BIT Numer Math (2016) 56:543–556
DOI 10.1007/s10543-015-0584-7

The generalized HSS method with a flexible shift-parameter for non-Hermitian positive definite linear systems

Guo-Yan Meng[1] · Rui-Ping Wen[2] · Qing-Shan Zhao[3]

Received: 20 March 2015 / Accepted: 21 September 2015 / Published online: 13 October 2015

Abstract Based on the Hermitian and skew-Hermitian splitting (HSS), we come up with a generalized HSS iteration method with a flexible shift-parameter for solving the non-Hermitian positive definite system of linear equations. This iteration method utilizes the optimization technique to obtain the optimal value of the flexible shift-parameter at iteration process. Both theory and experiment have shown that the new strategy is efficient.

Keywords HSS method · Shift-parameter · Non-Hermitian positive definite matrix

Mathematics Subject Classification 65F10 · 65F50 · 15A06

1 Introduction

In this study we focus on solving the large sparse non-Hermitian and positive definite system of linear equations

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

✉ Rui-Ping Wen
wenrp@163.com

[1] Department of Mathematics, Xinzhou Normal University, Xinzhou 034000, Shanxi, People's Republic of China

[2] Higher Education Key Laboratory of Engineering and Scientific Computing in Shanxi Province, Taiyuan Normal University, Taiyuan 030012, Shanxi, People's Republic of China

[3] Department of Computer, Xinzhou Normal University, Xinzhou 034000, Shanxi, People's Republic of China

where $A \in \mathbb{C}^{n\times n}$ is nonsingular and $x, b \in \mathbb{C}^n$. Based on the *Hermitian/skew-Hermitian* (HS) splitting [25]

$$A = H + S$$

of the coefficient matrix A, where

$$H = \frac{1}{2}(A + A^*), \;\; S = \frac{1}{2}(A - A^*), \quad A^* \text{ is the conjugate transpose of } A,$$

Bai et al. [9] first proposed the efficient Hermitian and skew-Hermitian splitting (HSS) iteration method with a fixed shift-parameter in 2003 for solving the system (1.1). The method was designed as follows: Given an initial guess $x_0 \in \mathbb{C}^n$, for $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x_k\}_{k=0}^{\infty} \subset \mathbb{C}^n$ converges, compute the next iterate x_{k+1} according to the following formulation:

$$\begin{cases} (\alpha I + H)x_{k+\frac{1}{2}} = (\alpha I - S)x_k + b, \\ (\alpha I + S)x_{k+1} = (\alpha I - H)x_{k+\frac{1}{2}} + b, \end{cases} \tag{1.2}$$

where I is the identity matrix and α is a fixed shift-parameter (a given positive constant).

Note that the HSS iteration (1.2) may also be considered as a splitting iteration induced from the splitting of the matrix A as follows,

$$A = M(\alpha) - N(\alpha),$$

where

$$M(\alpha) = \frac{1}{2\alpha}(\alpha I + H)(\alpha I + S) \quad \text{and} \quad N(\alpha) = \frac{1}{2\alpha}(\alpha I - H)(\alpha I - S). \tag{1.3}$$

It was proved in [9] that the HSS iteration method converges unconditionally to the unique solution of the linear system (1.1), and the upper bound of the convergence rate of the method is about the same as that of the conjugate gradient method applied to a linear system of the coefficient matrix H. The optimal shift-parameter [9] is estimated as

$$\alpha_{\text{opt}} = \arg\min_{\alpha} \left\{ \max_{\lambda_{\min} \le \lambda \le \lambda_{\max}} \left| \frac{\alpha - \lambda}{\alpha + \lambda} \right| \right\} = \sqrt{\lambda_{\min}\lambda_{\max}},$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and the maximum eigenvalues of the matrix H, respectively.

Because of its outstanding performance and elegant mathematical properties, the HSS iteration method obtains widespread attention. For instance, it is extended to solve non-Hermitian positive semi-definite matrices [7,11], the saddle point problems [2,4,6,23], complex symmetric linear systems [5,16] and so on. Many modifications and generalizations of the HSS iteration method can be found in [1,10,12,18–21] and the references therein. On the other hand, some effective algorithms use the HSS iteration as a preconditioner or use the preconditioning variants of the HSS iteration, which often work out excellent results; see [13–15] and their references.

However, we have noticed that the shift-parameter α is a constant throughout the iteration process in all discussed above, which reduces the effectiveness of the methods from one point of view.

For the Hermitian positive definite linear system, Pearcy [22] has designed the device of changing iteration shift-parameters on the half-steps

$$\begin{cases} x_{k+\frac{1}{2}} = -\left(V_1 + \alpha_{k+\frac{1}{2}} D\right)^{-1} \left[(V_2 - \alpha_{k+\frac{1}{2}} D)x_k - b\right], \\ x_{k+1} = -(V_2 + \alpha_{k+1} D)^{-1} \left[(V_1 - \alpha_{k+1} D)x_{k+\frac{1}{2}} - b\right], \end{cases}$$

where D is a positive definite normalizing matrix, V_1, V_2 are Hermitian positive definite matrices such that $A = V_1 + V_2$, $\alpha_{\frac{1}{2}} \geq \alpha_1 \geq \alpha_{1+\frac{1}{2}} \geq \cdots \geq \alpha_{t-\frac{1}{2}} \geq \alpha_t$, and $\alpha_k = \alpha_{k(\mathrm{mod}\ t)}$ for $k > t$. But the convergence of this alternating direction implicit (ADI) iteration method depends on the positive definiteness of the splitting matrices V_1 and V_2. Furthermore, for the case $V_1 V_2 = V_2 V_1$ and $D = I$, Douglas [17] showed that ADI was always convergent and the problem of choosing an optimal parameter sequence [24] has also been solved for this case. Unfortunately, the problems of practical interest are usually with $V_1 V_2 \neq V_2 V_1$. So we propose the generalized HSS iteration method with a flexible shift-parameter. Moreover, motivated by the optimization models [26], shift-parameters $\alpha_k, k = 1, 2, \ldots$ are constructed by the minimization of residuals.

The remained of the paper is organized as follows. In Sect. 2, we describe the proposed iteration method. In Sect. 3, we use some numerical experiments to show the effectiveness of the new iteration method. Finally, we end the paper with a conclusion in Sect. 4.

2 The generalized HSS method with a flexible shift-parameter

In this section, we present a generalized HSS method to solve the system of linear equations (1.1) and it is stated as follows.

Method 2.1 (The generalized HSS method with a flexible shift-parameter) *Let* $x_0 \in \mathbb{C}^n$ *be an arbitrary initial guess and* $\varepsilon > 0$ *be a given tolerance. The splitting matrices* $M(\alpha)$ *and* $N(\alpha)$ *are defined by* (1.3). *For* $k = 0, 1, 2, \ldots$ *until the sequence of iterates* $\{x_k\}_{k=0}^{\infty} \subset \mathbb{C}^n$ *converges, compute the next iteration* x_{k+1} *according to the following procedure.*

(1) *Compute* $r_k = b - Ax_k$.
(2) *Solve the following system of linear equations:*

$$\begin{cases} (\alpha_{k+1} I + H)x_{k+\frac{1}{2}} = (\alpha_{k+1} I - S)x_k + b, \\ (\alpha_{k+1} I + S)x_{k+1} = (\alpha_{k+1} I - H)x_{k+\frac{1}{2}} + b, \end{cases} \tag{2.1}$$

where α_{k+1} is the solution of the following optimization problem

$$\min_{\alpha} r_{k+1}^* (\alpha I - H)^{-2} r_{k+1}, \tag{2.2}$$

here, $r_{k+1} = N(\alpha)M(\alpha)^{-1}r_k$.

(3) *If $\|r_{k+1}\|_2 \le \varepsilon$, stop; otherwise, $k \Leftarrow k+1$ and go to (1).*

If $\alpha = \alpha_1 = \alpha_2 = \cdots$ in (2.1), Method 2.1 reduces to the HSS iteration method with a fixed shift-parameter. So we only consider the situation that the shift-parameters α_k, $k = 1, 2, \ldots$ are not all equal, which is said the generalized HSS method with a flexible shift-parameter.

In matrix-vector form, the scheme (2.1) can be equivalently written as

$$x_{k+1} = T_{\alpha_{k+1}} x_k + G_{\alpha_{k+1}} b, \quad k = 0, 1, 2, \ldots,$$

where

$$T_{\alpha_{k+1}} = (\alpha_{k+1} I + S)^{-1}(\alpha_{k+1} I - H)(\alpha_{k+1} I + H)^{-1}(\alpha_{k+1} I - S) \tag{2.3}$$

and

$$G_{\alpha_{k+1}} = 2\alpha_{k+1}(\alpha_{k+1} I + S)^{-1}(\alpha_{k+1} I + H)^{-1}.$$

Evidently, we can express x_{k+1} as

$$x_{k+1} = Q_{k+1} x_0 + c_{k+1}, \quad k = 0, 1, 2, \ldots,$$

where

$$Q_{k+1} = T_{\alpha_{k+1}} T_{\alpha_k} \cdots T_{\alpha_1} \tag{2.4}$$

and

$$c_{k+1} = \left(G_{\alpha_{k+1}} + T_{\alpha_{k+1}} G_{\alpha_k} + T_{\alpha_{k+1}} T_{\alpha_k} G_{\alpha_{k-1}} + \cdots + T_{\alpha_{k+1}} T_{\alpha_k} \cdots T_{\alpha_2} G_{\alpha_1}\right) b.$$

Lemma 2.1 *Let x_k be generated by Method 2.1, $M(\alpha)$ and $N(\alpha)$ be given by (1.3). Then at the (k+1)-st step it holds that*

$$\left\| (\alpha I - H)^{-1} r_{k+1} \right\|_2^2 = r_k^* (\alpha I + H)^{-2} r_k, \tag{2.5}$$

where α is obtained by the optimization model (2.2). Furthermore,

$$\frac{d}{d\alpha}\left(\left\| (\alpha I - H)^{-1} r_{k+1} \right\|_2^2 \right) = -2 r_k^* (\alpha I + H)^{-3} r_k. \tag{2.6}$$

Proof Let $\Phi(\alpha) = (\alpha I - S)(\alpha I + S)^{-1}$, we see that $\Phi(\alpha)^*\Phi(\alpha) = I$, which means that $\Phi(\alpha)$ is a unitary matrix. Since

$$(\alpha I - H)^{-1}N(\alpha)M(\alpha)^{-1} = (\alpha I - S)(\alpha I + S)^{-1}(\alpha I + H)^{-1} = \Phi(\alpha)(\alpha I + H)^{-1},$$

we have

$$\begin{aligned}\left\|(\alpha I - H)^{-1}r_{k+1}\right\|_2^2 &= \left((\alpha I - H)^{-1}N(\alpha)M(\alpha)^{-1}r_k, (\alpha I - H)^{-1}N(\alpha)M(\alpha)^{-1}r_k\right)\\ &= \left(\Phi(\alpha)(\alpha I + H)^{-1}r_k, \Phi(\alpha)(\alpha I + H)^{-1}r_k\right)\\ &= r_k^*(\alpha I + H)^{-2}r_k.\end{aligned}$$

Note that

$$\frac{d\left((\alpha I + H)^{-1}\right)}{d\alpha} = -(\alpha I + H)^{-1}\frac{d\,(\alpha I + H)}{d\alpha}(\alpha I + H)^{-1} \quad \text{and} \quad \frac{d\,(\alpha I + H)}{d\alpha} = I.$$

Then,

$$\begin{aligned}&\frac{d}{d\alpha}\left(\left\|(\alpha I - H)^{-1}r_{k+1}\right\|_2^2\right)\\ &= r_k^*\frac{d\left((\alpha I + H)^{-1}\right)}{d\alpha}(\alpha I + H)^{-1}r_k + r_k^*(\alpha I + H)^{-1}\frac{d\left((\alpha I + H)^{-1}\right)}{d\alpha}r_k\\ &= -2r_k^*(\alpha I + H)^{-3}r_k.\end{aligned}$$

□

Theorem 2.1 *Let A be a non-Hermitian positive definite matrix. Then the iteration sequence $\{x_k\}$ generated by Method 2.1 converges to the unique solution x_* of the system of linear equations (1.1). Furthermore, if A is a normal matrix, then the 2-norm of the error vector $e_k = x_k - x_*$ is strictly decreasing, i.e., $\|e_{k+1}\|_2 < \|e_k\|_2$, $k = 0, 1, 2, \ldots$.*

Proof Let $e_k = x_k - x_*$ be the error at the kth step of Method 2.1. If α_{k+1} is obtained by the minimization model (2.2), then for any $\alpha_k > 0$ it holds that

$$r_{k+1}^*(\alpha_{k+1}I - H)^{-2}r_{k+1} = r_k^*(\alpha_{k+1}I + H)^{-2}r_k \le r_k^*(\alpha_k I + H)^{-2}r_k. \quad (2.7)$$

From (2.7), we have

$$\begin{aligned}\left\|(\alpha_{k+1}I - H)^{-1}r_{k+1}\right\|_2 &\le \left\|(\alpha_k I + H)^{-1}r_k\right\|_2\\ &= \left\|(\alpha_k I + H)^{-1}(\alpha_k I - H)(\alpha_k I - H)^{-1}r_k\right\|_2\\ &\le \left\|(\alpha_k I + H)^{-1}(\alpha_k I - H)\right\|_2\left\|(\alpha_k I - H)^{-1}r_k\right\|_2\end{aligned}$$

$$= \max_{\lambda_i \in \lambda(H)} \left| \frac{\alpha_k - \lambda_i}{\alpha_k + \lambda_i} \right| \left\| (\alpha_k I - H)^{-1} r_k \right\|_2 .$$

It is noted that there exists a positive number γ such that for any $\alpha_k > 0$ the inequality $\max_{\lambda_i \in \lambda(H)} \left| \frac{\alpha_k - \lambda_i}{\alpha_k + \lambda_i} \right| \leq 1 - \gamma$ is valid. Thus we have

$$\begin{aligned} \left\| (\alpha_{k+1} I - H)^{-1} r_{k+1} \right\|_2 &\leq (1-\gamma) \left\| (\alpha_k I - H)^{-1} r_k \right\|_2 \leq \cdots \\ &\leq (1-\gamma)^{k+1} \left\| (\alpha_0 I - H)^{-1} r_0 \right\|_2 . \end{aligned}$$

Moreover,

$$\lim_{k \to \infty} (\alpha_{k+1} I - H)^{-1} r_{k+1} = 0.$$

For any $\alpha > 0$, the matrix $(\alpha I - H)^{-1}$ is of full rank. Hence, we have $\lim\limits_{k \to \infty} r_{k+1} = 0$.

On the other hand, if A is a normal matrix, we get

$$HS = SH.$$

Thus, $T(\alpha) = (\alpha I + S)^{-1}(\alpha I - S)(\alpha I - H)(\alpha I + H)^{-1}$ is also a normal matrix, which implies that

$$\|T(\alpha)\|_2 = \rho(T(\alpha)).$$

Therefore, for the iteration matrix Q_{k+1} in (2.4), it holds that

$$\|Q_{k+1}\|_2 \leq \|T_{\alpha_{k+1}}\|_2 \|T_{\alpha_k}\|_2 \cdots \|T_{\alpha_1}\|_2 = \rho\left(T_{\alpha_{k+1}}\right) \rho\left(T_{\alpha_k}\right) \cdots \rho\left(T_{\alpha_1}\right) < 1.$$

Moreover,

$$\|Q_{k+1}\|_2 = \|T_{\alpha_{k+1}} Q_k\|_2 < \|Q_k\|_2 < 1.$$

So,

$$\|e_{k+1}\|_2 = \|Q_{k+1} e_0\|_2 = \|T_{\alpha_{k+1}} Q_k e_0\|_2 < \|Q_k e_0\|_2 = \|e_k\|_2.$$

□

It is worthy to note that the solution of the minimization model (2.2) is equivalent to compute the root of $\frac{d}{d\alpha}\left(\|(\alpha I - H)^{-1} r_{k+1}\|_2^2\right) = 0$ in Theorem 2.1. However, the computational formula (2.6) is only of theoretical meaning and it is far away from actual applications, since the computational cost of the matrix $(\alpha I + H)^{-3}$ is expensive. Instead, we approximate the root of $f(\alpha) = \|(\alpha I - H)^{-1} r_{k+1}\|_2^2 = 0$ by the Newton method. An alternative procedure might be to approximate the value of $f(\alpha) = \|(\alpha I + H)^{-1} r_k\|_2^2 = 0$ by (2.5). Hence, Method 2.1 could be rewritten into a practical form stated in Method 2.2.

Method 2.2 (The practical generalized HSS method with a flexible shift-parameter) *Let $x_0 \in \mathbb{C}^n$ be an arbitrary initial guess and $\varepsilon > 0$ be a given precision. The splitting matrices $M(\alpha)$ and $N(\alpha)$ are defined by (1.3). For $k = 0, 1, 2, \ldots$ until the sequence of iterates $\{x_k\}_{k=0}^{\infty} \subset \mathbb{C}^n$ converges, compute the next iteration x_{k+1} according to the following procedure.*

(1) *Compute $r_k = b - Ax_k$.*
(2) *Solve the systems of linear equations:*

$$\begin{cases} (\alpha_{k+1} I + H)x_{k+\frac{1}{2}} = (\alpha_{k+1} I - S)x_k + b, \\ (\alpha_{k+1} I + S)x_{k+1} = (\alpha_{k+1} I - H)x_{k+\frac{1}{2}} + b, \end{cases}$$

where α_{k+1} is the root of the equation

$$f(\alpha) = \|(\alpha I + H)^{-1} r_k\|_2^2 = 0.$$

(3) *If $\|r_{k+1}\|_2 \leq \varepsilon$, stop; otherwise, $k \Leftarrow k + 1$ and go to (1).*

Remark 2.1 For the sake of saving computational cost, we can perform a new update on the shift-parameter α for every p iteration steps.

3 Numerical experiments

In this section, we provide numerical results to illustrate the effectiveness of Methods 2.1 and 2.2 in terms of the iteration steps (denoted as **IT**), the elapsed computing times in seconds (denoted as **CPU**), and the relative residual error (denoted as **RES**) defined by

$$\text{RES} = \frac{\|b - Ax_k\|_2}{\|b - Ax_0\|_2}.$$

All tests are started from the vector $x_0 = 0$, and terminated when the current iteration satisfies RES$\leq 10^{-6}$. In addition, the numerical experiments are performed in MATLAB (version R2013a) on PC in double precision, which is 2.40 GHz central processing unit [Intel(R) Core(TM)i7-4500 CPU] with 8G memory and Microsoft Window 8 operating system. In actual computation, the α_k is updated for every 5 iteration steps.

Example 3.1 [8] Consider the two-dimensional convection-diffusion equation

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

on the unit square $(0, 1) \times (0, 1)$ with constant coefficient β and subject to Dirichlet-type boundary condition. By applying the five-point centered finite difference discretization, we get the system of linear equations (1.1) with the coefficient matrix

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

where $h = \frac{1}{m+1}$ is the equidistant step-size (In this test, $m = 32$.), $\otimes$ denotes the Kronecker product, and T is a tridiagonal matrix given by

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

where

$$R_e = \frac{\beta h}{2} \tag{3.1}$$

is the mesh Reynolds number. Moreover, the right-hand side vector b is taken to be $b = Ax_*$ with $x_* = (1, 1, \dots, 1)^T \in \mathbb{R}^n$ being the exact solution.

Here and in the sequel, BGN-HSS means the HSS method by Bai et al. [9], and H-HSS means the new practical methods by Huang [19]. Correspondingly, Methods 2.1 and 2.2 are termed briefly as MWZ$_1$-HSS, and MWZ$_2$-HSS, respectively.

In Table 1, we give the iteration numbers, the CPU times and the RES values of BGN-HSS, H-HSS, MWZ$_1$-HSS and MWZ$_2$-HSS methods for Example 3.1 with respect to different choices of the problem parameter β.

From Table 1, it can be seen that for different β the numbers of iteration steps of MWZ$_1$-HSS and MWZ$_2$-HSS methods are less than those of BGN-HSS and H-HSS methods at almost the same CPU times. The case $\beta = 50$ and $\beta = 5000$ of these observations can be further illustrated by the iteration pictures plotted in Fig. 1. Clearly, MWZ$_2$-HSS method outperforms both BGN-HSS and H-HSS methods.

Table 1 Iteration Step, CPU Time, and RES of BGN-HSS, H-HSS, MWZ$_1$-HSS and MWZ$_2$-HSS methods for Example 3.1

Method	β					
	50	100	500	1000	5000	10000
BGN-HSS						
IT	45	46	56	74	155	216
CPU	0.20	0.22	0.25	0.32	0.67	0.94
RES	8.61e−7	6.70e−7	9.03e−7	8.89e−7	9.66e−7	9.98e−7
H-HSS						
IT	66	35	50	70	163	252
CPU	0.29	0.15	0.22	0.31	0.71	1.10
RES	9.57e−7	7.53e−7	9.88e−7	9.83e−7	9.59e−7	9.81e−7
MWZ$_1$-HSS						
IT	32	41	58	70	113	115
CPU	0.81	1.03	1.44	1.73	2.80	2.84
RES	9.58e−7	8.70e−7	9.26e−7	9.72e−7	9.96e−7	9.93e−7
MWZ$_2$-HSS						
IT	38	43	56	62	83	92
CPU	0.19	0.21	0.27	0.30	0.41	0.45
RES	4.28e−7	9.56e−7	9.77e−7	9.77e−7	9.29e−7	9.83e−7

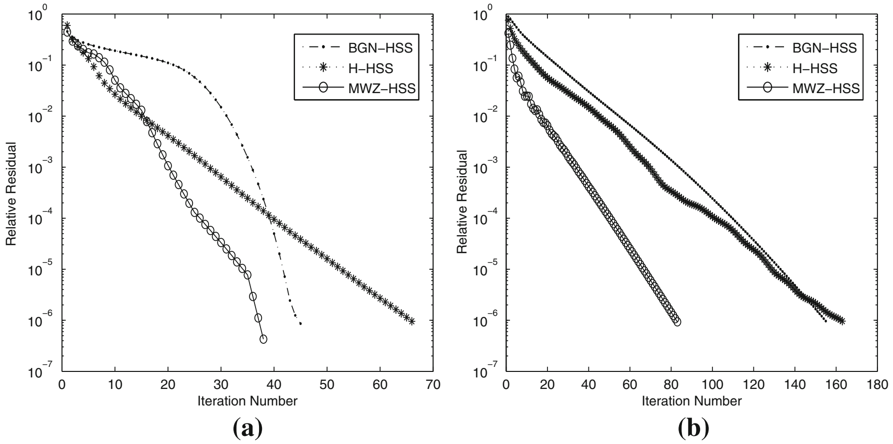


Fig. 1 Relative residual versus iteration step for the BGN-HSS, H-HSS and MWZ$_2$-HSS methods with $m = 32$

Example 3.2 [8] We solve the linear system $Ax = b$, where A is a 2-by-2 block matrix as follows:

$$A = \begin{pmatrix} B & E \\ -E^T & 0.5I \end{pmatrix},$$

where

$$B = \begin{pmatrix} I \otimes T_H + T_H \otimes I & 0 \\ 0 & I \otimes T_H + T_H \otimes I \end{pmatrix} \in \mathbb{R}^{2m^2 \times 2m^2}, \; E = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \mathbb{R}^{2m^2 \times m^2},$$

with

$$T_H = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{m \times m}, \quad F = \delta h \cdot \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{m \times m},$$

and $h = \frac{1}{m+1}$ being the discretization mesh-size.

In this example, the total number of variables is $3m^2$, and the right-hand side vector b is taken to be $b = Ax_*$ with $x_* = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$ being the exact solution.

In Table 2, we provide the iteration numbers, the CPU times and the RES values of BGN-HSS, H-HSS, MWZ$_1$-HSS and MWZ$_2$-HSS methods for Example 3.2 with respect to different choices of the problem parameter δ as well as the problem size m. Clearly, the iteration steps of MWZ$_2$-HSS method is much less than those of H-HSS and BGN-HSS methods. For the case $\delta = 1000$, when the problem size m becomes large the iteration steps of MWZ$_2$-HSS method unchange but those of BGN-HSS and H-HSS methods are increasing. As for the computing time, the situation is almost the same except for MWZ$_1$-HSS method.

Table 2 Iteration Step, CPU Time, and RES of BGN-HSS, H-HSS, MWZ$_1$-HSS and MWZ$_2$-HSS methods for Example 3.2

Method	m	$\delta = 10$			$\delta = 100$			$\delta = 1000$		
		IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
BGN-HSS	16	83	1.08	9.41e−7	44	0.10	9.61e−7	44	0.10	9.99e−7
	24	76	0.39	9.74e−7	64	0.33	8.83e−7	64	0.33	9.05e−7
	32	101	0.96	9.54e−7	83	0.80	9.41e−7	83	0.80	9.40e−7
H-HSS	16	172	0.39	9.94e−7	56	0.13	9.55e−7	52	0.12	9.10e−7
	24	369	1.89	9.82e−7	108	0.56	9.23e−7	85	0.44	9.44e−7
	32	631	6.13	9.99e−7	179	1.74	9.48e−7	130	1.25	9.50e−7
MWZ$_1$-HSS	16	83	1.08	9.41e−7	27	0.36	9.71e−7	37	0.49	8.61e−7
	24	192	5.29	9.59e−7	56	1.57	8.86e−7	45	1.28	9.81e−7
	32	336	20.10	9.80e−7	99	5.80	9.29e−7	62	3.62	9.53e−7
MWZ$_2$-HSS	16	44	0.11	8.86e−7	36	0.09	7.62e−7	40	0.10	7.25e−7
	24	58	0.33	9.35e−7	45	0.24	7.80e−7	40	0.24	8.42e−7
	32	69	0.86	9.69e−7	61	0.74	8.38e−7	42	0.52	8.64e−7

Furthermore, we use the following singular linear system to further examine the effectiveness of MWZ$_2$-HSS method. Note that H-HSS method fails for this example. The generalized HSS method proposed by Li et al. [20] (as the above manner, denoted as LLP-GHSS) could be applied for comparison.

Example 3.3 [3] Consider the two-dimensional variable-coefficient second-order differential equation satisfying the periodic boundary condition given as follows:

$$\begin{cases} -\frac{\partial}{\partial x}\left(c(x,y)\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(c(x,y)\frac{\partial u}{\partial y}\right) + \gamma\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) = f(x,y), & (x,y) \in (0,1)\times(0,1), \\ u(x,0) = u(x,1), & x \in (0,1), \\ u(0,y) = u(1,y), & y \in (0,1). \end{cases} \tag{3.2}$$

The right-hand side vector is taken to be $b = Ax_*$ with $x_* = (1, 2, \ldots, n)^T$ being the exact solution.

When $c(x, y) \equiv 1$, the differential equation (3.2) is discretized by using the same approach as Example 3.1, with the equidistant stepsize $h = \frac{1}{m}$ and $n = m^2$. So we can get the real linear system with the singular and positive semi-definite n-by-n coefficient matrix

$$A = \begin{pmatrix} \triangle & -(1-R_e)I & 0 & \cdots & 0 & -(1+R_e)I \\ -(1+R_e)I & \triangle & -(1-R_e)I & \cdots & 0 & 0 \\ 0 & -(1+R_e)I & \triangle & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \triangle & -(1-R_e)I \\ -(1-R_e)I & 0 & 0 & \cdots & -(1+R_e)I & \triangle \end{pmatrix} \in \mathbb{R}^{n\times n},$$

Table 3 Iteration Step, CPU Time, and RES of BGN-HSS, H-HSS, MWZ$_1$-HSS and MWZ$_2$-HSS methods for Example 3.3

m	Method	β					
		5	10	10^2	10^3	10^4	10^5
16	BGN-HSS						
	IT	43	44	48	48	48	48
	CPU	0.05	0.05	0.06	0.06	0.06	0.12
	RES	8.01e−7	8.62e−7	8.09e−7	8.53e−7	8.53e−7	8.54e−7
	LLP-GHSS						
	IT	24	40	34	32	32	32
	CPU	0.03	0.05	0.04	0.04	0.04	0.07
	RES	7.77e−7	9.50e−7	7.75e−7	9.99e−7	9.93e−7	9.93e−7
	MWZ$_1$-HSS						
	IT	196	195	174	141	118	94
	CPU	0.71	0.71	0.66	0.62	0.74	0.74
	RES	9.59e−7	9.81e−7	9.51e−7	9.86e−7	9.68e−7	9.50e−7
	MWZ$_2$-HSS						
	IT	21	20	26	26	26	26
	CPU	0.04	0.03	0.04	0.04	0.05	0.07
	RES	6.87e−7	9.66e−7	7.31e−7	6.65e−7	6.61e−7	6.61e−7
24	BGN-HSS						
	IT	61	63	70	70	70	70
	CPU	0.17	0.18	0.21	0.22	0.22	0.57
	RES	8.42e−7	8.41e−7	8.91e−7	9.94e−7	9.95e−7	9.95e−7
	LLP-GHSS						
	IT	34	59	52	48	48	48
	CPU	0.10	0.18	0.16	0.15	0.15	0.37
	RES	7.85e−7	9.53e−7	8.44e−7	8.01e−7	7.88e−7	7.88e−7
	MWZ$_1$-HSS						
	IT	184	184	170	136	110	88
	CPU	1.77	1.80	1.65	1.59	1.76	2.04
	RES	9.93e−7	9.79e−7	9.83e−7	9.52e−7	9.91e−7	9.42e−7
	MWZ$_2$-HSS						
	IT	25	25	27	26	26	26
	CPU	0.09	0.09	0.11	0.09	0.11	0.22
	RES	7.38e−7	9.55e−7	4.54e−7	8.34e−7	8.11e−7	8.11e−7
32	BGN-HSS						
	IT	78	80	91	93	93	93
	CPU	0.43	0.45	0.53	0.54	0.54	1.81
	RES	8.72e−7	9.63e−7	9.88e−7	8.93e−7	8.95e−7	8.95e−7

Table 3 continued

m	Method	β					
		5	10	10^2	10^3	10^4	10^5
	LLP-GHSS						
	IT	44	77	71	63	63	63
	CPU	0.28	0.44	0.42	0.37	0.37	1.19
	RES	7.74e−7	9.48e−7	9.21e−7	8.43e−7	8.19e−7	8.19e−7
	MWZ$_1$-HSS						
	IT	177	177	167	130	106	88
	CPU	3.38	3.46	3.39	3.14	3.66	4.23
	RES	9.87e−7	9.78e−7	9.52e−7	9.91e−7	9.65e−7	9.31e−7
	MWZ$_2$-HSS						
	IT	31	32	34	43	43	43
	CPU	0.22	0.22	0.26	0.29	0.44	0.82
	RES	9.58e−7	5.62e−7	5.90e−7	4.17e−7	3.56e−7	3.55e−7

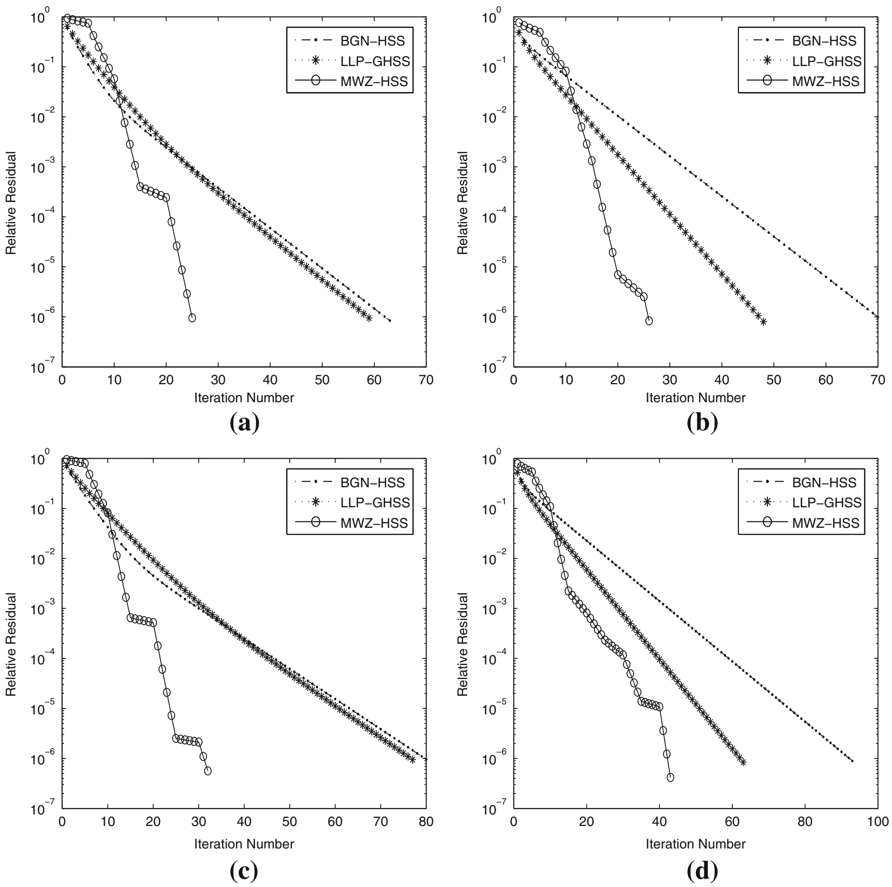
where $\triangle \in \mathbb{R}^{m\times m}$ is given by

$$\begin{pmatrix} 4 & -1+R_e & 0 & \cdots & 0 & -1-R_e \\ -1-R_e & 4 & -1+R_e & \cdots & 0 & 0 \\ 0 & -1-R_e & 4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 4 & -1+R_e \\ -1+R_e & 0 & 0 & \cdots & -1-R_e & 4 \end{pmatrix},$$

and R_e is defined by (3.1).

In Table 3, we list the numbers of iteration steps, the CPU times and the RES values of BGN-HSS, LLP-HSS, MWZ$_1$-HSS and MWZ$_2$-HSS methods for Example 3.3 with respect to different choices of the problem parameter β as well as the problem size m.

From Table 3, we see that MWZ$_1$-HSS method is not sensitive to the shift-parameter α_{k+1}. In this table, all methods can get a satisfactory approximation for solving this singular linear system. Evidently, the number of iteration steps tends to a constant when β becomes large. Table 3 shows that MWZ$_2$-HSS method requires less iteration steps and computing times than BGN-HSS and LLP-GHSS methods. It is also noted that MWZ$_2$-HSS method needs to compute the updated optimal shift-parameter for every 5 iteration steps, while BGN-HSS and LLP-GHSS methods do not require this additional computation. In Fig. 2, we further show the numerical advantages of MWZ$_2$-HSS method over the other three methods for different values of β and n.



(a) (b) (c) (d)

Fig. 2 Relative residual versus iteration number for the BGN-HSS, LLP-HSS and MWZ$_2$-HSS iteration methods with respect to the different n

4 Conclusion

In this paper, we propose a generalized HSS iteration method for solving the non-Hermitian positive definite system of linear equations. The numerical experiments show that the proposed MWZ$_2$-HSS method is superior to the methods given in [9,19,20]. Even though the coefficient matrix is singular and non-Hermitian positive semi-definite, the new iteration method can gain almost the same excellent properties as the HSS method.

Acknowledgments The authors are very much indebted to the anonymous referees for their helpful comments and suggestions which greatly improved the original manuscript of this paper. This work is supported by the National Natural Science Foundation of China (No.11371275), the National Natural Science Foundation of Shanxi Province, China (No.2014011019-3) and the Key Construction Disciplines Project of Xinzhou Normal University, China (No.XK201301).

References

1. Bai, Z.-Z.: Splitting iteration methods for non-Hermitian positive definite systems of linear equations. Hokkaido Math. J. **36**, 801–814 (2007)
2. Bai, Z.-Z.: Optimal parameters in the HSS-like methods for saddle-point problems. Numer. Linear Algebra Appl. **16**, 447–479 (2009)
3. Bai, Z.-Z.: On semi-convergence of Hermitian and skew-Hermitian splitting methods for singular linear systems. Computing **89**, 171–197 (2010)
4. Bai, Z.-Z.: Block alternating splitting implicit iteration methods for saddle-point problems from time-harmonic eddy current models. Numer. Linear Algebra Appl. **19**, 914–936 (2012)
5. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
6. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**, 1–23 (2007)
7. Bai, Z.-Z., Golub, G.H., Li, C.-K.: Convergence properties of preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite matrices. Math. Comput. **76**, 287–298 (2007)
8. Bai, Z.-Z., Golub, G.H., Li, C.-K.: Optimal parameter in Hermitian and skew-Hermitian splitting method for certain two-by-two block matrices. SIAM J. Sci. Comput. **28**, 583–603 (2006)
9. Bai, Z.-Z., Golub, G.H., Ng, M.K.: Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. SIAM J. Matrix Anal. Appl. **24**, 603–626 (2003)
10. Bai, Z.-Z., Golub, G.H., Ng, M.K.: On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems. Linear Algebra Appl. **428**, 413–440 (2008)
11. Bai, Z.-Z., Golub, G.H., Pan, J.-Y.: Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems. Numer. Math. **98**, 1–32 (2004)
12. Benzi, M.: A generalization of the Hermitian and skew-Hermitian splitting iteration. SIAM J. Matrix Anal. Appl. **31**, 360–374 (2009)
13. Bertaccini, D., Golub, G.H., Capizzano, S.S., Possio, C.T.: Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation. Numer. Math. **99**, 441–484 (2005)
14. Chan, L.-C., Ng, M.K., Tsing, N.-K.: Spectral analysis for HSS preconditioners. Numer. Math. Theor. Meth. Appl. **1**, 57–77 (2008)
15. Chen, F., Jiang, Y.-L.: On HSS and AHSS iteration methods for nonsymmetric positive definite Toeplitz systems. J. Comput. Appl. Math. **234**, 2432–2440 (2010)
16. Chen, F., Liu, Q.-Q.: On semi-convergence of modified HSS iteration methods. Numer. Algorithms **64**, 507–518 (2013)
17. Douglas, J.: On the numerical integration $\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=\frac{\partial u}{\partial t}$ by implicit methods. J. Soc. Indust. Appl. Math. **3**, 42–65 (1955)
18. Guo, X.-X., Wang, S.: Modified HSS iteration methods for a class of non-Hermitian positive-definite linear systems. Appl. Math. Comput. **218**, 10122–10128 (2012)
19. Huang, Y.-M.: A practical formula for computing optimal parameters in the HSS iteration methods. J. Comput. Appl. Math. **255**, 142–149 (2014)
20. Li, W., Liu, Y.-P., Peng, X.-F.: The generalized HSS method for solving singular linear systems. J. Comput. Appl. Math. **236**, 2338–2353 (2012)
21. Li, W.-W., Wang, X.: A modified GPSS method for non-Hermitian positive definite linear systems. Appl. Math. Comput. **234**, 253–259 (2014)
22. Pearcy, C.: On convergence of alternating direction procedures. Numer. Math. **4**, 172–176 (1962)
23. Simoncini, V., Benzi, M.: Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 377–389 (2004)
24. Varga, R.S.: Matrix Iterative Analysis, 2nd edn. Springer, Berlin (2000)
25. Wang, C.-L., Bai, Z.-Z.: Sufficient conditions for the convergent splittings of non-Hermitian positive definite matrices. Linear Algebra Appl. **330**, 215–218 (2001)
26. Wang, C.-L., Meng, G.-Y., Bai, Y.-H.: Practical convergent splittings and acceleration methods for non-Hermitian positive definite linear systems. Adv. Comput. Math. **39**, 257–271 (2013)