GLASNIK MATEMATIČKI
Vol. 57(77)(2022), 149 – 159

A GENERALIZED TWO-SWEEP SHIFT SPLITTING METHOD FOR NON-HERMITIAN POSITIVE DEFINITE LINEAR SYSTEMS

Shi-Liang Wu and Cui-Xia Li
Yunnan Normal University, P.R. China

Abstract. In this paper, based on the shift splitting of the coefficient matrix, a generalized two-sweep shift splitting (GTSS) method is introduced to solve the non-Hermitian positive definite linear systems. Theoretical analysis shows that the GTSS method is convergent to the unique solution of the linear systems under a loose restriction on the iteration parameter. Numerical experiments are reported to the efficiency of the GTSS method.

1. Introduction

Consider the numerical solution of the following linear system

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

with A being non-Hermitian positive definite (that is, its Hermitian part $H = \frac{1}{2}(A + A^*)$ is positive definite, where A^* denotes the conjugate transpose of the matrix A), x and b be an unknown vector and a given vector, respectively. The non-Hermitian positive definite linear systems (1.1) is a class of important equations and often arises in many problems in scientific and engineering computing, including quantum chemistry, electrical engineering, Helmholtz equation, and so on. One can see [10, 7, 11, 9] for more details.

In recent years, to efficiently solve the non-Hermitian positive definite linear systems (1.1), a large amount of iteration methods have been proposed in the literature, such as the Hermitian and skew-Hermitian splitting (HSS) method in [3], the preconditioned HSS (PHSS) method in [4], the accelerated

2020 *Mathematics Subject Classification.* 65F10, 65F15.
Key words and phrases. Non-Hermitian matrix, shift splitting, GTSS method, convergence.

HSS (AHSS) method in [2], the modified HSS (MHSS) method in [1] and so on.

In [5], Bai *et al.* designed the following shift splitting of the matrix A

$$A = \frac{1}{2}(\beta I + A) - \frac{1}{2}(\beta I - A),\ \beta > 0.$$

This splitting naturally leads to the shift splitting (SS) iteration scheme for solving the large sparse non-Hermitian positive definite linear systems (1.1) and is described below.

The SS method: Given an initial guess $x^{(0)}$, for $k = 0, 1, 2, \dots$ until $\{x^{(k)}\}$ converges, compute

$$(\beta I + A)x^{(k+1)} = (\beta I - A)x^{(k)} + 2b, \tag{1.2}$$

where β is a given positive constant.

Bai *et al.* proved in [5] that the SS method (1.2) converges unconditionally to the unique solution of the linear systems when A is non-Hermitian positive definite. Meanwhile, in theory, the optimal iteration parameter $\beta = \|A\|_2$, where $\|A\|_2$ denotes the 2-norm of the matrix A, is obtained to minimize an upper bound of the spectral radius of the iteration matrix of the SS method (1.2).

In this paper, the shift splitting technique is generalized and a generalized two-sweep shift splitting (GTSS) iteration method for the non-Hermitian positive definite linear systems (1.1) is proposed. Theoretical analysis shows that the GTSS method is convergent to the unique solution of the linear system under a loose restriction on the iteration parameter. Numerical experiments are reported to the efficiency of the GTSS method.

The remainder of this paper is organized as follows: in Section 2, the generalized two-sweep shift splitting (GTSS) iteration method is described and the convergence properties of the GTSS method are studied. In Section 3, numerical experiments are provided to show the effectiveness of the GTSS method. Finally, some concluding remarks are given in Section 4.

2. The GTSS method

In this section, to establish the GTSS method, we can express A as

$$A = \alpha I - (\alpha I - A) = (\beta I + A) - \beta I.$$

Based on this, the linear systems (1.1) has the following two equivalent forms,

$$\alpha x = (\alpha I - A)x + b, \tag{2.1}$$

$$(\beta I + A)x = \beta x + b. \tag{2.2}$$

Based on the fixed-point equations (2.1) and (2.2), the following generalized two-sweep shift splitting (GTSS) iteration method is established to solve the non-Hermitian positive definite linear systems (1.1).

The GTSS method. Let $x^{(0)} \in \mathbb{C}^n$ be an arbitrary initial guess. For $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ converges, compute the next iterate $x^{(k+1)}$ by the following procedure

$$\begin{cases} \alpha x^{(k+\frac{1}{2})} = (\alpha I - A)x^{(k)} + b, \\ (\beta I + A)x^{(k+1)} = \beta x^{(k+\frac{1}{2})} + b, \end{cases} \tag{2.3}$$

where α, β are given positive constants and I is the identity matrix.

Eliminating the intermediate vector $x^{(k+\frac{1}{2})}$ in (2.3) leads to the following iteration in fixed-point form

$$x^{(k+1)} = M_{\alpha,\beta}x^{(k)} + N_{\alpha,\beta}b, k = 0, 1, 2, \ldots \tag{2.4}$$

where

$$M_{\alpha,\beta} = \frac{\beta}{\alpha}(\beta I + A)^{-1}(\alpha I - A)$$

and

$$N_{\alpha,\beta} = \frac{\alpha + \beta}{\alpha}(\beta I + A)^{-1}.$$

In addition, let

$$M = (\alpha + \beta)^{-1}\alpha(\beta I + A) \text{ and } N = (\alpha + \beta)^{-1}\beta(\alpha I - A).$$

Then $A = M - N$ and $M_{\alpha,\beta} = M^{-1}N$. Therefore, the GTSS method is also obtained by the matrix splitting $A = M - N$. It follows that the matrix M can be used as a preconditioner for the non-Hermitian positive definite linear systems (1.1).

It is noted that when $\alpha = \beta$ in (2.3), the two-sweep shift splitting (TSS) iteration method is obtained and described below.

The TSS method. Let $x^{(0)} \in \mathbb{C}^n$ be an arbitrary initial guess. For $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x^{(k)}\}_{k=0}^{\infty}$ converges, compute the next iterate $x^{(k+1)}$ by the following procedure

$$\begin{cases} \alpha x^{(k+\frac{1}{2})} = (\alpha I - A)x^{(k)} + b, \\ (\alpha I + A)x^{(k+1)} = \alpha x^{(k+\frac{1}{2})} + b, \end{cases} \tag{2.5}$$

where α is a given positive constant and I is the identity matrix.

In fact, the TSS method (2.5) is equal to the SS method (1.2).

The GTSS method (2.3) is convergent if and only if the spectral radius $\rho(M_{\alpha,\beta})$ of the iteration matrix $M_{\alpha,\beta}$ is less than one, where $\rho(\cdot)$ denotes the spectral radius of the matrix.

For the convergence property of the GTSS method (2.3), we have the following theorem.

THEOREM 2.1. *Let $A \in \mathbb{C}^{n\times n}$ be non-Hermitian positive definite with $\alpha, \beta > 0$. Then the spectral radius $\rho(M_{\alpha,\beta})$ of the iteration matrix $M_{\alpha,\beta}$ of* *the GTSS method* (2.3) *is bounded by*

$$\delta_{\alpha,\beta} = \frac{\beta}{\alpha}\sqrt{\frac{\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2}{\beta^2 + 2\beta\eta_l(H) + \|A\|_2^2}}, \tag{2.6}$$

where $\eta_l(H)$ *is the smallest eigenvalue of the matrix* $H = \frac{1}{2}(A + A^*)$. *Moreover,*

(*i*) *if* $\alpha \geq \beta$, *then* $\delta_{\alpha,\beta} < 1$, *namely, the GTSS method* (2.3) *is convergent;*

(*ii*) *if* $\eta_l(H) + \frac{\|A\|_2^2}{\beta} > \frac{\|A\|_2^2}{\alpha}$, *then* $\delta_{\alpha,\beta} < 1$, *namely, the GTSS method* (2.3) *is convergent.*

Proof. Let $t = \frac{x^*Hx}{x^*x} > 0$. By the direct calculations, we have

$$\begin{aligned}
\rho(M_{\alpha,\beta})^2 &= \frac{\beta^2}{\alpha^2}\rho((\beta I + A)^{-1}(\alpha I - A)(\alpha I - A)^*(\beta I + A)^{-*}) \\
&= \frac{\beta^2}{\alpha^2}\max_{x\neq 0}\frac{x^*(\alpha I - A)(\alpha I - A)^*x}{x^*(\beta I + A)(\beta I + A)^*x} \\
&= \frac{\beta^2}{\alpha^2}\max_{x\neq 0}\frac{\alpha^2x^*x - \alpha x^*(A^* + A)x + x^*A^*Ax}{\beta^2x^*x + \beta x^*(A^* + A)x + x^*A^*Ax} \\
&= \frac{\beta^2}{\alpha^2}\max_{x\neq 0}\frac{\alpha^2 - 2\alpha t + \|A\|_2^2}{\beta^2 + 2\beta t + \|A\|_2^2}.
\end{aligned}$$

Let

$$f(t) = \frac{\alpha^2 - 2\alpha t + \|A\|_2^2}{\beta^2 + 2\beta t + \|A\|_2^2}.$$

Then

$$f'(t) = \frac{-2(\alpha + \beta)(\alpha\beta + \|A\|_2^2)}{(\beta^2 + 2\beta t + \|A\|_2^2)^2} < 0.$$

Thus, it holds that $\rho(M_{\alpha,\beta}) \leq \delta_{\alpha,\beta}$.

Clearly, $\delta_{\alpha,\beta} < 1$ is equal to

$$\alpha\beta\eta_l(H) + \|A\|_2^2(\alpha - \beta) > 0. \tag{2.7}$$

If $\alpha \geq \beta$, then (2.7) is true for any $\alpha > 0$, i.e., the GTSS method (2.3) is convergent; if $\eta_l(H) + \frac{\|A\|_2^2}{\beta} > \frac{\|A\|_2^2}{\alpha}$, then $\delta_{\alpha,\beta} < 1$ holds, i.e., the GTSS method (2.3) is convergent. ∎

When $\alpha = \beta$ in Theorem 2.1, the following corollary 2.2 is obtained.

Corollary 2.2. *Let* $A \in \mathbb{C}^{n\times n}$ *be non-Hermitian positive definite with* $\alpha > 0$. *Then the spectral radius* $\rho(M_\alpha)$ *of the iteration matrix* M_α *of the TSS(SS) method is bounded by*

$$\delta_\alpha = \sqrt{\frac{\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2}{\alpha^2 + 2\alpha\eta_l(H) + \|A\|_2^2}}, \tag{2.8}$$

where $\eta_l(H)$ is the smallest eigenvalue of the matrix $H = \frac{1}{2}(A + A^*)$.

Corollary 2.2 is a mail result in [5]. That is to say, $2\eta_l(H)$ in $\varrho_u(\alpha)$ in Theorem 2.2 [5] should be $2\alpha\eta_l(H)$.

Since $\beta^2 < \beta^2 + 2\beta\eta_l(H) + \|A\|_2^2$, from (2.6), we have

$$\frac{\beta}{\alpha}\sqrt{\frac{\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2}{\beta^2 + 2\beta\eta_l(H) + \|A\|_2^2}} < \frac{\sqrt{\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2}}{\alpha}.$$

Further, we have the following theorem.

THEOREM 2.3. *Let $A \in \mathbb{C}^{n\times n}$ be a non-Hermitian positive definite with $\alpha, \beta > 0$. Then the spectral radius $\rho(M_{\alpha,\beta})$ of the iteration matrix $M_{\alpha,\beta}$ of the GTSS method* (2.3) *is bounded by*

$$\bar{\delta}_\alpha = \frac{\sqrt{\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2}}{\alpha}, \tag{2.9}$$

where $\eta_l(H)$ is the smallest eigenvalue of the matrix $H = \frac{1}{2}(A + A^*)$. Moreover, if

$$\alpha \geq \frac{\|A\|_2^2}{2\eta_l(H)}, \tag{2.10}$$

then $\bar{\delta}_\alpha < 1$, which means that the GTSS method (2.3) *is convergent.*

PROOF. Here, we just need to prove $\bar{\delta}_\alpha < 1$, which is equal to

$$\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2 < \alpha^2.$$

Further,

$$\alpha \geq \frac{\|A\|_2^2}{2\eta_l(H)}.$$

The proof of Theorem 2.3 is completed. ∎

Next, the theoretical optimal parameter to minimize the upper bound $\bar{\delta}_\alpha$ is obtained.

COROLLARY 2.4. *Let the conditions of Theorem 2.3 be satisfied. Let $\eta_l(H)$ be the smallest eigenvalue of the matrix $H = \frac{1}{2}(A + A^*)$. Then*

$$\alpha^* = \frac{\|A\|_2^2}{\eta_l(H)}, \qquad \bar{\delta}_{\alpha^*} = \frac{\sqrt{\|A\|_2^2 - \eta_l^2(H)}}{\|A\|_2}.$$

PROOF. Let

$$g(\alpha) = \frac{\alpha^2 - 2\alpha\eta_l(H) + \|A\|_2^2}{\alpha^2}.$$

Simple calculation gives

$$g'(\alpha) = 2\frac{\alpha\eta_l(H) - \|A\|_2^2}{\alpha^3}.$$

It is obvious that $g'(\alpha) > 0$ for $\alpha > \frac{\|A\|_2^2}{\eta_l(H)}$ and $g'(\alpha) < 0$ for $\alpha < \frac{\|A\|_2^2}{\eta_l(H)}$. Hence, the upper bound $\bar{\delta}_\alpha$ of the spectral radius $\rho(M_{\alpha,\beta})$ achieves its minimum at $\alpha^* = \frac{\|A\|_2^2}{\eta_l(H)}$, and its minimum value of $\bar{\delta}_{\alpha^*}$ is obtained. ∎

3. Numerical experiments

In this section, we present numerical results to demonstrate the performance of the GTSS method (2.3) for solving the linear systems (1.1) on the basis of the following two examples. Numerical comparisons with the GTSS, TSS(SS) and HSS methods are also presented to show the advantages of the GTSS method, where the HSS method is of the form

$$\begin{cases} (\beta I + H)x^{(k+\frac{1}{2})} = (\beta I - S)x^{(k)} + b, \\ (\beta I + S)x^{(k+1)} = (\beta I - H)x^{(k+\frac{1}{2})} + b, \end{cases}$$

with $\beta > 0$, $H = \frac{1}{2}(A + A^*)$ and $S = \frac{1}{2}(A - A^*)$.

Example 3.1. We consider the two-dimensional convection-diffusion equation

$$-(u_{xx} + u_{yy}) + \gamma(u_x + u_y) = g,$$

where γ is a constant and g is a given function, and u satisfies Dirichlet boundary conditions in [8]. The following coefficient matrix

$$A = T \otimes I + I \otimes T$$

can be obtained by the five-point centered finite difference discretization on the unit square $[0, 1] \times [0, 1]$ with mesh-size $h = \frac{1}{m+1}$, where $\otimes$ denotes the Kronecker product, T is a tridiagonal matrix given by

$$T = \text{tridiag}(-1 - R_e, 2, -1 + R_e),$$

and $R_e = \frac{\gamma h}{2}$ is the mesh Reynolds number.

	β	0.05	0.1	0.2	0.3	0.4
GTSS	IT	7	9	12	19	31
	CPU	0.014	0.016	0.017	0.035	0.061
	RES	2.2292e-7	2.2487e-7	9.1099e-7	6.2602e-7	9.3442e-7
TSS	IT	–	373	187	125	94
	CPU	–	0.536	0.275	0.183	0.147
	RES		9.8509e-7	9.5615e-7	9.2682e-7	8.9718e-7
HSS	IT	–	279	143	98	74
	CPU	–	0.764	0.392	0.272	0.211
	RES		9.5485e-7	9.1787e-7	8.6613e-7	9.1156e-7

Table 1. IT, CPU and RES for GTSS, TSS(SS) and HSS with $n = 16^2$ and $\gamma = 10$ for Example 3.1.

	β	0.05	0.1	0.2	0.3	0.4
GTSS	IT	15	23	35	44	51
	CPU	0.084	0.128	0.191	0.234	0.270
	RES	9.5876e-7	8.8880e-7	9.1589e-7	8.9723e-7	8.8086e-7
TSS	IT	–	353	177	118	88
	CPU	–	1.800	0.897	0.608	0.450
	RES		9.7855e-7	9.4910e-7	9.4583e-7	9.9649e-7
HSS	IT	–	278	144	102	81
	CPU	–	2.637	1.358	0.970	0.775
	RES		9.8040e-7	9.2622e-7	9.0817e-7	9.6609e-7

Table 2. IT, CPU and RES for GTSS, TSS(SS) and HSS with $n = 32^2$ and $\gamma = 10$ for Example 3.1.

	β	0.05	0.1	0.2	0.3	0.4
GTSS	IT	6	8	13	19	32
	CPU	0.013	0.016	0.023	0.033	0.055
	RES	2.8001e-7	4.3173e-7	3.3476e-7	7.1884e-7	7.7294e-7
TSS	IT	–	385	193	128	96
	CPU	–	0.573	0.286	0.192	0.153
	RES		9.7739e-7	9.4929e-7	9.9936e-7	9.9494e-7
HSS	IT	–	309	157	107	81
	CPU	–	0.875	0.450	0.308	0.248
	RES		9.9697e-7	9.6736e-7	8.9043e-7	9.1950e-7

Table 3. IT, CPU and RES for GTSS, TSS(SS) and HSS with $n = 16^2$ and $\gamma = 15$ for Example 3.1.

	β	0.05	0.1	0.2	0.3	0.4
GTSS	IT	12	17	24	29	33
	CPU	0.067	0.092	0.130	0.153	0.177
	RES	4.5502e-7	4.8172e-7	6.2649e-7	7.3884e-7	8.0411e-7
TSS	IT	–	357	179	119	90
	CPU	–	1.818	0.911	0.608	0.462
	RES		9.8438e-7	9.5509e-7	9.7911e-7	8.9547e-7
HSS	IT	–	310	158	107	84
	CPU	–	2.904	1.492	1.008	0.787
	RES		9.6398e-7	9.7616e-7	9.5343e-7	8.6008e-7

Table 4. IT, CPU and RES for GTSS, TSS(SS) and HSS with $n = 32^2$ and $\gamma = 15$ for Example 3.1.

In our implementations, the initial guess is chosen to be $x^{(0)} = 0$ and the stopping criteria for the GTSS, TSS(SS) and HSS methods is

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

where 'RES' denotes the relative residual error. We compare three methods from aspects of the number of iterations, the relative residual error and the average value of the CPU times. Since the CPU time could be measured multiple times, in our computations, we investigate the average value of the CPU times in seconds (denoted as CPU) for three methods for 50 times. In this process, every time, the CPU time merely shows that the algorithm execution costs time when it is used to solve the corresponding non-Hermitian positive definite linear systems. All the test results are executed in MATLAB 7.0, which run on an Intel@ Celeron@ G4900, where the CPU 3.10GHz and the memory is 8.00 GB.

In our experiments, to test the convergence behavior of the GTSS, TSS(SS) and HSS methods, the value of γ is chosen to be $10, 15$. In this way, Tables 1-4 list the iteration numbers, the average value of the CPU times and the relative residual error for the GTSS, TSS(SS) and HSS methods under the same parameter β for the different problem sizes of n. The value of β is selected by the statement on the choice of the iteration parameter [6], that is, experience suggests that in most applications and for an appropriate scaling of the problem, a 'small' value of (usually between 0.01 and 0.5) may give good results. In the meantime, based on Theorem 2.1, the value of β satisfies $\beta \leq \alpha$. In our experiments, for convenience, we take $\alpha = 0.5$ for the GTSS method. In Tables 1-4, 'IT' denotes the number of iterations, 'CPU' denotes the average value of the CPU times, and '$-$' denotes the non-convergences of the TSS(SS) method and the HSS method when the number of iterations of the TSS(SS) method and the HSS method achieve 500.

From Tables 1-4, the presented results show that in all cases GTSS is superior to TSS(SS) and HSS in terms of the iteration numbers and the average value of the CPU times if it is appropriate to choose the value of the iteration parameters. This implies that the GTSS method may be given priority under certain conditions, compared with the TSS(SS) and HSS methods. In addition, we find that the variation trends of the iteration steps of the GTSS, TSS(SS) and HSS methods are that the iteration steps of the GTSS method may be decreased with γ increasing, whereas, the iteration steps of the TSS(SS) and HSS methods may be increased with γ increasing.

Example 3.2. ([1]) We consider the following non-Hermitian positive definite linear systems

$$\left[\left(K + \frac{3-\sqrt{3}}{\tau} I\right) + \left(K + \frac{3+\sqrt{3}}{\tau} I\right) i\right] x = b,$$

where $i = \sqrt{-1}$, $K = I \otimes V_m + V_m \otimes I$ with $V_m = h^{-2}\text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$ and the mesh-size $h = \frac{1}{m+1}$, τ is the time step-size, and b is composed of the entries $b_j = \frac{(1-i)j}{\tau(1+j)^2}, j = 1, 2, \ldots, n$. Here, matrix $K \in \mathbb{R}^{n \times n}$ with $n = m^2$ is from the five-point centered difference matrix approximating the negative Laplacian operator $L = -\Delta$ with homogeneous Dirichlet boundary conditions, on a uniform mesh in the unit square $[0, 1] \times [0, 1]$. In our tests, we take $\tau = h$ for Example 3.2.

	β	0.05	0.1	0.2	0.3	0.4
GTSS	IT	6	9	16	27	62
	CPU	0.016	0.023	0.039	0.065	0.139
	RES	9.9518e-7	5.0797e-7	4.2254e-7	9.9196e-7	9.0626e-7
TSS	IT	–	–	–	–	–
	CPU	–	–	–	–	–
	RES	–	–	–	–	–
HSS	IT	–	–	–	–	–
	CPU	–	–	–	–	–
	RES	–	–	–	–	–

TABLE 5. IT, CPU and RES for GTSS, TSS(SS) and HSS with $n = 16^2$ for Example 3.2.

	β	0.05	0.1	0.2	0.3	0.4
GTSS	IT	6	9	16	28	62
	CPU	0.055	0.087	0.145	0.252	0.537
	RES	9.9852e-7	5.1076e-7	4.2734e-7	6.0798e-7	9.5698e-7
TSS	IT	–	–	–	–	–
	CPU	–	–	–	–	–
	RES	–	–	–	–	–
HSS	IT	–	–	–	–	–
	CPU	–	–	–	–	–
	RES	–	–	–	–	–

TABLE 6. IT, CPU and RES for GTSS, TSS(SS) and HSS with $n = 32^2$ for Example 3.2.

In our computations, we extend the test method of Example 3.1 to Example 3.2, and give some numerical results for three testing methods in terms of the number of iterations, the relative residual error and the average value of the CPU times in seconds. Specifically, see Tables 5 and 6. From Tables 5 and 6, numerical results show that the GTSS method is convergent, whereas, the TSS(SS) and HSS methods are not convergent. This further confirms that the GTSS method overmatches the TSS(SS) and HSS methods in terms of the number of iterations and the average value of the CPU times. That is to say, the GTSS method for solving the non-Hermitian positive definite linear systems can be an attractive alternative to the original one.

4. Conclusions

In this paper, a generalized two-sweep shift splitting (GTSS) iteration method has been presented to solve the non-Hermitian positive definite linear systems on the base of the shift splitting of the coefficient matrix and its some convergence conditions under certain conditions are given. Numerical experiments show that the GTSS method for solving the non-Hermitian positive definite linear systems is feasible and effective.

Acknowledgements.
The authors would like to thank the handling editor and the anonymous referees for providing helpful suggestions, which greatly improved the paper. This research was supported by National Natural Science Foundation of China (No. 11961082).

References

[1] Z.-Z. Bai, M. Benzi and F. Chen, *Modified HSS iteration methods for a class of complex symmetric linear systems,* Computing **87** (2010), 93–111.
[2] Z.-Z. Bai and G. H. Golub, *Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems,* IMA J. Numer. Anal. **27** (2007), 1–23.
[3] Z.-Z. Bai, G. H. Golub and 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.
[4] Z.-Z. Bai, G. H. Golub and J.-Y. Pan, *Perconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems,* Numer. Math. **98** (2004), 1–32.
[5] Z.-Z. Bai, J.-F. Yin and Y.-F. Su, *A shift-splitting preconditioner for non-Hermitian positive definite matrices,* J. Comput. Math. **24** (2006), 539–552.
[6] M. Benzi, *A generalization of the Hermitian and skew-Hermitian splitting iteration,* SIAM J. Matrix Anal. Appl. **31** (2009), 360–374.
[7] A. Feriani, F. Perotti and V. Simoncini, *Iterative system solvers for the frequency analysis of linear mechanical systems,* Comput. Methods Appl. Mech. Eng. **190** (2000), 1719–1739.
[8] Y.-M. Huang, *A practical formula for computing optimal parameters in the HSS iteration methods,* J. Comput. Appl. Math. **255** (2014), 142–149.
[9] Y. Saad, Iterative methods for sparse linear systems, PWS Publishing Company, Boston, 1996.
[10] T. Sogabe and S.-L. Zhang, *A COCR method for solving complex symmetric linear systems,* J. Comput. Appl. Math. **199** (2007), 297–303.
[11] R. S. Varga, Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, NJ, 1962.

S.-L. Wu
School of Mathematics
Yunnan Normal University
Kunmin 650500, Yunnan
P.R. China
E-mail: `wushiliang1999@126.com`

C.-X. Li
School of Mathematics
Yunnan Normal University
Kunmin 650500, Yunnan
P.R. China
E-mail: `lixiatk@126.com`

Received: `23.4.2020.`
Revised: `23.11.2020.`