

# Accelerated RHSS Iteration Method for Stabilized Saddle-Point Problems

Zhengkui Song, Pingping Zhang

School of Science, Chongqing University of Posts and Telecommunications, Chongqing, China

Email: cquptszh@foxmail.com

**How to cite this paper:** Song, Z.H. and Zhang, P.P. (2022) Accelerated RHSS Iteration Method for Stabilized Saddle-Point Problems. *Journal of Applied Mathematics and Physics*, 10, 1019-1027.  
<https://doi.org/10.4236/jamp.2022.104069>

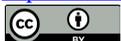
**Received:** March 6, 2022

**Accepted:** April 3, 2022

**Published:** April 6, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing 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

For stabilized saddle-point problems, we apply the two iteration parameters idea for regularized Hermitian and skew-Hermitian splitting (RHSS) method and establish accelerated RHSS (ARHSS) iteration method. Theoretical analysis shows that the ARHSS method converges unconditionally to the unique solution of the saddle point problem. Finally, we use a numerical example to confirm the effectiveness of the method.

## Keywords

Stabilized Saddle-Point Problems, Regularized Hermitian and Skew-Hermitian Splitting, Iteration Parameters, Convergence Property

## 1. Introduction

Consider the stabilized saddle-point problem

$$Ax \equiv \begin{pmatrix} B & E \\ -E^* & C \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \equiv b, \quad (1)$$

where  $B \in \mathbb{C}^{p \times p}$  is a Hermitian positive definite matrix,  $C \in \mathbb{C}^{q \times q}$  is a Hermitian positive semidefinite matrix,  $E \in \mathbb{C}^{p \times q}$  is rectangular matrix,  $E^* \in \mathbb{C}^{q \times p}$  is the complex conjugate transpose of  $E$  and  $f \in \mathbb{C}^p$  and  $g \in \mathbb{C}^q$ . In addition, the null spaces of the matrices  $C$  and  $E$  do not overlap, i.e.,  $\text{null}(C) \cap \text{null}(E) = \{0\}$ , then in accordance with [1], we know that the stabilized saddle-point problem (1) admits a unique solution; see also [2] and its reference therein. For stabilized saddle-point problem, it has a broad background in scientific computing and engineering applications. For example, it frequently appears in Navier-Stokes equations, finite element methods, regularized and weighted least-squares problems. For more details, see [3] [4] [5] [6] [7].

When  $C = 0$ , the stabilized saddle-point problem (1) degenerated into stan-

standard saddle-point problem. Benzi and Golub [8] proposed the Hermitian and skew-Hermitian splitting (HSS) iteration method to solve the standard saddle-point problem. This iteration method is a direct generalization of the HSS iteration method initially proposed in [9]. To improve the convergence rate of the HSS iteration method, Bai, Golub and Pan [10] proposed the preconditioned HSS (PHSS) iteration method, and then Bai and Golub [11] proposed the accelerated HSS (AHSS) iteration method by calculating the HSS iteration sequence with different parameters. Bai and Benzi [12] proposed the regularized Hermitian and skew-Hermitian splitting (RHSS) iteration method to solve the standard saddle-point problem. The biggest highlight is that the condition number of iterative systems can be improved by regularizing the matrix. In addition, Behzadi and Abdollahi [13] combined the AHSS iteration method and the RHSS iteration method to build the accelerated RHSS (ARHSS) iteration method for the standard saddle-point linear system, and experimental results show that the ARHSS iteration method is better than the HSS iteration method and the RHSS iteration method.

Recently, based on the regularized Hermitian and skew-Hermitian splitting of the stabilized saddle-point matrix  $A$  in (1):

$$\begin{aligned} A &= \begin{pmatrix} B & 0 \\ 0 & Q_1 + \omega C \end{pmatrix} + \begin{pmatrix} 0 & E \\ -E^* & -Q_1 + (1-\omega)C \end{pmatrix} = H_+(\omega) + S_-(\omega) \\ &= \begin{pmatrix} 0 & E \\ -E^* & Q_1 + (1+\omega)C \end{pmatrix} + \begin{pmatrix} B & 0 \\ 0 & -Q_1 - \omega C \end{pmatrix} = S_+(\omega) + H_-(\omega), \end{aligned} \quad (2)$$

where  $Q_1$  is a Hermitian positive semidefinite matrix and  $\omega$  is a non-negative constant. Bai [2] extended the RHSS iteration method for the standard saddle-point problem to the stabilized saddle-point problem as follows:

$$\begin{cases} (\alpha I + H_+(\omega))x^{(k+\frac{1}{2})} = (\alpha I - S_-(\omega))x^{(k)} + b, \\ (\alpha I + S_+(\omega))x^{(k+1)} = (\alpha I - H_-(\omega))x^{(k+\frac{1}{2})} + b, \end{cases} \quad (3)$$

where  $I$  is the identity matrix and  $\alpha$  is a positive constant. And Bai proved the RHSS iteration method converges unconditionally to the exact solution of the stabilized saddle-point problem (1). Numerical results on stabilized saddle-point problems in [2] show that the RHSS iteration method significantly outperforms the Hermitian and skew-Hermitian splitting iteration method in iteration counts and computing times.

In this paper, inspired by Behzadi and Abdollahi [13] and Bai [2], we know that the accelerated RHSS iteration method has advantages over the HSS iteration method and the RHSS iteration method on the standard saddle-point problem. So we extend the accelerated RHSS iteration method for the standard saddle-point problem to the stabilized saddle-point problem, and establish the ARHSS iteration method for the stabilized saddle-point problem. Then we confirm the convergence of the ARHSS iteration method. Finally, experimental re-

sults show that the ARHSS iteration method is more effective than the HSS iteration method and the RHSS iteration method for the stabilized saddle-point problem (1).

The remainder of this work is organized as follows. In Section 2, we propose the ARHSS iteration method for the stabilized saddle-point problem (1). In Section 3, we give the convergence property of the ARHSS iteration method. In Section 4, A numerical experiment is given to show the effectiveness of the method. In Section 5, we end the paper with some brief conclusions.

## 2. The ARHSS Iteration Method

In this section, we derive the ARHSS iteration method for solving (1). Let  $Q \in \mathbb{C}^{q \times q}$  be a Hermitian positive semidefinite matrix and  $\omega$  be a prescribed non-negative parameter. Then we can split the stabilized saddle-point matrix  $A \in \mathbb{C}^{n \times n}$  in (1), and obtain the regularized Hermitian and skew-Hermitian (RHS) splitting:

$$\begin{aligned} A &= \begin{pmatrix} B & 0 \\ 0 & Q + \omega C \end{pmatrix} + \begin{pmatrix} 0 & E \\ -E^* & -Q + (1 - \omega)C \end{pmatrix} = H_+ + S_- \\ &= \begin{pmatrix} 0 & E \\ -E^* & Q + (1 + \omega)C \end{pmatrix} + \begin{pmatrix} B & 0 \\ 0 & -Q - \omega C \end{pmatrix} = S_+ + H_- \end{aligned} \quad (4)$$

By applying the regularized Hermitian and Skew-Hermitian (RHS) splitting to (4), we then obtain the iteration scheme

$$\begin{cases} (\Lambda + H_+)x = (\Lambda - S_-)x + b \\ (\Lambda + S_+)x = (\Lambda - H_-)x + b \end{cases}$$

where

$$\Lambda = \begin{pmatrix} \alpha I & 0 \\ 0 & \beta I \end{pmatrix}$$

with  $\alpha$  and  $\beta$  positive constants.

By iterating alternatively between these two fixed-point systems as

$$(\Lambda + H_+)x^{(k+1/2)} = (\Lambda - S_-)x^{(k)} + b,$$

and

$$(\Lambda + S_+)x^{(k+1)} = (\Lambda - H_-)x^{(k+1/2)} + b,$$

or in their blockwise forms

$$\begin{aligned} &\begin{pmatrix} \alpha I + B & 0 \\ 0 & \beta I + Q + \omega C \end{pmatrix} \begin{pmatrix} y^{(k+1/2)} \\ z^{(k+1/2)} \end{pmatrix} \\ &= \begin{pmatrix} \alpha I & -E \\ E^* & \beta I + Q - (1 - \omega)C \end{pmatrix} \begin{pmatrix} y^{(k)} \\ z^{(k)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix}, \end{aligned} \quad (5)$$

and

$$\begin{aligned} & \begin{pmatrix} \alpha I & E \\ -E^* & \beta I + Q + (1 + \omega)C \end{pmatrix} \begin{pmatrix} y^{(k+1)} \\ z^{(k+1)} \end{pmatrix} \\ &= \begin{pmatrix} \alpha I - B & 0 \\ 0 & \beta I + Q + \omega C \end{pmatrix} \begin{pmatrix} y^{(k+1/2)} \\ z^{(k+1/2)} \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix}. \end{aligned} \tag{6}$$

So, we obtain the ARHSS iteration method for solving the stabilized saddle-point problem (1) as follows:

**The ARHSS iteration method.** Let  $\alpha$  and  $\beta$  be positive constants,  $\omega$  be a non-negative constant and  $Q \in \mathbb{C}^{p \times p}$  be a Hermitian positive semidefinite matrix. Give an initial guess  $x^{(0)} = (y^{(0)*}, z^{(0)*})^* \in \mathbb{C}^n$ , for  $k = 0, 1, 2, \dots$  until the iteration sequence  $\{x^{(k)}\} = \left\{ \left( y^{(k)*}, z^{(k)*} \right)^* \right\} \subset \mathbb{C}^n$  converges, compute the next iterate  $x^{(k+1)} = (y^{(k+1)*}, z^{(k+1)*})^* \in \mathbb{C}^n$  according to following procedure:

(i) solve for  $y^{(k+1/2)} \in \mathbb{C}^p$  from the linear subsystem

$$(\alpha I + B)y^{(k+1/2)} = \alpha Iy^{(k)} - Ez^{(k)} + f;$$

(ii) compute

$$f^{(k+1/2)} = (\alpha I - B)y^{(k+1/2)} + f;$$

and

$$g^{(k+1/2)} = (\beta I + Q + (\omega - 1)C)z^{(k)} + E^*y^{(k)} + 2g;$$

(iii) solve for  $z^{(k+1)} \in \mathbb{C}^q$  from the linear subsystem

$$\left( \beta I + Q + (1 + \omega)C + \frac{1}{\alpha}E^*E \right) z^{(k+1)} = \frac{1}{\alpha}E^*f^{(k+1/2)} + g^{(k+1/2)};$$

(iv) compute

$$y^{(k+1)} = \frac{1}{\alpha} \left( -Ez^{(k+1)} + f^{(k+1/2)} \right).$$

### 3. The Convergence Property of the ARHSS Iteration Method

In this section we prove the unconditional convergence of the ARHSS iteration method. We know that calculating the spectral radius of the iterative matrix  $L(\alpha, \beta, \omega)$  is the easiest way to prove the convergence of ARHSS iteration method. From (5) and (6), we easily know  $L(\alpha, \beta, \omega)$  can be equivalently rewritten as

$$L(\alpha, \beta, \omega) = M(\alpha, \beta, \omega)^{-1}N(\alpha, \beta, \omega),$$

where

$$M(\alpha, \beta, \omega) = \frac{1}{2} \begin{pmatrix} \alpha I + B & \frac{1}{\alpha}(\alpha I + B)E \\ -E^* & \beta I + Q + (\omega + 1)C \end{pmatrix}$$

and

$$N(\alpha, \beta, \omega) = \frac{1}{2} \begin{pmatrix} \alpha I - B & -\frac{1}{\alpha}(\alpha I - B)E \\ E^* & \beta I + Q + (\omega - 1)C \end{pmatrix}$$

To prove the convergence of the ARHSS iteration method, we first introduce a lemma.

**Lemma 3.1.** ([2], Theorem 3.1) *For the stabilized saddle-point problem (1), the RHSS iteration method (3) converges unconditionally to the exact solution of the stabilized saddle-point problem (1).*

In the following theorem, we prove the unconditional convergence of the ARHSS iteration method.

**Theorem 3.2.** *For the stabilized saddle-point problem (1), assume that  $B \in \mathbb{C}^{p \times p}$  is Hermitian positive definite,  $C \in \mathbb{C}^{q \times q}$  is Hermitian positive semidefinite and  $E \in \mathbb{C}^{p \times q}$  satisfies  $\text{null}(C) \cap \text{null}(E) = 0$ . Let  $\alpha, \beta > 0$  be prescribed positive constant,  $\omega$  be a given non-negative constant and  $Q \in \mathbb{C}^{q \times q}$  be a given Hermitian positive semidefinite matrix. Then it holds that  $\rho(L(\alpha, \beta, \omega)) < 1$ , i.e., the ARHSS iteration method converges unconditionally to the exact solution of the stabilized saddle-point problem (1).*

*Proof.* There are two cases to consider.

Case I: If  $\beta \geq \alpha$ , according to the ARHS splitting, we obtain

$$\begin{aligned} A &= (\Lambda + H_+) - (\Lambda - S_-) \\ &= \left( \alpha I + \begin{pmatrix} B & 0 \\ 0 & \tilde{Q} + \omega C \end{pmatrix} \right) - \left( \alpha I - \begin{pmatrix} 0 & E \\ -E^* & -\tilde{Q} + (1 - \omega)C \end{pmatrix} \right) \end{aligned} \quad (7)$$

and

$$\begin{aligned} A &= (\Lambda + S_+) - (\Lambda - H_-) \\ &= \left( \alpha I + \begin{pmatrix} 0 & E \\ -E^* & \tilde{Q} + (1 + \omega)C \end{pmatrix} \right) - \left( \alpha I - \begin{pmatrix} B & 0 \\ 0 & -\tilde{Q} - \omega C \end{pmatrix} \right) \end{aligned} \quad (8)$$

where  $\tilde{Q} = (\beta - \alpha)I + Q$ . Since  $\tilde{Q}$  is Hermitian positive semidefinite matrix, relations (7) and (8) are exactly the RHS splitting of  $A$ . By Lemma 3.1, this split converges to the exact solution of the saddle point linear system (1).

Case II: If  $\beta < \alpha$ , according to the ARHS splitting, we obtain

$$\begin{aligned} A &= (\Lambda + H_+) - (\Lambda - S_-) \\ &= \left( \beta I + \begin{pmatrix} B & 0 \\ 0 & \tilde{Q} + \omega C \end{pmatrix} \right) - \left( \beta I - \begin{pmatrix} 0 & E \\ -E^* & -\tilde{Q} + (1 - \omega)C \end{pmatrix} \right) \end{aligned} \quad (9)$$

and

$$\begin{aligned} A &= (\Lambda + S_+) - (\Lambda - H_-) \\ &= \left( \beta I + \begin{pmatrix} 0 & E \\ -E^* & \tilde{Q} + (1 + \omega)C \end{pmatrix} \right) - \left( \beta I - \begin{pmatrix} B & 0 \\ 0 & -\tilde{Q} - \omega C \end{pmatrix} \right) \end{aligned} \quad (10)$$

where  $\tilde{Q} = (\alpha - \beta)I + Q$ . Since  $\tilde{Q}$  is Hermitian positive semidefinite matrix,

the relations (9) and (10) are exactly the RHS splitting of  $A$ . By Lemma 3.1, this split converges to the exact solution of the saddle point linear system (1). The desired result holds by Case I and Case II. Therefore, we complete the proof.

### 4. Numerical Experiment

In this section, a numerical example is given to illustrate the effectiveness of the ARHSS iteration method for the stabilized saddle-point problem (1). We compare the number of iteration steps (denoted as IT) and computing times in seconds (denoted as CPU) of the ARHSS iteration method with the HSS iteration method and the RHSS iteration method. All our tests are started from the initial vector  $x^{(0)} = 0$ . All iterations are terminated as the current residuals satisfy  $\|b - Ax^{(k)}\| \leq 10^{-6} \times \|b\|$  or the number of iterations steps exceeds 5000. In addition, all the computations were implemented in MATLAB [version 9.1.0.441655 (R2016b)] in double precision on a personal computer with 1.60 GHZ central processing unit [Intel(R) Core(TM)i5-5250U] and 4.00 GB memory.

**Example 4.1.** ([2], Example 4.3) *Consider the nonlinear image restoration problem*

$$\min \|\tilde{f} - s(Ky)\|_2^2 + \beta \|y\|_2^2, \tag{11}$$

where  $\tilde{f}$  and  $y \in \mathbb{C}^p$  represent the observed and the original images, respectively,  $s : \mathbb{C}^p \rightarrow \mathbb{C}$  denotes a nonlinear point spread function and  $K = ([K]_{ij}) \in \mathbb{C}^{p \times p}$  is a blurring matrix. When this regularized nonlinear least-squares problem (11) is solved by the Gauss-Newton iteration method, at each step for the currently available approximant  $y_c = ([y_c]_1, [y_c]_2, \dots, [y_c]_p)$ , we need to solve a linear system of the form

$$(\beta I + K^T D^2 K) y = K^T D [\tilde{f} - s(Ky_c) + DKy_c]$$

to obtain the next approximant, where  $D = \text{diag}([D]_1, [D]_2, \dots, [D]_p) \in \mathbb{C}^{p \times p}$  is a diagonal matrix with its diagonal entries being given by

$$[D]_i = \frac{\partial s}{\partial \xi} \Big|_{\xi = \sum_{j=1}^p [K]_{ij} [y_c]_j}, \quad i = 1, 2, \dots, p.$$

This linear system can be equivalently reformulated into the stabilized saddle-point problem (1), in which

$$B = D^{-2}, \quad E = K, \quad C = \beta I$$

and

$$f = D^{-1} [\tilde{f} - s(Ky_c)] + Ky_c, \quad g = 0.$$

In actual computations we set

$$\tilde{f} = \left[ \frac{254}{p} : \frac{254}{p} : 254 \right]^T$$

and

$$y_c = \left[ \left[ 0.5 + \frac{508}{p} : \frac{508}{p} : 254.5 \right], \left[ 254.5 : -\frac{508}{p} : 0.5 + \frac{508}{p} \right] \right]^T$$

and take  $\beta = 10^{-3}$ ,

$$s(\xi) = 30 \log(\xi), \quad [K]_{ij} = \frac{1}{\sqrt{2\pi\mu}} \exp\left(\frac{-|i-j|^2}{2\mu^2}\right), \quad i, j = 1, 2, \dots, p,$$

with  $\mu = 2$ . Here, the notation “:” is a standard MATLAB symbol used in indicating a row vector of the form

$$[x_f : \delta x : x_l] = [x_f, x_f + \delta x, \dots, x_f + k\delta x, \dots, x_l],$$

for which  $x_f$  and  $x_l$  are the first and the last elements,  $\delta x$  is a prescribed increment such that  $x_l - x_f$  is its multiple and  $k$  is a non-negative integer ranging from 0 to  $\frac{x_l - x_f}{\delta x}$ .

For this example we have  $p = q$ , so that the dimension of the stabilized saddle-point matrix  $A \in \mathbb{B}^{p \times p}$  is  $n = 2p$ . Note that the Toeplitz matrix  $E$  is highly ill-conditioned with rapidly decaying singular values so that it is almost rank-deficient, especially when the problem size  $p$  becomes sufficiently large. Besides, we take the regularization matrix to be

$$Q = (\alpha\gamma - \omega)C + \gamma E^T E.$$

Taking  $Q$  into (5) and (6), we get

$$g^{(k+1/2)} = (\beta I + (\alpha\gamma - 1)C + \gamma E^* E) z^{(k)} + E^* y^{(k)} + 2g;$$

and

$$\left( \beta I + (\alpha\gamma + 1)C + \frac{1}{\alpha} E^* E \right) z^{(k+1)} = \frac{1}{\alpha} E^* f^{(k+1/2)} + g^{(k+1/2)}.$$

So, in the specific algorithms, the parameter  $\omega$  does not need to be determined. In our experiments, let the parameter  $\gamma$  be taken as the optimal value of [[2], Example 4.3], and the optimal values of the parameters  $\alpha$  and  $\beta$  can be determined by experience and trial runs. For example, first, suppose the ranges of  $\alpha$  and  $\beta$  are  $(0, 1)$ , and then divide it into 100 equal parts. Second, fix parameter  $\alpha$  to a certain value in  $(0, 1)$ , and then find the corresponding optimal parameter  $\beta$  by trial runs. Third, using the second step method, fix parameter  $\beta$  to a certain value in  $(0, 1)$ , and then find the corresponding optimal parameter  $\alpha$  by trial runs. Finally, we choose the best parameters  $\alpha$  and  $\beta$  through experience and trial runs.

In **Table 1**, we list the best parameters  $\alpha, \beta$  and regularization parameter  $\gamma$  for the three iteration methods including HSS, RHSS and ARHSS with different mesh number  $p$ . Through these optimal parameters, we list the experimental results of the three methods in **Table 2**. The experimental results show that compared with HSS and RHSS, the ARHSS iteration method takes the least CPU and the smallest IT to converge to the solution of (11).

**Table 1.** Optimal parameter and regularization parameter.

$p$		512	1024	1536	2048
HSS	$\alpha$	1	2	4	5.2
	$\gamma$	0.2	0.18	0.15	0.11
RHSS	$\alpha$	1	1	1	0.98
	$\gamma$	0.2	0.18	0.15	0.11
ARHSS	$\alpha$	1	1	1	1
	$\beta$	0.9	0.9	0.9	0.96
	$\gamma$	0.2	0.18	0.15	0.11

**Table 2.** IT and CPU.

$p$		512	1024	1536	2048
HSS	IT	623	806	1216	1651
	CPU	7.360713	17.068340	36.843068	61.9992798
RHSS	IT	747	717	691	555
	CPU	2.804141	7.434071	10.852548	12.2928
ARHSS	IT	659	551	609	535
	CPU	2.532601	5.497928	9.359197	11.253577

## 5. Conclusion

For stabilized saddle-point problem, we extend the accelerated RHSS iteration method for the standard saddle-point problem to the stabilized saddle-point problem. The convergence property of the ARHSS iteration method is carefully studied, and the numerical result illustrates that the ARHSS is more effective for the stabilized saddle-point problem. In this paper, optimal iteration parameters can only be determined experimentally. So, how to establish practical formulas for computing the optimal iteration parameters is worth further study.

## Conflicts of Interest

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

## References

- [1] Bai, Z.-Z. (2019) Regularized HSS iteration Methods for Stabilized Saddle-Point Problems. *IMA Journal of Numerical Analysis*, **39**, 1888-1923. <https://doi.org/10.1093/imanum/dry046>
- [2] Wathen, A.J. and Silvester, D.J. (1993) Fast Iterative Solution of Stabilished Stokes Systems I: Using Simple Diagonal Preconditioners. *SIAM Journal on Numerical Analysis*, **30**, 630-649. <https://doi.org/10.1137/0730031>
- [3] Silvester, D.J. and Kechkar, N. (1990) Stabilised Bilinear-Constant Velocity-Pressure Finite Elements for the Conjugate Gradient Solution of the Stokes Problems. *Computer Methods in Applied Mechanics and Engineering*, **79**, 71-86.

- [https://doi.org/10.1016/0045-7825\(90\)90095-4](https://doi.org/10.1016/0045-7825(90)90095-4)
- [4] Elman, H.C., Silvester, D.J. and Wathen, A.J. (2002) Performance and Analysis of Saddle Point Preconditioners for the Discrete Steady-State Navier-Stokes Equations. *Numerische Mathematik*, **90**, 665-688. <https://doi.org/10.1007/s002110100300>
- [5] 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>
- [6] Bai, Z.-Z. (2006) Structured Preconditioners for Nonsingular Matrices of Block Two-by-Two Structures. *Mathematics of Computation*, **75**, 791-815. <https://doi.org/10.1090/S0025-5718-05-01801-6>
- [7] Wathen, A.J. (2015) Preconditioning. *Acta Numerica*, **124**, 329-376. <https://doi.org/10.1017/S0962492915000021>
- [8] Benzi, M. and Golub, G.H. (2004) A Preconditioner for Generalized Saddle Point Problems. *SIAM Journal on Matrix Analysis and Applications*, **26**, 20-41. <https://doi.org/10.1137/S0895479802417106>
- [9] 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>
- [10] Bai, Z.-Z., Golub, G.H. and Pan, J.-Y. (2004) Preconditioned Hermitian and Skew-Hermitian Splitting Methods for Non-Hermitian Positive Semidefinite Linear Systems. *Numerische Mathematik*, **98**, 1-32. <https://doi.org/10.1007/s00211-004-0521-1>
- [11] 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>
- [12] Bai, Z.-Z. and Benzi, M. (2017) Regularized HSS Iteration Methods for Saddle-Point Linear Systems. *BIT Numerical Mathematics*, **57**, 287-311. <https://doi.org/10.1007/s10543-016-0636-7>
- [13] Behzadi, R. and Abdollahi, F. (2018) Accelerated RHSS Iteration Methods for Saddle-Point Linear Systems. *UPB Scientific Bulletin, Series A*, **80**, 153-162.