
- [2] Arrow, K.J., Hurwicz, L. and Uzawa, H. (1958) Studies in Nonlinear Programming. Stanford University Press, Redwood City, CA.
- [3] Bai, Z.-Z., Golub, G.H. and Ng, M.K. (2003) Hermitian and Skew-Hermitian Splitting Methods for Non-Hermitian Positive Definite Linear Systems. *SIAM Journal on Matrix Analysis and Applications*, **24**, 603-626. <https://doi.org/10.1137/S0895479801395458>
- [4] Bai, Z.-Z. (2015) Motivations and Realizations of Krylov Subspace Methods for Large Sparse Linear Systems. *Journal of Computational and Applied Mathematics*, **283**, 71-78. <https://doi.org/10.1016/j.cam.2015.01.025>
- [5] Bramble, J.H., Pasciak, J.E. and Vassilev, A.T. (1997) Analysis of the Inexact Uzawa Algorithm for Saddle Point Problems. *SIAM Journal on Numerical Analysis*, **34**, 1072-1092. <https://doi.org/10.1137/S0036142994273343>
- [6] Cao, Z.-H. (2003) Fast Uzawa Algorithm for Generalized Saddle Point Problems. *Applied Numerical Mathematics*, **46**, 157-171. [https://doi.org/10.1016/S0168-9274\(03\)00023-0](https://doi.org/10.1016/S0168-9274(03)00023-0)
- [7] Bai, Z.-Z. and Wang, Z.-Q. (2008) On Parameterized Inexact Uzawa Methods for Generalized Saddle Point Problems. *Linear Algebra and Its Applications*, **428**, 2900-2932. <https://doi.org/10.1016/j.laa.2008.01.018>
- [8] Zhang, J. and Shang, J. (2010) A Class of Uzawa-SOR Methods for Saddle Point Problems. *Applied Mathematics and Computation*, **216**, 2163-2168. <https://doi.org/10.1016/j.amc.2010.03.051>
- [9] Cao, Y. and Yi, S.-C. (2016) A Class of Uzawa-PSS Iteration Methods for Nonsingular and Singular Non-Hermitian Saddle Point Problems. *Applied Mathematics and Computation*, **275**, 41-49. <https://doi.org/10.1016/j.amc.2015.11.049>
- [10] Bai, Z.-Z. and Golub, G.H. (2007) Accelerated Hermitian and Skew-Hermitian Splitting Iteration Methods for Saddle-Point Problems. *IMA Journal of Numerical Analysis*, **27**, 1-23. <https://doi.org/10.1093/imanum/drl017>

- [11] Bai, Z.-Z., Benzi, M. and Chen, F. (2010) Modified HSS Iteration Methods for a Class of Complex Symmetric Linear Systems. *Computing*, **87**, 93-111. <https://doi.org/10.1007/s00607-010-0077-0>
- [12] Cao, Y., Yao, L., Jiang, M. and Niu, Q. (2013) A Relaxed HSS Preconditioner for Saddle Point Problems from Meshfree Discretization. *Journal of Computational and Applied Mathematics*, **31**, 398-421. <https://doi.org/10.4208/jcm.1304-m4209>
- [13] Pan, J.-Y., Ng, M.K. and Bai, Z.-Z. (2006) New Preconditioners for Saddle Point Problems. *Applied Mathematics and Computation*, **172**, 762-771. <https://doi.org/10.1016/j.amc.2004.11.016>
- [14] Benzi, M. and Guo, X.-P. (2011) A Dimensional Split Preconditioner for Stokes and Linearized Navier-Stokes Equations. *Applied Numerical Mathematics*, **61**, 66-76. <https://doi.org/10.1016/j.apnum.2010.08.005>
- [15] Zhang, G.-F., Ren, Z.-R. and Zhou, Y.-Y. (2011) On HSS-Based Constraint Preconditioners for Generalized Saddle-Point Problems. *Numerical Algorithms*, **57**, 273-287. <https://doi.org/10.1007/s11075-010-9428-3>
- [16] Wang, N.-N. and Li, J.-C. (2019) A Class of New Extended Shift-Splitting Preconditioners for Saddle Point Problems. *Journal of Computational and Applied Mathematics*, **357**, 123-145. <https://doi.org/10.1016/j.cam.2019.02.015>
- [17] Rozložník, M. (2018) Saddle-Point Problems and Their Iterative Solution. Birkhäuser. <https://doi.org/10.1007/978-3-030-01431-5>
- [18] Bai, Z.-Z., Yin, J.-F. and Su, Y.-F. (2006) A Shift-Splitting Preconditioner for Non-Hermitian Positive Definite Matrices. *Journal of Computational and Applied Mathematics*, **24**, 539-552.
- [19] Cao, Y., Du, J. and Niu, Q. (2014) Shift-Splitting Preconditioners for Saddle Point Problems. *Journal of Computational and Applied Mathematics*, **272**, 239-250. <https://doi.org/10.1016/j.cam.2014.05.017>
- [20] Cao, Y., Tao, H. and Jiang, M. (2014) Generalized Shift Splitting Preconditioners for Saddle Point Problems, in Chinese. *Mathematica Numerica Sinica*, **36**, 16-26.
- [21] Salkuyeh, D.K., Masoudi, M. and Hezari, D. (2015) On the Generalized Shift-Splitting Preconditioner for Saddle Point Problems. *Applied Mathematics Letters*, **48**, 55-61. <https://doi.org/10.1016/j.aml.2015.02.026>
- [22] Cao, Y. (2019) Shift-Splitting Preconditioners for a Class of Block Three-by-Three Saddle Point Problems. *Applied Mathematics Letters*, **96**, 40-46. <https://doi.org/10.1016/j.aml.2019.04.006>
- [23] Bai, Z.-Z. and Hadjidimos, A. (2014) Optimization of Extrapolated Cayley Transform with Non-Hermitian Positive Definite Matrix. *Linear Algebra and Its Applications*, **463**, 322-339. <https://doi.org/10.1016/j.laa.2014.08.021>
- [24] Zhang, K., Zhang, J.-L. and Gu, C.-Q. (2017) A New Relaxed PSS Preconditioner for Nonsymmetric Saddle Point Problems. *Applied Mathematics and Computation*, **308**, 115-129. <https://doi.org/10.1016/j.amc.2017.03.022>
- [25] Huang, N. and Ma, C.-F. (2019) Spectral Analysis of the Preconditioned System for the 3×3 Block Saddle Point Problem. *Numerical Algorithms*, **81**, 421-444. <https://doi.org/10.1007/s11075-018-0555-6>