Journal of Applied Analysis and Computation
Volume 13, Number 6, December 2023, 3221–3230
Website:http://www.jaac-online.com
DOI:10.11948/20220552

THE SHSS PRECONDITIONER FOR SADDLE POINT PROBLEMS

Cuixia Li[1] and Shiliang Wu[1,2,†]

Abstract In this paper, building on the previous published work by Li and Wu [Appl. Math, Lett., 2015, 44, 26–29], we extend the single-step HSS (SHSS) method for saddle point problems. Based on the idea of SHSS method, the SHSS preconditioner for solving saddle point problems is introduced. We discuss the spectral properties of the preconditioned matrix in detail. By some numerical experiments, we demonstrate the effectiveness of the SHSS preconditioner.

Keywords Single-step HSS method, saddle point problems, convergence, preconditioning, eigenvalue.

MSC(2010) 65F10, 65F15.

1. Introduction

To efficiently solve the following non-Hermitian positive definite linear systems

$$Ax = b, \tag{1.1}$$

where b is a given vector, and A is a given matrix with non-Hermitian and positive definite (its the Hermitian part, $H = \frac{1}{2}(A + A^*)$, is positive definite), in [11], Li and Wu presented a single-step HSS (SHSS) iteration method for solving the non-Hermitian positive definite linear systems (1.1), which was described below

$$(\alpha I + H)x^{(k+1)} = (\alpha I - S)x^{(k)} + b, \quad k = 0, 1, \ldots, \tag{1.2}$$

where $H = \frac{1}{2}(A + A^*)$, and $S = \frac{1}{2}(A - A^*)$ is the skew-Hermitian part of A, and $\alpha > 0$. Whereafter, the SHSS iteration method was employed to solve the complex symmetric linear systems [18,19,21]. Clearly, comparing the following HSS iteration method in [2]

$$\begin{cases} (\alpha I + H)x^{(k+\frac{1}{2})} = (\alpha I - S)x^{(k)} + b, \\ (\alpha I + S)x^{(k+1)} = (\alpha I - H)x^{(k+\frac{1}{2})} + b, \end{cases} \quad k = 0, 1, \ldots, \tag{1.3}$$

the SHSS iteration method (1.2) has certain advantages. Concretely, the SHSS iteration method (1.2) successfully avoids a shifted skew-Hermitian linear subsystem

†The corresponding author.
[1]School of Mathematics, Yunnan Normal University, Kunming, Yunnan 650500, China
[2]Yunnan Key Laboratory of Modern Analytical Mathematics and Applications, Yunnan Normal University, Kunming, Yunnan 650500, China
Email: lixiatk@126.com(C. Li), slwuynnu@126.com(S. Wu)

with coefficient matrix $\alpha I + S$. As is known, the coefficient matrix $\alpha I + S$ of linear systems in (1.3) is skew-Hermitian, usually, its solution is difficult to gain [3].

In this paper, we will extend the SHSS iteration method for the following classical saddle point problems

$$\mathcal{A}\begin{bmatrix} x \\ y \end{bmatrix} \equiv \begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ q \end{bmatrix} \equiv f, \tag{1.4}$$

where $B \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}(B) = m < n$ and $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite. In this way, matrix $\mathcal{A}$ is nonsingular. This implies that the saddle point problems (1.4) has a unique solution. As is known, the saddle point problems (1.4) occurs in many different applications of engineering and scientific computing, see [4–10, 12–15, 22]. When the SHSS iteration method is used to solve the saddle point problems (1.4), in theory, we present the convergence conditions of the SHSS iteration method. In particular, we propose the SHSS preconditioner with Krylov method (such as GMRES) for the saddle point problems (1.4). Numerical experiments are shown to demonstrate the effectiveness of the SHSS preconditioner.

2. The SHSS preconditioner

Based on the HSS iteration method in [2], matrix $\mathcal{A}$ can be constructed as

$$\mathcal{A} = (\alpha I + H) - (\alpha I - S) = \begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix} - \begin{bmatrix} \alpha I & -B^T \\ B & \alpha I \end{bmatrix}, \tag{2.1}$$

where $\alpha > 0$ and I denotes the corresponding dimension identity matrix. From the matrix splitting (2.1), naturally, the SHSS iteration method is constructed to solve the saddle point problems (1.4) and is described below.

The SHSS iteration method: Assume that the initial vectors $x^{(0)} \in \mathbb{R}^m$ and $y^{(0)} \in \mathbb{R}^n$ are arbitrarily given, for $k = 0, 1, 2, ...,$ until the sequence of iterations $\{x^{(k)}, y^{(k)}\}_{k=0}^{+\infty}$ is convergent, calculate

$$\begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix}\begin{bmatrix} x^{(k+1)} \\ y^{(k+1)} \end{bmatrix} = \begin{bmatrix} \alpha I & -B^T \\ B & \alpha I \end{bmatrix}\begin{bmatrix} x^{(k)} \\ y^{(k)} \end{bmatrix} + \begin{bmatrix} b \\ q \end{bmatrix}, \tag{2.2}$$

where $\alpha > 0$.

Since matrix A is symmetric positive definite, the iteration matrix M_α of the SHSS method is

$$M_\alpha = \begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix}^{-1}\begin{bmatrix} \alpha I & -B^T \\ B & \alpha I \end{bmatrix}. \tag{2.3}$$

Clearly, the SHSS iteration method is convergent if and only if $\rho(M_\alpha) < 1$, where $\rho(M_\alpha)$ denotes the spectral radius of matrix M_α.

To study the convergence condition of the SHSS iteration method (2.2), let λ be an eigenvalue of matrix M_α in (2.3) and $[x, y]^T$ be the corresponding eigenvector, then we can obtain

$$\begin{bmatrix} \alpha I & -B^T \\ B & \alpha I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

i.e.,

$$\alpha x - B^T y = \lambda(\alpha I + A)x, \tag{2.4}$$

$$Bx + \alpha y = \alpha \lambda y. \tag{2.5}$$

Lemma 2.1. *Let λ be an eigenvalue of the matrix M_α. Then $\lambda \neq 1$.*

Proof. Assume that $\lambda = 1$, from (2.4) and (2.5) we have

$$\begin{cases} Ax + B^T y = 0, \\ -Bx = 0. \end{cases} \tag{2.6}$$

Then $y = 0$ and $x = 0$ because matrix $\mathcal{A}$ is nonsingular. This contradicts with $[x, y]^T \neq 0$. Hence $\lambda \neq 1$. □

Lemma 2.2. *If λ is an eigenvalue of M_α and $[x, y]^T$ is the corresponding eigenvector, then $x \neq 0$. Moreover, if $y = 0$, then $0 < \lambda < 1$.*

Proof. Assume that $x = 0$, from (2.4) we have $B^T y = 0$. Based on $\text{rank}(B) = m$, we get $y = 0$. Clearly, this is contradictory with $[x, y]^T \neq 0$. This implies that $x \neq 0$.

When $y = 0$, from (2.4) we have

$$\lambda \alpha x + \lambda A x = \alpha x. \tag{2.7}$$

For both sides of Eq. (2.7) by multiplying x^*, we can get

$$\lambda = \frac{\alpha x^* x}{\alpha x^* x + x^* A x} = \frac{\alpha}{\alpha + \frac{x^* A x}{x^* x}}.$$

Since $\frac{x^* A x}{x^* x} > 0$. Thus, $0 < \lambda < 1$. □

To obtain our main result, the following lemma is required.

Lemma 2.3 ([16]). *Let $x^2 - bx + d = 0$, where $b, d \in \mathbb{R}$, and λ denote the root of this equation. Then $|\lambda| < 1$ if and only if $|d| < 1$ and $|b| < 1 + d$.*

Based on Lemma 2.3, the following result is obtained.

Theorem 2.1. *Let*

$$a = x^* A x, \ x^* B^T B x = b, \ \|x\|_2 = 1.$$

If $b < \alpha a$, then

$$\rho(M_\alpha) < 1,$$

from which the SHSS iteration method (2.2) *converges to the unique solution of the saddle point problem* (1.4)*.*

Proof. From Lemma 2.1, together with (2.5), we have

$$y = \frac{Bx}{\alpha(\lambda - 1)}. \tag{2.8}$$

Substituting (2.8) into (2.4) leads to

$$(\lambda - 1)\alpha x + \lambda A x + \frac{B^T B x}{\alpha(\lambda - 1)} = 0. \tag{2.9}$$

Here, without loss of generality, we take $\|x\|_2 = 1$. For both sides of Eq. (2.9) by multiplying x^*, we can obtain

$$\alpha^2(\lambda - 1)^2 + \alpha\lambda(\lambda - 1)x^* A x + x^* B^T B x = 0,$$

or,

$$\alpha^2(\lambda - 1)^2 + \alpha(\lambda^2 - \lambda)a + b = 0, \tag{2.10}$$

where $a = x^* A x > 0$ and $b = x^* B^T B x \geq 0$. Further, from (2.10), we have

$$\lambda^2 - \frac{2\alpha^2 + \alpha a}{\alpha^2 + \alpha a}\lambda + \frac{\alpha^2 + b}{\alpha^2 + \alpha a} = 0. \tag{2.11}$$

Based on Lemma 2.3, $|\lambda| < 1$ if and only if

$$\left|\frac{\alpha^2 + b}{\alpha^2 + \alpha a}\right| < 1 \tag{2.12}$$

and

$$\left|\frac{2\alpha^2 + \alpha a}{\alpha^2 + \alpha a}\right| < 1 + \frac{\alpha^2 + b}{\alpha^2 + \alpha a}. \tag{2.13}$$

When $b > 0$, it is easy to confirm that (2.12) and (2.13) is valid for $b < \alpha a$. When $b = 0$, there exist a nonzero vector x such that $Bx = 0$. Based on (2.5), we get $y = 0$. So, based on Lemma 2.2 we obtain $0 < \lambda < 1$. Hence, $\rho(M_\alpha) < 1$. □

In fact, if $\lambda_{\min}$ is the smallest eigenvalue of matrix A and $\sigma_{\max}$ is the largest singular-value of matrix B, then we have

$$\frac{x^* B^T B x}{x^* A x} \leq \frac{\sigma_{\max}^2}{\lambda_{\min}}. \tag{2.14}$$

Based on Eq. (2.14), we have the following corollary.

Corollary 2.1. *Let $\lambda_{\min}$ be the smallest eigenvalue of matrix A and $\sigma_{\max}$ be the largest singular-value of matrix B. If $\frac{\sigma_{\max}^2}{\lambda_{\min}} < \alpha$, then*

$$\rho(M_\alpha) < 1,$$

from which the SHSS iteration method (2.2) *converges to the unique solution of the saddle point problems* (1.4).

Further, let $y = A^{\frac{1}{2}} x$ and $\|y\|_2 = 1$. Then

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

Therefore, we have the following result as well.

Corollary 2.2. *Let $\bar{\sigma}_{\max}$ be the largest singular-value of matrix $BA^{-\frac{1}{2}}$. If $\bar{\sigma}_{\max} < \alpha$, then*

$$\rho(M_\alpha) < 1,$$

from which the SHSS iteration method (2.2) *converges to the unique solution of the saddle point problems* (1.4).

Clearly, the SHSS iteration method (2.2) for the saddle point problems (1.4) is convergent when Theorem 2.1, Corollaries 2.1 or 2.2 are satisfied. Whereas, in general, the convergence of this stationary iteration method is too slow such that it is not competitive. For this reason, based on the matrix splitting (2.1), the splitting matrix $P_H = \alpha I + H$ can be used as a preconditioner matrix for the matrix $\mathcal{A}$. This is to say, we can make use of the Krylov subspace method with the preconditioner $P_H = \alpha I + H$ to solve the saddle point problems (1.4). In general, the spectral distribution of the preconditioned matrix effects the convergence behavior of the preconditioned Krylov subspace methods. Therefore, we here need to establish the spectral distribution of the preconditioned matrix $P_H^{-1}\mathcal{A}$.

With regard to the spectral distribution of $P_H^{-1}\mathcal{A}$, the following theorem is obtained.

Theorem 2.2. *Let the condition of Theorem 2.1 be satisfied. Then the absolute value of all the eigenvalues of $P_H^{-1}\mathcal{A}$ is less than one.*

Proof. Assume that λ is the eigenvalue of $P_H^{-1}\mathcal{A}$ and $[x, y]^T$ is the corresponding eigenvector. Then we have

$$\begin{bmatrix} A & B^T \\ -B & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

or equivalently,

$$Ax + B^T y = \lambda(\alpha I + A)x, \tag{2.15}$$
$$-Bx = \alpha\lambda y. \tag{2.16}$$

Note that $\lambda \neq 0$. Then from (2.16) we have

$$y = \frac{-Bx}{\alpha\lambda}. \tag{2.17}$$

Substituting (2.17) into (2.15) leads to

$$Ax - \frac{B^T Bx}{\alpha\lambda} = \lambda(\alpha I + A)x. \tag{2.18}$$

Here, without loss of generality, we take $\|x\|_2 = 1$. For both sides of Eq. (2.18) by multiplying x^*, we can obtain

$$\lambda^2 - \frac{\alpha a}{\alpha^2 + \alpha a}\lambda + \frac{b}{\alpha^2 + \alpha a} = 0, \tag{2.19}$$

where $a = x^* Ax > 0$ and $b = x^* B^T Bx \geq 0$. Based on Theorem 2.1, it is easy to see that $|\lambda| < 1$. □

3. Numerical experiments

In this section, we report some numerical experiments to demonstrate the performance of the preconditioner P_H. To illustrate the advantage of the preconditioner P_H, in our numerical computations, we compare the preconditioner P_H with the well-known HSS preconditioner in [2]. Here, P_{HSS} denotes the HSS preconditioner and is form

$$P_{HSS} = \frac{1}{2\alpha}(\alpha I + H)(\alpha I + S) = \frac{1}{2\alpha}\begin{bmatrix} \alpha I + A & 0 \\ 0 & \alpha I \end{bmatrix}\begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix}.$$

In test example, we employ the restarted GMRES(#) for the corresponding saddle point-type systems (1.4), where f is adjusted such that its solution is $(1, 1, \ldots, 1)^T$.

In our numerical experiments, the initial guess is the zero vector, all iterations are stopped when the numbers of iteration steps surpass 500 or the current iterates satisfy

$$\|f - \mathcal{A}x^{(k)}\|_2 \leq 10^{-6}\|f\|_2.$$

All the computations are done with MATLAB R2016b. In the following tables, 'IT' denotes the numbers of iteration steps, 'CPU' denotes the CPU times in second and 'Res' denotes the relative residual. Additionally, α^* denotes the optimal experimental parameter, i.e., under the optimal experimental parameter α^*, our testing preconditioners have the mini numbers of iteration steps and the mini relative residual; '$-$' fails to converge in 500 iterations and seconds.

Table 1. CPU and IT for P_H and P_{HSS} with $p = 60$.

	α	0.1	0.5	1	1.5
P_H	IT	30	28	28	26
	CPU	0.5664	0.5342	0.5375	0.4829
	Res	6.65e-7	8.25e-7	2.92e-7	9.77e-7
P_{HSS}	IT	22	48	54	56
	CPU	2.3186	4.6283	5.1663	5.3502
	Res	9.83e-7	9.46e-7	8.39e-7	9.29e-7

Table 2. CPU and IT for P_H and P_{HSS} with $p = 60$ and α^*.

	α^*	IT	CPU	Res
P_H	3.3	26	0.4485	8.85e-7
P_{HSS}	1e-04	2	0.4468	2.18e-7

Example 3.1. [16] Consider the saddle point problems (1.4), in which

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

Table 3. CPU and IT for P_H and P_{HSS} with $p = 80$.

	α	0.1	0.5	1	1.5
P_H	IT	32	30	28	28
	CPU	1.4612	1.3509	1.2605	1.2381
	Res	9.99e-7	4.24e-7	7.13e-7	5.31e-7
P_{HSS}	IT	21	52	56	58
	CPU	5.1357	12.3287	13.3923	13.3558
	Res	9.24e-7	9.77e-7	9.76e-7	9.81e-7

Table 4. CPU and IT for P_H and P_{HSS} with $p = 80$ and α^*.

	α^*	IT	CPU	Res
P_H	2.8	28	1.1992	4.85e-7
P_{HSS}	1e-04	2	0.9136	1.65e-07

and

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

where $\otimes$ stands for the Kronecker product symbol and $h = \frac{1}{p+1}$ stands for the mesh-size.

For the sake of simply, in our computations, we take # as 30. The numerical results on IT, CPU and Res of GMRES(30) with P_H and P_{HSS} are presented in Tables 1, 2, 3 and 4. From Tables 1, 2, 3 and 4, the preconditioner P_H needs less CPU times than the preconditioner P_{HSS}. This implies that the preconditioner P_H outperforms the preconditioner P_{HSS} from the view of the computational efficiency under certain conditions.

Example 3.2. [20] Consider the saddle point problems (1.4), in which

$$A = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2\times 2p^2}, \ B = \begin{pmatrix} \bar{B}, b_1, b_2 \end{pmatrix} \in \mathbb{R}^{2p^2\times(p^2+2)},$$

where

$$T = \frac{\nu}{h^2} \cdot \mathrm{tridiag}(-1, 2, -1) + \frac{1}{2h} \cdot \mathrm{tridiag}(-1, 0, 1) \in \mathbb{R}^{p\times p},$$

$$\bar{B} = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \mathbb{R}^{2p^2\times p^2},$$

$$b_1 = \bar{B} \begin{pmatrix} e \\ 0 \end{pmatrix}, \ b_2 = \bar{B} \begin{pmatrix} 0 \\ e \end{pmatrix}, \ e = (1, 1, \ldots, 1) \in \mathbb{R}^{p^2/2},$$

$$F = \frac{1}{h} \cdot \mathrm{tridiag}(-1, 1, 0) \in \mathbb{R}^{p\times p}, \ h = \frac{1}{p+1},$$

where $\otimes$ stands for the Kronecker product symbol and $h = \frac{1}{p+1}$ stands for the mesh-size.

Table 5. CPU and IT for P_H and P_{HSS} with $p = 60$.

	α	0.01	0.05	0.1	0.5
P_H	IT	25	25	25	41
	CPU	1.1764	1.2426	1.2553	1.9034
	Res	8.37e-7	8.41e-7	8.70e-7	8.74e-7
P_{HSS}	IT	10	18	23	58
	CPU	75.083	94.752	124.77	290.41
	Res	8.36e-7	9.65e-7	9.65e-7	9.73e-7

Table 6. CPU and IT for P_H and P_{HSS} with $p = 60$ and α^*.

	α^*	IT	CPU	Res
P_H	1e-04	25	1.3532	8.36e-7
P_{HSS}	1e-04	2	20.2368	2.41e-07

Table 7. CPU and IT for P_H and P_{HSS} with $p = 80$.

	α	0.001	0.005	0.01	0.05
P_H	IT	25	25	25	25
	CPU	2.4857	2.4917	2.5608	3.1111
	Res	7.76	7.77	7.77	7.81
P_{HSS}	IT	4	7	9	–
	CPU	174.68	257.09	270.78	–
	Res	6.02	9.73	9.01	–

Table 8. CPU and IT for P_H and P_{HSS} with $p = 80$ and α^*.

	α^*	IT	CPU	Res
P_H	1e-04	25	2.4194	7.76e-7
P_{HSS}	1e-04	2	99.8795	1.84e-07

In our computations, we take $\nu = 1$ for Example 3.2. Similarly, we still take # as 30. Tables 5, 6, 7 and 8 list some numerical results on IT, CPU and Res of GMRES(30) with P_H and P_{HSS}. From Tables 5, 6, 7 and 8, these numerical results still confirm that the preconditioner P_H requires less CPU times than the preconditioner P_{HSS}. This further shows that the preconditioner P_H outperforms the preconditioner P_{HSS} from the view of the computational efficiency under certain conditions.

4. Conclusion

In this paper, a SHSS preconditioner has been introduced to solve the classical saddle point problems. The convergence properties of the SHSS method are discussed and the spectral properties of the preconditioned matrix are presented. Numerical examples confirm the effectiveness of the SHSS preconditioner.

Acknowledgements

This research of this author is supported by the National Natural Science Foundation of China (11961082).

The authors would like to express their great thankfulness to the referees and editor for your much constructive, detailed and helpful advice regarding revising this manuscript.

References

[1] Z. Bai and G. H. Golub, *Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems*, IMA Journal of Numerical Analysis, 2007, 27, 1–23.

[2] Z. Bai, G. H. Golub and M. K. Ng, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM Journal on Matrix Analysis and Applications, 2003, 24, 603–626.

[3] M. Benzi, *A generalization of the Hermitian and skew-Hermitian splitting iteration*, SIAM Journal on Matrix Analysis and Applications, 2009, 31, 360–374.

[4] M. Benzi, G. H. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 2005, 14, 1–137.

[5] S. Bradley and C. Greif, *Eigenvalue bounds for saddle-point systems with singular leading blocks*, Journal of Computational and Applied Mathematics, 2023, 424, 114996.

[6] S. Cafieri, M. D′Apuzzo, V. De Simone and D. Di Serafino, *On the iterative solution of KKT systems in potential reduction software for large-scale quadratic problems*, Computational Optimization and Applications, 2007, 38, 27–45.

[7] H. C. Elman, *Preconditioning for the steady-state Navier-Stokes equations with low viscosity*, Siam Journal on Scientific Computing, 1999, 20, 1299–1316.

[8] C. Greif and D. Schötzau, *Preconditioners for the discretized time-harmonic Maxwell equaitons in mixed form*, Numerical Linear Algebra with Applications, 2007, 14, 281–297.

[9] C. Greif and D. Schötzau, *Preconditioners for saddle point linear systems with highly singular (1,1) blocks*, Electronic Transactions on Numerical Analysis, 2006, 22, 114–121.

[10] A. Hadjidimos and M. Tzoumas, *On equivalence of three-parameter iterative methods for singular symmetric saddle-point problem*, Numerical Algorithms, 2021, 86, 1391–1419.

[11] C. Li and S. Wu, *A single-step HSS method for non-Hermitian positive definite linear systems*, Applied Mathematics Letters, 2015, 44, 26–29.

[12] T. Rees and C. Greif, *A preconditioner for linear systems arising from interior point optimization methods*, Siam Journal on Scientific Computing, 2007, 29, 1992–2007.

[13] D. K. Salkuyeh, *Shifted skew-symmetric/skew-symmetric splitting method and its application to generalized saddle point problems*, Applied Mathematics Letters, 2020, 103, 106184.

[14] J. Scott and M. Tuma, *A null-space approach for large-scale symmetric saddle point systems with a small and non zero (2, 2) block*, Numerical Algorithms, 2022, 90, 1639–1667.

[15] S. Vakili, G. Ebadi and C. Vuik, *A parameterized extended shift-splitting preconditioner for nonsymmetric saddle point problems*, IMA Journal of Numerical Analysis, 2023, 30, e2478.

[16] S. Wu, T. Huang and X. Zhao, *A modified SSOR iterative method for augmented systems*, Journal of Computational and Applied Mathematics, 2009, 228, 424–433.

[17] S. Wu and C. Li, *A splitting method for complex symmetric indefinite linear system*, Journal of Computational and Applied Mathematics, 2017, 313, 343–354.

[18] X. Xiao and X. Wang, *A new single-step iteration method for solving complex symmetric linear systems*, Numerical Algorithms, 2018, 78, 643–660.

[19] X. Xiao, X. Wang and H. Yin, *Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems*, Computers and Mathematics with Applications, 2017, 74, 2269–2280.

[20] A. Yang, X. Li and Y. Wu, *On semi-convergence of the Uzawa-HSS method for singular saddle-point problems*, Applied Mathematics and Computation, 2015, 252, 88–98.

[21] M. Zeng and C. Ma, *A parameterized SHSS iteration method for a class of complex symmetric system of linear equations*, Computers and Mathematics with Applications, 2016, 71, 2124–2131.

[22] G. Zilli and L. Bergamaschi, *Block preconditioners for linear systems in interior point methods for convex constrained optimization*, Annali Dell'universita Di Ferrara, 2022, 68, 337–368.