Computers and Mathematics with Applications 74 (2017) 2269–2280

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Efficient single-step preconditioned HSS iteration methods for complex symmetric linear systems

Xiao-Yong Xiao, Xiang Wang *, Hong-Wei Yin

Department of Mathematics, School of Sciences, Nanchang University, Nanchang 330031, China
Numerical Simulation and High-Performance Computing Laboratory, School of Sciences, Nanchang University, Nanchang 330031, China

ARTICLE INFO

Article history:
Received 16 March 2017
Received in revised form 19 June 2017
Accepted 5 July 2017
Available online 21 July 2017

MSC 2010:
65F10
65F50

Keywords:
Complex symmetric linear system
Positive definite
HSS iteration
Spectral radius
Convergence analysis

ABSTRACT

We propose a single-step preconditioned variant of HSS (SPHSS) and an efficient parameterized SPHSS (PSPHSS) iteration method for solving a class of complex symmetric linear systems. Under suitable conditions, we analyze the convergence properties of the SPHSS and PSPHSS iteration methods. Theoretical analysis shows that the minimal upper bounds for the spectral radius of the SPHSS and PSPHSS iteration matrices are less than those of the SHSS and PSHSS iteration matrices when using the optimal parameters, respectively. Numerical results show that the PSPHSS iteration method has comparable advantage over several other iteration methods whether the experimental optimal parameters are used or not.

1. Introduction and motivations

We consider to solve the complex symmetric system of linear equations

$$Ax \equiv (W + iT)x = b, \tag{1.1}$$

with $W, T \in \mathbb{R}^{n\times n}$ being symmetric positive semi-definite, and without loss of generality, we suppose W is positive definite. Throughout the paper, the vector $b \in \mathbb{C}^n$ is given, $x \in \mathbb{C}^n$ is unknown, and $i = \sqrt{-1}$ is the imaginary unit. This kind of complex symmetric linear systems can be found in numerical analysis and applied scientific branches, we refer to [1–5] and the references therein.

To solve the problem (1.1) iteratively, a simple Hermitian and skew-Hermitian splitting (HSS) about the coefficient matrix is commonly used. It is given by

$$A = H + S,$$

where

$$H = \frac{1}{2}(A + A^*) = W \quad \text{and} \quad S = \frac{1}{2}(A - A^*) = iT.$$

* Corresponding author at: Department of Mathematics, School of Sciences, Nanchang University, Nanchang 330031, China.
E-mail address: wangxiang49@ncu.edu.cn (X. Wang).

http://dx.doi.org/10.1016/j.camwa.2017.07.007
0898-1221/

It is apparent that A is a non-Hermitian, but positive definite matrix. Here A^* denotes the conjugate transpose of the matrix A. Bai et al. [6] first proposed the HSS iteration method to find the solution of the system (1.1) by the scheme

$$\begin{cases} (\alpha I + W)x^{\left(k+\frac{1}{2}\right)} = (\alpha I - iT)x^{(k)} + b, \\ (\alpha I + iT)x^{(k+1)} = (\alpha I - W)x^{\left(k+\frac{1}{2}\right)} + b, \end{cases}$$

where $\alpha > 0$ is a given constant and I is the identity matrix. For more details about the HSS iteration method and its variants, we refer to [7–13].

Later, a modification of the HSS (MHSS) iteration scheme is introduced in [2]. To speed up its convergence rate, Bai et al. [14,15] proposed a preconditioned variant of the MHSS (PMHSS) iteration method and applied it to solve distributed control problems. The PMHSS iteration method is given by

$$\begin{cases} (\alpha V + W)x^{\left(k+\frac{1}{2}\right)} = (\alpha V - iT)x^{(k)} + b, \\ (\alpha V + T)x^{(k+1)} = (\alpha V + iW)x^{\left(k+\frac{1}{2}\right)} - ib, \end{cases}$$

where $V \in \mathbb{R}^{n\times n}$ is a prescribed symmetric positive definite matrix. Moreover, Zheng and Ma [16] introduced the accelerated PMHSS iteration methods for complex symmetric linear systems. Li et al. [17] developed a lopsided preconditioned variant of the MHSS iteration method when the real part of the coefficient matrix is dominant. Pour and Goughery [18] proposed a new HSS (NHSS) iteration for non-Hermitian positive-definite linear systems, and a parameterized variant of NHSS iteration method is developed in [19].

Notice that all the iteration schemes mentioned above are two-step methods. Recently, Li and Wu [20] introduced the following single-step HSS (SHSS) iteration method for non-Hermitian positive definite linear systems:

$$(\alpha I + W)x^{(k+1)} = (\alpha I - iT)x^{(k)} + b.$$

The SHSS iteration method can be regarded as a general case of the shift-splitting iteration method, which has been studied for non-Hermitian positive definite linear systems in [21], for saddle point problems in [22,23] and block two-by-two linear systems in [24]. Zeng and Ma [25] proposed a parameterized variant of the SHSS (PSHSS) iteration method, which is more efficient than the SHSS iteration from their numerical experiments. The PSHSS iteration method is given by

$$(\alpha I + \omega W + T)x^{(k+1)} = [\alpha I - i(\omega T - W)]x^{(k)} + (\omega - i)b,$$

where α and ω are given positive constants.

We mention that the HSS-like iteration methods for solving complex systems of linear equations have been extended in a number of literatures. We refer to [21,26–39].

On the other hand, let $x = u+iv$ and $b = p+iq$ with $u, v, p, q \in \mathbb{R}^n$, then the complex linear system (1.1) can be rewritten as 2-by-2 block real equivalent formulation

$$\begin{bmatrix} W & -T \\ T & W \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} p \\ q \end{bmatrix}, \tag{1.2}$$

where u, v are unknown real vectors. Chen and Ma [40] introduced the AOR-Uzawa iterative method to solve this real equivalent system. Salkuyeh et al. [41] proposed the following generalized successive overrelaxation (GSOR) iterative method

$$\begin{cases} Wu^{(k+1)} = (1-\alpha)Wu^{(k)} + \alpha T v^{(k)} + \alpha p, \\ W v^{(k+1)} = -\alpha T u^{(k+1)} + (1-\alpha)W v^{(k)} + \alpha q, \end{cases}$$

where $u^{(0)}$ and $v^{(0)}$ are the initial approximations of u and v, respectively. In order to accelerate the convergence rate of the GSOR method, Hezari et al. [42] developed a preconditioned variant of the GSOR (PGSOR) iterative method by the scheme

$$\begin{cases} (\omega W + T)u^{(k+1)} = (1-\alpha)(\omega W + T)u^{(k)} + \alpha(\omega T - W)v^{(k)} + \alpha(\omega p + q), \\ (\omega W + T)v^{(k+1)} = -\alpha(\omega T - W)u^{(k+1)} + (1-\alpha)(\omega W + T)v^{(k)} + \alpha(\omega q - p), \end{cases}$$

where α and ω are given positive constants.

In this paper, we want to make improvement on the PMHSS and the SHSS iteration methods.

This paper is organized as follows. In Section 2, two single-step preconditioned HSS iteration methods (SPHSS and PSPHSS) are proposed. In Section 3, the convergence properties of the SPHSS iteration method are analyzed, including the convergence conditions, the spectral radius of the iterative matrix and the quasi-optimal parameter. In Section 4, the convergence properties of the PSPHSS iteration method are considered. In Section 5, some numerical examples are presented to show the computational efficiencies of the SPHSS and PSPHSS iteration methods with comparisons with several other iteration methods. Finally, a brief conclusion is made in Section 6.

2. Single-step preconditioned HSS iteration methods

It has been shown in [20] that the SHSS iteration method is convergent to the unique solution of the linear system with a loose restriction on the parameter α. Notice that the first half step of the HSS iteration method is just the SHSS method, which inspires us to take out the first half step of the PMHSS iteration method to establish the following single-step preconditioned HSS (SPHSS) iteration method.

Algorithm 1. The SPHSS Iteration Method.
Given an initial guess $x^{(0)} \in \mathbb{C}^n$, for $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x^{(k)}\}$ converges, compute $x^{(k+1)}$ by the scheme

$$(\alpha V + W)x^{(k+1)} = (\alpha V - iT)x^{(k)} + b \tag{2.1}$$

where α is a given positive constant and $V \in \mathbb{R}^{n\times n}$ is a prescribed symmetric positive definite matrix.

It is obvious that the SHSS iteration method is a special case of $V = I$.
On the other hand, multiplying a parameter $\omega - i$ by the two sides of the original linear system (1.1) yields that

$$(\omega - i)Ax \equiv [(\omega W + T) + i(\omega T - W)]x = (\omega - i)b. \tag{2.2}$$

Let

$$\tilde{A} = (\omega - i)A, \qquad \tilde{W} = \omega W + T, \qquad \tilde{T} = \omega T - W, \qquad \tilde{b} = (\omega - i)b. \tag{2.3}$$

Then we can rewrite (2.2) as

$$\tilde{A}x \equiv (\tilde{W} + i\tilde{T})x = \tilde{b}.$$

By comparing it with the original linear system (1.1), we can easily obtain the following variant of SPHSS iteration method:

$$(\alpha V + \tilde{W})x^{(k+1)} = (\alpha V - i\tilde{T})x^{(k)} + \tilde{b}. \tag{2.4}$$

Substituting (2.3) into (2.4), we get the parameterized variant of the SPHSS (PSPHSS) iteration method.

Algorithm 2. The PSPHSS Iteration Method.
Given an initial guess $x^{(0)} \in \mathbb{C}^n$, for $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x^{(k)}\}$ converges, compute $x^{(k+1)}$ by the scheme

$$(\alpha V + \omega W + T)x^{(k+1)} = [\alpha V - i(\omega T - W)]x^{(k)} + (\omega - i)b, \tag{2.5}$$

where α, ω are given positive constants, and $V \in \mathbb{R}^{n\times n}$ is a prescribed symmetric positive definite matrix.

It is easy to see that the PSHSS iteration method is a special case of $V = I$.

3. Convergence analysis about the SPHSS iteration method

First, we rewrite the SPHSS iteration scheme (2.1) as

$$x^{(k+1)} = M^V(\alpha)x^{(k)} + N^V(\alpha)b, \quad k = 0, 1, 2, \ldots,$$

where

$$\begin{aligned} M^V(\alpha) &= (\alpha V + W)^{-1}(\alpha V - iT) = (\alpha I + V^{-1}W)^{-1}(\alpha I - iV^{-1}T), \\ N^V(\alpha) &= (\alpha V + W)^{-1}, \end{aligned} \tag{3.1}$$

with $V \in \mathbb{R}^{n\times n}$ being a prescribed symmetric positive definite matrix.
If we define

$$F^V(\alpha) = \alpha V + W, \quad \text{and} \quad G^V(\alpha) = \alpha V - iT,$$

then it holds that

$$A = F^V(\alpha) - G^V(\alpha), \quad \text{and} \quad M^V(\alpha) = F^V(\alpha)^{-1}G^V(\alpha).$$

Hence, the splitting matrix $F^V(\alpha)$ can be taken as a preconditioner for the complex symmetric matrix $A \in \mathbb{C}^{n\times n}$, which is referred as the SPHSS preconditioner.
In particular, if we take $V = W$, then

$$M^W(\alpha) = \frac{1}{\alpha + 1}(\alpha I - iW^{-1}T), \qquad N^W(\alpha) = \frac{1}{\alpha + 1}W^{-1},$$

and the SPHSS iteration scheme is now induced by the matrix splitting

$$A = F^W(\alpha) - G^W(\alpha)$$

with

$$F^W(\alpha) = (\alpha+1)W, \quad \text{and} \quad G^W(\alpha) = \alpha W - iT.$$

In order to explore the convergence properties of the SPHSS iteration method, we use the following notations

$$\begin{aligned} \tilde{\lambda}^V_{\min} &= \min_{\tilde{\lambda}_j \in sp(V^{-\frac{1}{2}} W V^{-\frac{1}{2}})} \{\tilde{\lambda}_j\}, \qquad \tilde{\sigma}^V_{\max} = \max_{\tilde{\sigma}_j \in sp(V^{-\frac{1}{2}} T V^{-\frac{1}{2}})} \{\tilde{\sigma}_j\}, \\ \tilde{\lambda}^V_{\max} &= \max_{\tilde{\lambda}_j \in sp(V^{-\frac{1}{2}} W V^{-\frac{1}{2}})} \{\tilde{\lambda}_j\}, \qquad \tilde{\sigma}^V_{\min} = \min_{\tilde{\sigma}_j \in sp(V^{-\frac{1}{2}} T V^{-\frac{1}{2}})} \{\tilde{\sigma}_j\}, \end{aligned} \tag{3.2}$$

where $sp(X)$ is the spectral set of the matrix X.

Theorem 1. *If $W, T \in \mathbb{R}^{n\times n}$ are symmetric positive semi-definite matrices with at least one of them being positive definite, then the spectral radius $\rho(M^V(\alpha))$ is bounded by*

$$\delta^V(\alpha) = \frac{\sqrt{\alpha^2 + (\tilde{\sigma}^V_{\max})^2}}{\alpha + \tilde{\lambda}^V_{\min}}. \tag{3.3}$$

Moreover,

(i) *if $\tilde{\lambda}^V_{\min} \geq \tilde{\sigma}^V_{\max}$, then $\delta^V(\alpha) < 1$ i.e. the iteration converges for any $\alpha > 0$;*

(ii) *if $\tilde{\lambda}^V_{\min} < \tilde{\sigma}^V_{\max}$, then $\delta^V(\alpha) < 1$ if and only if*

$$\alpha > \frac{(\tilde{\sigma}^V_{\max})^2 - (\tilde{\lambda}^V_{\min})^2}{2\tilde{\lambda}^V_{\min}}. \tag{3.4}$$

Proof. According to (3.1), we obtain

$$\begin{aligned} \rho(M^V(\alpha)) &= \rho((\alpha I + V^{-1}W)^{-1}(\alpha I - iV^{-1}T)) \\ &\leq \|(\alpha I + V^{-1}W)^{-1}(\alpha I - iV^{-1}T)\|_2 \\ &\leq \|(\alpha I + V^{-1}W)^{-1}\|_2 \|(\alpha I - iV^{-1}T)\|_2 \\ &= \left(\max_{\tilde{\lambda}_j \in sp(V^{-\frac{1}{2}} W V^{-\frac{1}{2}})} \frac{1}{\alpha + \tilde{\lambda}_j}\right)\left(\max_{\tilde{\sigma}_j \in sp(V^{-\frac{1}{2}} T V^{-\frac{1}{2}})} \sqrt{\alpha^2 + \tilde{\sigma}_j^2}\right) \\ &= \frac{\sqrt{\alpha^2 + (\tilde{\sigma}^V_{\max})^2}}{\alpha + \tilde{\lambda}^V_{\min}}. \end{aligned}$$

Therefore, $\tilde{\lambda}^V_{\min} \geq \tilde{\sigma}^V_{\max}$ definitely leads to $\delta^V(\alpha) < 1$, and if $\tilde{\lambda}^V_{\min} < \tilde{\sigma}^V_{\max}$, then $\delta^V(\alpha) < 1$ if and only if

$$\alpha^2 + (\tilde{\sigma}^V_{\max})^2 < (\alpha + \tilde{\lambda}^V_{\min})^2,$$

which is equivalent to (3.4). □

Corollary 1. *Under the conditions of Theorem 1, the optimal parameter α^V_* which minimizes the upper bound $\delta^V(\alpha)$ of the spectral radius $\rho(M^V(\alpha))$ is given by*

$$\alpha^V_* = \operatorname*{argmin}_{\alpha} \left\{ \frac{\sqrt{\alpha^2 + (\tilde{\sigma}^V_{\max})^2}}{\alpha + \tilde{\lambda}^V_{\min}} \right\} = \frac{(\tilde{\sigma}^V_{\max})^2}{\tilde{\lambda}^V_{\min}}, \tag{3.5}$$

and

$$\delta^V(\alpha^V_*) = \frac{\tilde{\sigma}^V_{\max}}{\sqrt{(\tilde{\lambda}^V_{\min})^2 + (\tilde{\sigma}^V_{\max})^2}}. \tag{3.6}$$

Proof. By simple calculations, we get

$$\delta^V(\alpha)' = \frac{\alpha\tilde{\lambda}^V_{\min} - (\tilde{\sigma}^V_{\max})^2}{(\alpha + \tilde{\lambda}^V_{\min})^2\sqrt{\alpha^2 + (\tilde{\sigma}^V_{\max})^2}}.$$

Therefore, the upper bound $\delta^V(\alpha)$ of the spectral radius $\rho(M^V(\alpha))$ achieves its minimum at $\alpha^V_* = (\tilde{\sigma}^V_{\max})^2/\tilde{\lambda}^V_{\min}$, i.e., (3.5) holds true. Taking α^V_* into $\delta^V(\alpha)$, we obtain the minimum value of $\delta^V(\alpha)$ which is given by (3.6). □

Moreover, according to (3.6), when $\tilde{\lambda}^V_{\min} \geq \tilde{\sigma}^V_{\max}$, if we take the optimal parameter α^V_*, then it holds that

$$\rho(M^V(\alpha^V_*)) \leq \delta^V(\alpha^V_*) \leq \frac{1}{\sqrt{2}}.$$

In particular, when $V = W$, the SPHSS iteration is given by

$$(\alpha + 1)Wx^{(k+1)} = (\alpha W - iT)x^{(k)} + b,$$

and we have $\tilde{\lambda}^W_{\min} = 1$, thus the right hand sides of (3.3)–(3.6) are related only with $\tilde{\sigma}^W_{\max}$, and we have

$$\delta^W(\alpha^W_*) = \frac{\tilde{\sigma}^W_{\max}}{\sqrt{1 + (\tilde{\sigma}^W_{\max})^2}}. \tag{3.7}$$

Remark 1. When $V = I$, the SPHSS iteration is reduced to the SHSS iteration, and the minimal upper bound $\delta^I(\alpha)$ of the spectral radius $\rho(M^I(\alpha))$ is given by

$$\delta^I(\alpha^I_*) = \frac{\tilde{\sigma}^I_{\max}}{\sqrt{(\tilde{\lambda}^I_{\min})^2 + (\tilde{\sigma}^I_{\max})^2}} = \frac{\frac{\tilde{\sigma}^I_{\max}}{\tilde{\lambda}^I_{\min}}}{\sqrt{1 + \left(\frac{\tilde{\sigma}^I_{\max}}{\tilde{\lambda}^I_{\min}}\right)^2}}. \tag{3.8}$$

From (3.7) and (3.8), using the fact that $\|W^{-1}T\|_2 \leq \|W^{-1}\|_2\|T\|_2$, we obtain

$$\delta^W(\alpha^W_*) \leq \delta^I(\alpha^I_*),$$

since $\tilde{\sigma}^W_{\max} \leq \frac{\tilde{\sigma}^I_{\max}}{\tilde{\lambda}^I_{\min}}$ and $f(x) \equiv \frac{x}{\sqrt{1+x^2}}$ is a strictly monotonic increasing function. Therefore, the SPHSS iteration using $V = W$ is more efficient than the SHSS iteration when the optimal parameters are used.

Remark 2. According to Theorem 1, the SPHSS iteration would always converge if the parameter α is properly chosen. Moreover, from (3.6), theoretically the optimal symmetric positive definite matrix V_* minimizing $\delta^V(\alpha^V_*)$ is given by

$$V_* = \operatorname*{argmin}_V \left\{ \frac{\tilde{\sigma}^V_{\max}}{\sqrt{(\tilde{\lambda}^V_{\min})^2 + (\tilde{\sigma}^V_{\max})^2}} \right\} = \operatorname*{argmin}_V \left\{ \frac{\tilde{\sigma}^V_{\max}}{\tilde{\lambda}^V_{\min}} \right\}.$$

While practically we usually take the positive definite matrix between W and T to substitute the matrix V. If both W and T are positive definite, then the better matrix V can be chosen such that

$$\frac{\tilde{\sigma}^V_{\max}}{\tilde{\lambda}^V_{\min}} = \min\left\{ \frac{\tilde{\sigma}^W_{\max}}{\tilde{\lambda}^W_{\min}}, \frac{\tilde{\sigma}^T_{\max}}{\tilde{\lambda}^T_{\min}} \right\} = \min\left\{ \tilde{\sigma}^W_{\max}, \frac{1}{\tilde{\lambda}^T_{\min}} \right\}.$$

We mention that the SPHSS iteration method has also been studied in [43] for solving a general class of non-Hermitian positive definite linear systems, and similar results about the convergence conditions and the quasi-optimal parameters are given there. However, we provide new analysis of the new SPHSS iteration method in Remarks 1 and 2.

4. Convergence analysis about the PSPHSS iteration method

In this section, we consider the convergence properties of the PSPHSS iteration method and search for the optimal parameters α^V_* and ω^V_*. First, we rewrite the PSPHSS iteration scheme (2.5) as

$$x^{(k+1)} = M^V(\alpha, \omega)x^{(k)} + N^V(\alpha, \omega)(\omega - i)b, \quad k = 0, 1, 2, \ldots,$$

where

$$\begin{aligned} M^V(\alpha, \omega) &= (\alpha V + \omega W + T)^{-1}[\alpha V - i(\omega T - W)] \\ &= [\alpha I + V^{-1}(\omega W + T)]^{-1}[\alpha I - iV^{-1}(\omega T - W)] \end{aligned}$$

and

$$N^V(\alpha, \omega) = (\alpha V + \omega W + T)^{-1},$$

with α, ω being given positive constants.

As before, if we define

$$F^V(\alpha, \omega) = \alpha V + \omega W + T, \quad \text{and} \quad G^V(\alpha, \omega) = \alpha V - i(\omega T - W),$$

then it holds that

$$(\omega - i)A = F^V(\alpha, \omega) - G^V(\alpha, \omega), \quad \text{and} \quad M^V(\alpha, \omega) = F^V(\alpha, \omega)^{-1} G^V(\alpha, \omega).$$

Thus, the splitting matrix $F^V(\alpha, \omega)$ can be taken as a preconditioner for the complex symmetric matrix $(\omega - i)A \in \mathbb{C}^{n\times n}$, which is referred as the PSPHSS preconditioner.

In particular, if we take $V = W$, then the SPHSS iteration scheme is now induced by the matrix splitting

$$(\omega - i)A = F^W(\alpha, \omega) - G^W(\alpha, \omega)$$

with

$$F^W(\alpha, \omega) = (\alpha + \omega)W + T, \quad \text{and} \quad G^W(\alpha, \omega) = \alpha W - i(\omega T - W).$$

For convenience, we define

$$\hat{\lambda}^V_{\min}(\omega) = \min_{\hat{\lambda}_j \in sp(V^{-\frac{1}{2}}(\omega W + T)V^{-\frac{1}{2}})} \{\hat{\lambda}_j\} \quad \text{and} \quad \hat{\sigma}^V_{\max}(\omega) = \max_{\hat{\sigma}_j \in sp(V^{-\frac{1}{2}}(\omega T - W)V^{-\frac{1}{2}})} \{|\hat{\sigma}_j|\}. \tag{4.1}$$

Since $\tilde{W} = \omega W + T$ is also a symmetric positive definite matrix, then according to the convergence analysis about the SPHSS iteration method given in Section 3, we immediately obtain the following results.

Theorem 2. *If $W, T \in \mathbb{R}^{n\times n}$ are symmetric positive semi-definite matrices with at least one of them being positive definite, then the spectral radius $\rho(M^V(\alpha, \omega))$ is bounded by*

$$\delta^V(\alpha, \omega) = \frac{\sqrt{\alpha^2 + (\hat{\sigma}^V_{\max}(\omega))^2}}{\alpha + \hat{\lambda}^V_{\min}(\omega)}. \tag{4.2}$$

Furthermore, it is easy to see that

(i) *if $\hat{\lambda}^V_{\min}(\omega) \geq \hat{\sigma}^V_{\max}(\omega)$, then $\delta^V(\alpha, \omega) < 1$ i.e. the iteration converges for any $\alpha > 0$;*

(ii) *if $\hat{\lambda}^V_{\min}(\omega) < \hat{\sigma}^V_{\max}(\omega)$, then $\delta^V(\alpha, \omega) < 1$ if and only if*

$$\alpha > \frac{(\hat{\sigma}^V_{\max}(\omega))^2 - (\hat{\lambda}^V_{\min}(\omega))^2}{2\hat{\lambda}^V_{\min}(\omega)}. \tag{4.3}$$

Corollary 2. *Under the conditions of Theorem 2, the optimal parameter $\alpha^V_*(\omega)$ minimizing the upper bound $\delta^V(\alpha, \omega)$ of the spectral radius $\rho(M^V(\alpha, \omega))$ is*

$$\alpha^V_*(\omega) = \operatorname*{argmin}_{\alpha} \left\{ \frac{\sqrt{\alpha^2 + (\hat{\sigma}^V_{\max}(\omega))^2}}{\alpha + \hat{\lambda}^V_{\min}(\omega)} \right\} = \frac{(\hat{\sigma}^V_{\max}(\omega))^2}{\hat{\lambda}^V_{\min}(\omega)}, \tag{4.4}$$

and

$$\delta^V(\alpha^V_*(\omega), \omega) = \frac{\hat{\sigma}^V_{\max}(\omega)}{\sqrt{(\hat{\lambda}^V_{\min}(\omega))^2 + (\hat{\sigma}^V_{\max}(\omega))^2}}. \tag{4.5}$$

According to Theorem 2, the PSPHSS iteration method converges, for any $\omega > 0$ by choosing a proper parameter α satisfying (4.3), or for any $\alpha > 0$ if the parameter ω satisfies $\hat{\lambda}^V_{\min}(\omega) \geq \hat{\sigma}^V_{\max}(\omega)$ which leads to $\delta^V(\alpha^V_*(\omega), \omega) \leq \frac{1}{\sqrt{2}}$. In addition, we have the following results by similar analysis as in [25].

Theorem 3. *Let $\hat{\lambda}^V_{\min}(\omega)$, $\hat{\sigma}^V_{\max}(\omega)$ be defined as in (4.1), $\tilde{\lambda}^V_{\min}$, $\tilde{\sigma}^V_{\max}$, $\tilde{\lambda}^V_{\max}$ and $\tilde{\sigma}^V_{\min}$ be defined as in (3.2). Denote*

$$\bar{\omega}^V = \frac{\tilde{\lambda}^V_{\min} + \tilde{\lambda}^V_{\max}}{\tilde{\sigma}^V_{\min} + \tilde{\sigma}^V_{\max}}, \qquad \omega^V_1 = \frac{\tilde{\lambda}^V_{\min} + \tilde{\sigma}^V_{\min}}{\tilde{\sigma}^V_{\max} - \tilde{\lambda}^V_{\min}}, \qquad \omega^V_2 = \frac{\tilde{\lambda}^V_{\max} - \tilde{\sigma}^V_{\min}}{\tilde{\sigma}^V_{\min} + \tilde{\lambda}^V_{\min}}.$$

Then $\hat{\lambda}^V_{\min}(\omega) \geq \hat{\sigma}^V_{\max}(\omega)$, i.e. the PSPHSS iteration method converges for any $\alpha > 0$ if any of the following conditions hold:

(1) *$\tilde{\lambda}^V_{\min} \geq \tilde{\sigma}^V_{\max}$ and $\omega \geq \bar{\omega}^V$;*

(2) *$\tilde{\lambda}^V_{\min} < \tilde{\sigma}^V_{\max}$ and $\bar{\omega}^V \leq \omega \leq \omega^V_1$;*

(3) *$\omega^V_2 \leq \omega < \bar{\omega}^V$.*

Moreover, we can obtain the quasi-optimal parameter ω^V_* by similar analysis as in [25].

Theorem 4. *Let $\tilde{\lambda}^V_{\min}$, $\tilde{\sigma}^V_{\max}$, $\tilde{\lambda}^V_{\max}$ and $\tilde{\sigma}^V_{\min}$ be defined as in (3.2). Then the quasi-optimal parameter ω^V_* for the PSPHSS iteration method is given by*

$$\omega^V_* = \bar{\omega}^V = \frac{\tilde{\lambda}^V_{\min} + \tilde{\lambda}^V_{\max}}{\tilde{\sigma}^V_{\min} + \tilde{\sigma}^V_{\max}}.$$

Remark 3. When $V = I$, the PSPHSS iteration is reduced to the PSHSS iteration. According to Corollary 2, by similar analysis as in Remark 1, for comparing the PSPHSS iteration using $V = \tilde{W} = \omega W + T$ and the PSHSS iteration, we have

$$\delta^{\tilde{W}}(\alpha_*^{\tilde{W}}(\omega), \omega) \leq \delta^{I}(\alpha_*^{I}(\omega), \omega).$$

Therefore, the PSPHSS iteration using $V = \omega W + T$ is more efficient than the PSHSS iteration when the optimal parameters are used.

However, for simplicity, we consider the special case of $V = W$, and the PSPHSS iteration scheme (2.5) becomes

$$[(\alpha + \omega)W + T]x^{(k+1)} = [\alpha W - i(\omega T - W)]x^k + (\omega - i)b,$$

which can be rewritten as

$$x^{(k+1)} = M^W(\alpha, \omega)x^k + N^W(\alpha, \omega)(\omega - i)b,$$

where

$$\begin{aligned} M^W(\alpha, \omega) &= [(\alpha + \omega)W + T]^{-1}[\alpha W - i(\omega T - W)] \\ &= [(\alpha + \omega)I + W^{-1}T]^{-1}[\alpha I - i(\omega W^{-1}T - I)], \\ N^W(\alpha, \omega) &= [(\alpha + \omega)W + T]^{-1}. \end{aligned} \tag{4.6}$$

In this case, we have the following results.

Theorem 5. *If we take $V = W$ for the PSPHSS iteration method, then the spectral radius $\rho(M^W(\alpha, \omega))$ is bounded by*

$$\delta^W(\alpha, \omega) = \frac{\sqrt{\alpha^2 + \omega^2(\tilde{\sigma}_{\max}^W)^2 + 1}}{\alpha + \omega + \tilde{\sigma}_{\min}^W}. \tag{4.7}$$

Moreover, the PSPHSS iteration using $V = W$ converges if any of the following conditions hold:

(1) *$\omega = 1$ for any $\alpha > 0$;*

(2) *$\omega \neq 1$ and*

$$\alpha > \frac{\omega^2(\tilde{\sigma}_{\max}^W)^2 + 1 - (\omega + \tilde{\sigma}_{\min}^W)^2}{2(\omega + \tilde{\sigma}_{\min}^W)}, \tag{4.8}$$

in particular, $\omega \neq 1$ and $\alpha > \frac{\omega^2(\tilde{\sigma}_{\max}^W)^2 + 1 - \omega^2}{2\omega}$.

Proof. According to (4.6), it is easy to see that

$$\rho(M^W(\alpha, \omega)) \leq \max_{\tilde{\sigma}_j \in sp(W^{-\frac{1}{2}}TW^{-\frac{1}{2}})} \frac{\sqrt{\alpha^2 + (\omega\tilde{\sigma}_j - 1)^2}}{\alpha + \omega + \tilde{\sigma}_j} \leq \max_{\tilde{\sigma}_j \in sp(W^{-\frac{1}{2}}TW^{-\frac{1}{2}})} \frac{\sqrt{\alpha^2 + \omega^2\tilde{\sigma}_j^2 + 1}}{\alpha + \omega + \tilde{\sigma}_j} \leq \delta^W(\alpha, \omega). \tag{4.9}$$

Furthermore, when $\omega = 1$ for any $\alpha > 0$, we have

$$\frac{\sqrt{\alpha^2 + \omega^2\tilde{\sigma}_j^2 + 1}}{\alpha + \omega + \tilde{\sigma}_j} < 1$$

holds for any $\tilde{\sigma}_j \in sp(W^{-1}T)$, which leads to $\rho(M^W(\alpha, \omega)) < 1$. Meanwhile, when $\omega \neq 1$, the inequality (4.8) is equivalent to

$$\alpha^2 + \omega^2(\tilde{\sigma}_{\max}^W)^2 + 1 < (\alpha + \omega + \tilde{\sigma}_{\min}^W)^2,$$

which means that $\delta^W(\alpha, \omega) < 1$. □

Similar as Corollary 1, we have the following results.

Corollary 3. *Under the conditions of Theorem 5, the quasi-optimal parameter $\alpha_*^W(\omega)$ minimizing the upper bound $\delta^W(\alpha, \omega)$ of the spectral radius $\rho(M^W(\alpha, \omega))$ is*

$$\alpha_*^W(\omega) = \operatorname*{argmin}_{\alpha} \left\{ \frac{\sqrt{\alpha^2 + \omega^2(\tilde{\sigma}_{\max}^W)^2 + 1}}{\alpha + \omega + \tilde{\sigma}_{\min}^W} \right\} = \frac{\omega^2(\tilde{\sigma}_{\max}^W)^2 + 1}{\omega + \tilde{\sigma}_{\min}^W}, \tag{4.10}$$

and

$$\delta^W(\alpha_*^W(\omega), \omega) = \frac{\sqrt{\omega^2(\tilde{\sigma}_{\max}^W)^2 + 1}}{\sqrt{\omega^2(\tilde{\sigma}_{\max}^W)^2 + 1 + (\omega + \tilde{\sigma}_{\min}^W)^2}}. \tag{4.11}$$

Note that if the parameter ω is chosen such that $(\omega + \tilde{\sigma}^W_{\min})^2 \geq \omega^2(\tilde{\sigma}^W_{\max})^2 + 1$, then we obtain $\delta^W(\alpha^W_*(\omega), \omega) \leq \frac{1}{\sqrt{2}}$.
On the other hand, according to the definitions in (3.2) and (4.1), we have

$$\tilde{\lambda}^W_{\min} = \tilde{\lambda}^W_{\max} = 1, \qquad \hat{\lambda}^W_{\min}(\omega) = \omega + \tilde{\sigma}^W_{\min},$$

and

$$\hat{\sigma}^W_{\max}(\omega) = \max_{\tilde{\sigma}_j \in sp(W^{-\frac{1}{2}} T W^{-\frac{1}{2}})} \{|\omega\tilde{\sigma}_j - 1|\} \leq \sqrt{\omega^2(\tilde{\sigma}^W_{\max})^2 + 1}.$$

Thus, the results of Theorems 3–4 can be simplified.

5. Numerical results

In this section, we use some numerical examples to test the computational efficiencies of the SPHSS and PSPHSS iteration methods for solving the complex symmetric linear system (1.1). Numerical comparisons with the PMHSS [14], the SHSS [20], the PSHSS [25], and the PGSOR iteration methods are given to show the advantage of the SPHSS and PSPHSS iteration methods. We show the results for these methods by computations with the preconditioned GMRES methods.

We compare these methods mainly from the point of view of the number of iterations (denoted as IT) and the CPU times (denoted as CPU). Our experiments are carried out in MATLAB R2013a on Intel(R) Core(TM) CPU 1.8 × 2 Ghz and 4.00 GB of RAM. The CPU time is recorded by the command "tic-toc".

In our computations, the initial approximation $x^{(0)} = u^{(0)} + iv^{(0)}$ is always chosen to be zero vector and the stopping criteria for all the methods are

$$\frac{\|b - Ax^{(k)}\|_2}{\|b\|_2} \leq 10^{-6},$$

where $x^{(k)} = u^{(k)} + iv^{(k)}$ is the current approximation.

Example 1. (See [2,9]) Consider a system of linear equations of the form $(W + iT)x = b$. The matrix $T \in \mathbb{R}^{n\times n}$ possesses the tensor-product form

$$T = I \otimes B_m + B_m \otimes I \quad \text{with} \quad B_m = tridiag(-1, 2, -1) \in \mathbb{R}^{m\times m},$$

so T is an $n \times n$ block-tridiagonal matrix with $n = m^2$. The matrix $W \in \mathbb{R}^{n\times n}$ is defined by

$$W = 10(I \otimes B_c + B_c \otimes I) + 9(e_1 e_m^T + e_m e_1^T) \otimes I \quad \text{with} \quad B_c = B_m - e_1 e_m^T - e_m e_1^T \in \mathbb{R}^{m\times m},$$

where e_1 and e_m are the first and mth unit basis vectors in $\mathbb{R}^m$, respectively. Here W is the five-point centered difference matrix which approximates the negative Laplacian operator with periodic boundary conditions, on a uniform mesh in the square $[0, 1] \times [0, 1]$ with the mesh-size $h = 1/(m + 1)$. We choose the vector $b = (1 + i)A\mathbf{1}$, where $\mathbf{1}$ is a special vector of all entries equal to 1.

Example 2. In direct frequency domain analysis of an n-degree-of-freedom linear system, researchers often need to solve the following system of linear equations

$$[(K - \varpi^2 M) + i(C_H + \varpi C_V)]x = b,$$

where K is the (real symmetric) stiffness matrix, M is the inertia (mass) matrix, which is typically real symmetric, C_V is the (real diagonal) viscous damping matrix and C_H is the (real symmetric) hysteretic damping matrix. If $C_H = \beta K$, then β is a damping coefficient, and ϖ is the driving circular frequency. For more details, we refer to [4,44].

Similar as the numerical examples given in [2,41,42], we consider to solve the following linear system as well

$$\left[\left(K - \varpi^2 I\right) + i\left(\beta K + 10\varpi I\right)\right]x = b,$$

where ϖ, β are given real constants. The matrix $K \in \mathbb{R}^{n\times n}$ possesses the tensor-product form

$$K = I \otimes B_m + B_m \otimes I \quad \text{with} \quad B_m = \frac{1}{h^2} tridiag(-1, 2, -1) \in \mathbb{R}^{m\times m}.$$

In our tests, the right-hand side vector b with its jth entry b_j is given by

$$b_j = \frac{(1 + i)j}{h^2(j + 1)^2}, \quad j = 1, 2, \ldots, n.$$

In addition, we normalize the system by multiplying both sides by h^2.

Table 1
The numerical results of Example 1 for different iteration methods.

Method		Grid			
		16 × 16	32 × 32	48 × 48	64 × 64
PMHSS	α_*	0.7	0.7	0.7	0.7
($V = W$)	IT	3	3	3	3
	CPU(s)	0.0708	0.1157	0.2439	0.4796
SHSS	α_*	0.01	0.01	0.01	0.005
	IT	3	4	4	4
	CPU(s)	0.0691	0.1116	0.2391	0.4776
PSHSS	α_*	0.01	0.01	0.01	0.01
	ω_*	10	10	10	10
	IT	3	3	3	3
	CPU(s)	0.0699	0.1027	0.2169	0.4462
SPHSS	α_*	0.01	0.01	0.01	0.01
($V = W$)	IT	3	3	3	3
	CPU(s)	0.0665	0.0973	0.1724	0.3151
PSPHSS	α_*	0.01	0.01	0.01	0.01
($V = W$)	ω_*	10	10	10	10
	IT	3	3	3	3
	CPU(s)	0.0671	0.0979	0.1738	0.3178
PGSOR	α_*	0.99	0.99	0.99	0.99
	ω_*	10	10	10	10
	IT	2	2	2	2
	CPU(s)	0.0704	0.1541	0.3605	0.8958

Table 2
Numerical results of Example 2 for different iteration methods with $(\varpi, \beta) = (\pi/4, 0.02)$.

Method		Grid			
		16 × 16	32 × 32	48 × 48	64 × 64
PMHSS	α_*	0.9	0.9	0.9	0.9
($V = W$)	IT	5	5	5	5
	CPU(s)	0.0827	0.1417	0.2814	0.5141
SHSS	α_*	0.0005	0.0005	0.0005	0.0005
	IT	5	5	5	5
	CPU(s)	0.0621	0.0865	0.1705	0.3554
PSHSS	α_*	0.005	0.005	0.005	0.005
	ω_*	5	5	5	5
	IT	5	5	5	5
	CPU(s)	0.0624	0.0863	0.1701	0.3555
SPHSS	α_*	0.01	0.01	0.01	0.01
($V = W$)	IT	5	5	5	5
	CPU(s)	0.0613	0.0809	0.1328	0.2597
PSPHSS	α_*	0.01	0.01	0.01	0.01
($V = W$)	ω_*	5	5	5	5
	IT	5	5	5	5
	CPU(s)	0.0615	0.0816	0.1346	0.2647
PGSOR	α_*	0.99	0.99	0.99	0.99
	ω_*	5	5	5	5
	IT	4	4	4	4
	CPU(s)	0.0801	0.2103	0.5607	1.3715

Tables 1–2 show the numerical results of various iteration methods with respect to different problem sizes for Examples 1–2 by using the experimental optimal parameters α_* and ω_*, respectively. The parameters $\varpi = \pi/4$ and $\beta = 0.02$ are considered in Example 2. The results about iterations and CPU times are listed in the tables. It can be seen that the experimental optimal parameters of the considered methods are very stable, which brings convenience for us to find the optimal parameters for solving large complex symmetric linear systems. We can see that the SPHSS method has a little advantage over the SHSS method from the point of view of iterations and CPU times by using the experimental optimal parameters. Similarly, the PSPHSS iteration method is a little more efficient than the PSHSS iteration method by using the experimental optimal parameters from the point of view of CPU times, because it is a combination of the SPHSS and PSHSS methods, and it inherits the merits of both.

Note that if the computations are carried out without using the preconditioned GMRES methods, the SHSS method is much less efficient than the PSHSS iteration method, see [25]. Similarly, the SPHSS method is less efficient than the PSPHSS iteration method. Therefore, according to the results of Tables 1–2, we can conclude that the PSPHSS and the PSHSS iteration methods outperform the other methods when the experimental optimal parameters are used.

In order to make further comparisons among the PSPHSS, the PSHSS, and the PGSOR methods, Tables 3–4 show the numerical results of these three methods with respect to different problem sizes for Example 2 with and without using the

Table 3
Numerical results of Example 2 for the PSPHSS ($V = W$), the PSHSS and the PGSOR iteration method with $(\varpi, \beta) = (0.2, 0.5)$.

Method		Grid			
		16×16	32×32	48×48	64×64
PSPHSS	IT	4	4	4	4
$(\alpha_*, \omega_*) = (0.001, 2)$	CPU(s)	0.0595	0.0742	0.1273	0.2297
PSHSS	IT	4	4	4	4
$(\alpha_*, \omega_*) = (0.001, 2)$	CPU(s)	0.0609	0.0811	0.1566	0.3334
PGSOR	IT	3	3	3	3
$(\alpha_*, \omega_*) = (0.99, 2)$	CPU(s)	0.0707	0.1562	0.3856	0.9980
PSPHSS	IT	4	4	4	4
$(\alpha, \omega) = (0.5, 10)$	CPU(s)	0.0601	0.0745	0.1282	0.2318
PSHSS	IT	6	9	12	15
$(\alpha, \omega) = (0.5, 10)$	CPU(s)	0.0630	0.0947	0.2094	0.4626
PGSOR	IT	6	6	6	6
$(\alpha, \omega) = (0.5, 10)$	CPU(s)	0.0886	0.2724	0.7187	1.7283
PSPHSS	IT	4	4	4	4
$(\alpha, \omega) = (0.5, 0.1)$	CPU(s)	0.0612	0.0750	0.1286	0.2342
PSHSS	IT	15	28	40	52
$(\alpha, \omega) = (0.5, 0.1)$	CPU(s)	0.0711	0.1364	0.3605	1.0103
PGSOR	IT	8	8	8	8
$(\alpha, \omega) = (0.5, 0.1)$	CPU(s)	0.1024	0.2934	0.7756	1.8693

Table 4
Numerical results of Example 2 for the PSPHSS ($V = W$), the PSHSS and the PGSOR iteration method with $(\varpi, \beta) = (-1, 1)$.

Method		Grid			
		16×16	32×32	48×48	64×64
PSPHSS	IT	4	4	4	4
$(\alpha_*, \omega_*) = (0.01, 1.4)$	CPU(s)	0.0687	0.0829	0.1356	0.2674
PSHSS	IT	5	5	5	5
$(\alpha_*, \omega_*) = (0.01, 5.5)$	CPU(s)	0.0842	0.0848	0.1689	0.3518
PGSOR	IT	3	3	3	3
$(\alpha_*, \omega_*) = (0.99, 1.4)$	CPU(s)	0.0985	0.2423	0.6424	1.5156
PSPHSS	IT	5	5	5	5
$(\alpha, \omega) = (0.5, 9.5)$	CPU(s)	0.0695	0.0844	0.1413	0.2763
PSHSS	IT	6	9	13	15
$(\alpha, \omega) = (0.5, 9.5)$	CPU(s)	0.0689	0.0958	0.1974	0.4439
PGSOR	IT	8	8	8	8
$(\alpha, \omega) = (0.5, 9.5)$	CPU(s)	0.1243	0.3743	0.9804	2.2316
PSPHSS	IT	4	4	4	4
$(\alpha, \omega) = (0.5, 0.05)$	CPU(s)	0.0688	0.0848	0.1405	0.2731
PSHSS	IT	13	23	32	42
$(\alpha, \omega) = (0.5, 0.05)$	CPU(s)	0.0698	0.1242	0.2938	0.8362
PGSOR	IT	8	8	8	8
$(\alpha, \omega) = (0.5, 0.05)$	CPU(s)	0.1255	0.3781	0.9769	2.2287

experimental optimal parameters, when the parameters $(\varpi, \beta) = (0.2, 0.5)$ and $(\varpi, \beta) = (-1, 1)$, respectively, where the non-optimal parameter α is always fixed, because the distances between 0.5 and the experimental optimal parameters α_* are almost the same for all these three iteration methods. We can see that the speed of convergence for the PSPHSS iteration method is affected slightly with changes in the parameters, and the PSPHSS method is the most efficient from the point of view of iterations and CPU times when the experimental optimal parameters are not used. In a word, the numerical results confirm that the PSPHSS iteration method is very powerful whether the experimental optimal parameters are used or not.

6. Conclusions

In this work, we introduce a single-step preconditioned HSS (SPHSS) and an efficient parameterized SPHSS (PSPHSS) iteration method for solving a class of complex symmetric linear systems. Theoretical analysis shows that, under a loose restriction on the parameters α and ω, the iterative sequences are convergent to the unique solution of the linear system for any initial guess. Furthermore, the quasi-optimal values of the iteration parameters for the SPHSS and PSPHSS iteration methods are also discussed. Both theoretical and numerical results show that the SPHSS and PSPHSS iteration methods have a little advantage over the SHSS and PSHSS iteration methods, respectively. Moreover, the PSPHSS iteration method outperforms several iteration methods from the point of view of the number of iterations and the CPU times when the experimental optimal parameters are used. In particular, the PSPHSS iteration method is the most efficient by comparing with the PSHSS and PGSOR methods if the experimental optimal parameters are not used.

Acknowledgments

This work is supported by NNSF with Nos. 11461046, 61563033 and 11401293, NSF of Jiangxi Province with Nos. 20161ACB21005 and 20151BAB201009, and the Scientific Research Foundation of Graduate School of Nanchang University with Nos. YC2015-S018 and CX2016143.

References

[1] O. Axelsson, A. Kucherov, Real valued iterative methods for solving complex symmetric linear systems, Numer. Linear Algebra Appl. 7 (2000) 197–218.
[2] Z.-Z. Bai, M. Benzi, F. Chen, Modified HSS iteration methods for a class of complex symmetric linear systems, Computing 87 (2010) 93–111.
[3] Z.-Z. Bai, Rotated block triangular preconditioning based on PMHSS, Sci. China Math. 56 (2013) 2523–2538.
[4] M. Benzi, D. Bertaccini, Block preconditioning of real-valued iterative algorithms for complex linear systems, IMA J. Numer. Anal. 28 (2008) 598–618.
[5] D. Bertaccini, Efficient solvers for sequences of complex symmetric linear systems, Electron. Trans. Numer. Anal. 18 (2004) 49–64.
[6] Z.-Z. Bai, G.H. Golub, M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl. 24 (2003) 603–626.
[7] Z.-Z. Bai, Optimal parameters in the HSS-like methods for saddle-point problems, Numer. Linear Algebra Appl. 16 (2009) 447–479.
[8] Z.-Z. Bai, On semi-convergence of Hermitian and skew-Hermitian splitting methods for singular linear systems, Computing 89 (2010) 171–197.
[9] Z.-Z. Bai, G.H. Golub, L.-Z. Lu, J.-F. Yin, Block triangular and skew-Hermitian splitting methods for positive-definite linear systems, SIAM J. Sci. Comput. 26 (2005) 844–863.
[10] Z.-Z. Bai, G.H. Golub, M.K. Ng, On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations, Numer. Linear Algebra Appl. 14 (2007) 319–335.
[11] Z.-Z. Bai, G.H. Golub, M.K. Ng, On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, Linear Algebra Appl. 428 (2008) 413–440.
[12] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, Numer. Math. 98 (2004) 1–32.
[13] M. Benzi, A generalization of the Hermitian and skew-Hermitian splitting iteration, SIAM J. Matrix Anal. Appl. 31 (2009) 360–374.
[14] Z.-Z. Bai, M. Benzi, F. Chen, On preconditioned MHSS iteration methods for complex symmetric linear systems, Numer. Algorithms 56 (2011) 297–317.
[15] Z.-Z. Bai, M. Benzi, F. Chen, Z.-Q. Wang, Preconditioned MHSS iteration methods for a class of block two-by-two linear systems with applications to distributed control problems, IMA J. Numer. Anal. 33 (2013) 343–369.
[16] Q.-Q. Zheng, C.-F. Ma, Accelerated PMHSS iteration methods for complex symmetric linear systems, Numer. Algorithms 73 (2016) 501–516.
[17] X. Li, A.-L. Yang, Y.-J. Wu, Lopsided PMHSS iteration method for a class of complex symmetric linear systems, Numer. Algorithms 66 (2014) 555–568.
[18] H.N. Pour, H.S. Goughery, New Hermitian and skew-Hermitian splitting methods for non-Hermitian positive-definite linear systems, Numer. Algorithms 69 (2015) 207–225.
[19] X.-Y. Xiao, H.-W. Yin, Efficient parameterized HSS iteration methods for complex symmetric linear systems, Comput. Math. Appl. 73 (2017) 87–95.
[20] C.-X. Li, S.-L. Wu, A single-step HSS method for non-Hermitian positive definite linear systems, Appl. Math. Lett. 44 (2015) 26–29.
[21] Z.-Z. Bai, J.-F. Yin, Y.-F. Su, A shift-splitting preconditioner for non-Hermitian positive definite matrices, J. Comput. Math. 24 (2006) 539–552.
[22] Y. Cao, J. Du, Q. Niu, Shift-splitting preconditioners for saddle point problems, J. Comput. Appl. Math. 272 (2014) 239–250.
[23] Q.-Q. Shen, Q. Shi, Generalized shift-splitting preconditioners for nonsingular and singular generalized saddle point problems, Comput. Math. Appl. 72 (2016) 632–641.
[24] Q.-Q. Zheng, L.-Z. Lu, A shift-splitting preconditioner for a class of block two-by-two linear systems, Appl. Math. Lett. 66 (2017) 54–60.
[25] M.-L. Zeng, C.-F. Ma, A parameterized SHSS iteration method for a class of complex symmetric system of linear equations, Comput. Math. Appl. 71 (2016) 2124–2131.
[26] Z.-Z. Bai, Y.-B. Deng, Y.-H. Gao, Iterative orthogonal direction methods for Hermitian minimum norm solutions of two consistent matrix equations, Numer. Linear Algebra Appl. 13 (2006) 801–823.
[27] Z.-Z. Bai, A class of two-stage iterative methods for systmes of weakly nonlinear equations, Numer. Algorithms 14 (1997) 295–319.
[28] Z.-Z. Bai, On Hermitian and skew-Hermitian spliting iteration methods for continuous Sylvester equations, J. Comput. Math. 29 (2011) 185–198.
[29] Z.-Z. Bai, X.-P. Guo, On Newton-HSS methods for systems of nonlinear equations with positive definite Jacobian matrices, J. Comput. Math. 28 (2010) 235–260.
[30] Z.-Z. Bai, Several splittings for non-Hermitian linear systems, Sci. China (Ser. A: Math.) 51 (2008) 1339–1348.
[31] Z.-Z. Bai, G.H. Golub, C.-K. Li, Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices, Math. Comp. 76 (2007) 287–298.
[32] Z.-Z. Bai, Y.-M. Huang, M.K. Ng, On preconditioned iterative methods for Burgers equations, SIAM J. Sci. Comput. 29 (2007) 415–439.
[33] Z.-Z. Bai, M.K. Ng, Preconditioners for nonsymmetric block Toeplitz-like-plus-diagonal linear systems, Numer. Math. 96 (2003) 197–220.
[34] X. Wang, W.-W. Li, L.-Z. Mao, On positive-definite and skew-Hermitian splitting iteration methods for continuous Sylvester equation $AX + XB = C$, Comput. Math. Appl. 66 (2013) 2352–2361.
[35] R. Zhou, X. Wang, X.-B. Tang, A generalization of the Hermitian and skew-Hermitian splitting iteration method for solving Sylvester equations, Appl. Math. Comput. 271 (2015) 609–617.
[36] R. Zhou, X. Wang, P. Zhou, A modified HSS iteration method for solving the complex linear matrix equation $AXB = C$, J. Comput. Math. 34 (2016) 437–450.
[37] R. Zhou, X. Wang, X.-B. Tang, Preconditioned positive-definite and skew-Hermitian splitting iteration methods for continuous Sylvester equations $AX + XB = C$, East Asian J. Appl. Math. 7 (2017) 55–69.
[38] X. Wang, W.-W. Li, A modified GPSS method for non-Hermitian positive definite linear systems, Appl. Math. Comput. 234 (2014) 253–259.
[39] X. Wang, Y. Li, L. Dai, On Hermitian and skew-Hermitian splitting iteration methods for the linear matrix equation $AXB = C$, Comput. Math. Appl. 65 (2013) 657–664.
[40] C.-R. Chen, C.-F. Ma, AOR-Uzawa iterative method for a class of complex symmetric linear system of equations, Comput. Math. Appl. 72 (2016) 2462–2472.
[41] D.K. Salkuyeh, D. Hezari, V. Edalatpour, Generalized SOR iterative method for a class of complex symmetric linear system of equations, Int. J. Comput. Math. 92 (2015) 802–815.

[42] D. Hezari, V. Edalatpour, D.K. Salkuyeh, Preconditioned GSOR iterative method for a class of complex symmetric system of linear equations, Numer. Linear Algebra Appl. 22 (2015) 761–776.
[43] Y.-J. Wu, X. Li, Y.-Y. Yuan, A non-alternating preconditioned HSS iteration method for non-Hermitian positive definite linear systems, J. Comput. Appl. Math. 36 (2017) 367–381.
[44] A. Feriani, F. Perotti, V. Simoncini, Iterative system solvers for the frequency analysis of linear mechanical systems, Comput. Methods Appl. Mech. Engrg. 190 (2000) 1719–1739.