BIT Numerical Mathematics
https://doi.org/10.1007/s10543-018-0729-6

BIT

Minimum residual Hermitian and skew-Hermitian splitting iteration method for non-Hermitian positive definite linear systems

Ai-Li Yang[1] · **Yang Cao**[2] · **Yu-Jiang Wu**[1]

Received: 6 April 2017 / Accepted: 1 October 2018

Abstract
By applying the minimum residual technique to the Hermitian and skew-Hermitian splitting (HSS) iteration scheme, we introduce a non-stationary iteration method named minimum residual HSS (MRHSS) iteration method to solve non-Hermitian positive definite linear systems. The convergence property of the MRHSS iteration method together with the property of the iteration parameters are carefully studied. Numerical results verify the effectiveness and robustness of the MRHSS iteration method.

Keywords Hermitian and skew-Hermitian splitting · Minimum residual · Convergence property · Iteration parameter

Mathematics Subject Classification 65F10 · 65F15 · 65F35

1 Introduction

Many problems in scientific computing give rise to a system of linear equations

$$Ax = b, \quad A \in \mathbb{C}^{n\times n},\ x, b \in \mathbb{C}^n, \tag{1}$$

where A is a positive definite matrix.

Communicated by Lars Eldén.

✉ Ai-Li Yang
yangaili@lzu.edu.cn; cmalyang@gmail.com

[1] School of Mathematics and Statistics, Lanzhou University, Lanzhou 730000, Gansu, People's Republic of China

[2] School of Transportation, Nantong University, Nantong 226019, Jiangsu, People's Republic of China

Published online: 08 October 2018

When the coefficient matrix A is large and sparse, iteration methods are more attractive than direct methods for solving the system of linear equations (1) since the cost of direct methods is prohibitively high. Moreover, the direct methods may not stable because of the accumulation of round-off error. As a kind of popular iteration methods, Krylov subspace methods obtained wide attention in last few decades. Although they possess good convergence properties, the storage space needed by Krylov subspace methods always increases very fast with the increase of the problem size [38]. So, in this work we focus on another kind of iteration methods, i.e., the matrix splitting iteration methods.

Let $A = H(A) + S(A)$ be the Hermitian and skew-Hermitian splitting (HSS) of matrix A, where

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

are the Hermitian and the skew-Hermitian parts of matrix A, respectively. Bai, Golub and Ng introduced an HSS iteration method in Ref. [15] to approximate the solution of the system of linear equations (1). The iteration scheme of the HSS iteration method has the form of

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

where $k = 0, 1, 2, \ldots$, parameter α is a given positive constant, I and $x^{(0)}$ are an identity matrix and an arbitrary initial guess, respectively. Using

$$(\alpha I + H(A))^{-1}(\alpha I - S(A)) = I - (\alpha I + H(A))^{-1}A$$

and

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

the HSS iteration scheme (2) can be equivalently rewritten as

$$\begin{cases} x^{(k+\frac{1}{2})} &= x^{(k)} + (\alpha I + H(A))^{-1}(b - Ax^{(k)}), \\ x^{(k+1)} &= x^{(k+\frac{1}{2})} + (\alpha I + S(A))^{-1}(b - Ax^{(k+\frac{1}{2})}). \end{cases} \tag{3}$$

Although the iteration scheme (3) is mathematically equivalent to the HSS iteration scheme (2), the computing efficiencies of the two iteration schemes are different with each other in actual implementations. In general, the iteration scheme (3) always achieves higher computational efficiency than the HSS iteration scheme (2). We refer to [18] for the detailed analysis.

Owing to the high efficiency and the robustness of the HSS iteration method, many HSS variants were proposed in recent years to solve different kinds of systems of linear equations. For example, the preconditioned or the accelerated HSS iteration methods

were used to solve saddle-point problems [3,11,13,17,20,35,44], the modified HSS (MHSS) iteration method and its preconditioned versions were used to solve complex systems of linear equations or their equivalent real forms [8–10,26,28,29], the HSS iteration method and its parameterized version were used to solve singular problems [4,34]. As it is used as a solver, the HSS method and its variants were also used as preconditioners to accelerate the convergence of Krylov subspace methods [3,5,7,20–24,39].

Although lots of HSS-type iteration methods were published in recent years, most of them focused on the applications of these iteration methods to different kinds of problems, especially to the saddle-point problems. Only a few papers tried to propose some accelerated HSS-type methods by studying the HSS iteration scheme. For example, Bai et al. [16] used the successive-overrelaxation (SOR) technique to accelerate the convergence of the HSS iteration method and proposed an SOR accelerated HSS (abbreviated as SOR–HSS) iteration method of the form

$$\begin{cases} y^{(k+1)} &= (1-\omega)y^{(k)} + \omega(\alpha I + H(A))^{-1}((\alpha I - S(A))x^{(k)} + b), \\ x^{(k+1)} &= (1-\omega)x^{(k)} + \omega(\alpha I + S(A))^{-1}((\alpha I - H(A))y^{(k+1)} + b), \end{cases}$$

where α, ω are two given positive constants, and $x^{(0)}$, $y^{(0)}$ are two arbitrary initial guesses. In addition, since the Hermitian part $H(A)$ and the skew-Hermitian part $S(A)$ possess quite different properties, Yang et al. [43] used different shifting parameters for $H(A)$ and $S(A)$, respectively, in the iteration scheme (2) to accelerate the convergence of the HSS method. This leads to a generalized preconditioned HSS iteration method. Using the experimentally found optimal parameters, both of the two accelerated methods are more efficient than the original HSS iteration method. However, in the implementation of each of the two iteration methods, we should determine the values of two iteration parameters. This is a tough and time-consuming work. Other acceleration techniques for the HSS iteration method can be seen in [2,6,14,19,36] and references therein.

In this work, different from the ideas of all the existing HSS variants, we will introduce a non-stationary HSS iteration scheme by introducing two more control parameters for the HSS iteration scheme (3). Since the new control parameters involved in the new scheme are determined by minimizing the corresponding residual norms, we name the non-stationary HSS iteration scheme as the minimum residual HSS (abbreviated as MRHSS) iteration method. This acceleration idea comes from the non-stationary Uzawa method used for solving saddle-point problems [32]. Other similar and important acceleration techniques, such as the minimum residual smoothing, can be found in Refs. [30,31,40,41,45].

The remainder of this work is organized as follows. In Sect. 2 we define the MRHSS iteration method by introducing a control parameter for each step of the iteration scheme (3). In Sect. 3, the property of the parameters involved in the MRHSS iteration method is discussed. Moreover, the convergence property of the MRHSS iteration method is also studied. In Sect. 4, two examples are employed to verify the robustness and the efficiency of the MRHSS iteration method. Finally in Sect. 5, we give a brief conclusion for this paper.

2 The MRHSS iteration method

In this section, we will derive the MRHSS iteration method by modifying the HSS iteration scheme (3).

Denote $r^{(k)} = b - Ax^{(k)}$ and $r^{(k+\frac{1}{2})} = b - Ax^{(k+\frac{1}{2})}$, the HSS iteration scheme (3) can be rewritten as

$$x^{(k+\frac{1}{2})} = x^{(k)} + \delta^{(k)} \quad \text{and} \quad x^{(k+1)} = x^{(k+\frac{1}{2})} + \delta^{(k+\frac{1}{2})}, \tag{4}$$

where

$$\delta^{(k)} = (\alpha I + H(A))^{-1} r^{(k)}, \quad \delta^{(k+\frac{1}{2})} = (\alpha I + S(A))^{-1} r^{(k+\frac{1}{2})} \tag{5}$$

can be viewed as the search directions from $x^{(k)}$ to $x^{(k+\frac{1}{2})}$ and $x^{(k+\frac{1}{2})}$ to $x^{(k+1)}$, respectively. Thus, the step sizes are both unitary since the coefficients of $\delta^{(k)}$ and $\delta^{(k+\frac{1}{2})}$ in (4) are both 1. To further improve the efficiency of the iteration scheme (4), the following two interesting questions should be considered:

- whether the unitary step size is optimal?
- if it is not the optimal step size, can we find the optimal one?

To answer the two questions, we introduce two arbitrary positive parameters β_k and γ_k to control the step sizes, which leads to a new iteration scheme of the form

$$x^{(k+\frac{1}{2})} = x^{(k)} + \beta_k \delta^{(k)} \quad \text{and} \quad x^{(k+1)} = x^{(k+\frac{1}{2})} + \gamma_k \delta^{(k+\frac{1}{2})}. \tag{6}$$

Note that $A \in \mathbb{C}^{n\times n}$ and $x, b \in \mathbb{C}^n$, it should be better to choose β_k and γ_k in the complex field $\mathbb{C}$. Denote $M_1 := A(\alpha I + H(A))^{-1}$ and $M_2 := A(\alpha I + S(A))^{-1}$. The residual form of the iteration scheme (6) can be written as

$$r^{(k+\frac{1}{2})} = r^{(k)} - \beta_k M_1 r^{(k)} \quad \text{and} \quad r^{(k+1)} = r^{(k+\frac{1}{2})} - \gamma_k M_2 r^{(k+\frac{1}{2})}. \tag{7}$$

Let $\|\cdot\|$ be the 2-norm of a given vector. In the following, we determine the values of β_k and γ_k by minimizing the residual norms $\|r^{(k+\frac{1}{2})}\|$ and $\|r^{(k+1)}\|$, respectively. After simple calculation, the above two residual norms yield

$$\begin{aligned}\|r^{(k+\frac{1}{2})}\|^2 = \|r^{(k)}\|^2 &- 2\,Re(\beta_k)(H(M_1)r^{(k)},\, r^{(k)}) - 2\,Im(\beta_k)(iS(M_1)r^{(k)}, r^{(k)}) \\ &+ (Re(\beta_k)^2 + Im(\beta_k)^2)\|M_1 r^{(k)}\|^2\end{aligned} \tag{8}$$

and

$$\begin{aligned}\|r^{(k+1)}\|^2 = \|r^{(k+\frac{1}{2})}\|^2 &- 2\,Re(\gamma_k)(H(M_2)r^{(k+\frac{1}{2})},\, r^{(k+\frac{1}{2})}) \\ &- 2\,Im(\gamma_k)(iS(M_2)r^{(k+\frac{1}{2})}, r^{(k+\frac{1}{2})}) \\ &+ (Re(\gamma_k)^2 + Im(\gamma_k)^2)\|M_2 r^{(k+\frac{1}{2})}\|^2,\end{aligned} \tag{9}$$

where $Re(\cdot)$ and $Im(\cdot)$ denote the real and the imaginary parts of a complex number, respectively. Then, we can easily derive the minimum point for each of the functions $\|r^{(k+\frac{1}{2})}\|^2$ and $\|r^{(k+1)}\|^2$, i.e.,

$$Re(\beta_k) = \frac{(H(M_1)r^{(k)}, r^{(k)})}{\|M_1 r^{(k)}\|^2}, \quad Im(\beta_k) = \frac{(iS(M_1)r^{(k)}, r^{(k)})}{\|M_1 r^{(k)}\|^2} \tag{10}$$

and

$$Re(\gamma_k) = \frac{(H(M_2)r^{(k+\frac{1}{2})}, r^{(k+\frac{1}{2})})}{\|M_2 r^{(k+\frac{1}{2})}\|^2}, \quad Im(\gamma_k) = \frac{(iS(M_2)r^{(k+\frac{1}{2})}, r^{(k+\frac{1}{2})}))}{\|M_2 r^{(k+\frac{1}{2})}\|^2}. \tag{11}$$

It is difficult to compute the parameters β_k and γ_k according to the formulas (10) and (11). This is because the vectors $H(M_1)r^{(k)}$, $S(M_1)r^{(k)}$, $H(M_2)r^{(k+\frac{1}{2})}$ and $S(M_2)r^{(k+\frac{1}{2})}$ are hardly calculated for given vectors $r^{(k)}$ and $r^{(k+\frac{1}{2})}$. An easy way for calculating the parameter β_k can be derived as follows:

$$\begin{aligned} \beta_k &= Re(\beta_k) + i\, Im(\beta_k) = \frac{(H(M_1)r^{(k)}, r^{(k)})}{\|M_1 r^{(k)}\|^2} + i\frac{(i\, S(M_1)r^{(k)}, r^{(k)})}{\|M_1 r^{(k)}\|^2} \\ &= \frac{(r^{(k)}, H(M_1)r^{(k)})}{\|M_1 r^{(k)}\|^2} + \frac{(r^{(k)}, S(M_1)r^{(k)})}{\|M_1 r^{(k)}\|^2} = \frac{(r^{(k)}, M_1 r^{(k)})}{\|M_1 r^{(k)}\|^2}. \end{aligned} \tag{12}$$

Similarly, the parameter γ_k can be simplified as

$$\gamma_k = Re(\gamma_k) + i\, Im(\gamma_k) = \frac{(r^{(k+\frac{1}{2})}, M_2 r^{(k+\frac{1}{2})})}{\|M_2 r^{(k+\frac{1}{2})}\|^2}. \tag{13}$$

Using (5) and noticing that $M_1 = A(\alpha I + H(A))^{-1}$, $M_2 = A(\alpha I + S(A))^{-1}$, the parameters β_k and γ_k read

$$\beta_k = \frac{(r^{(k)}, A\delta^{(k)})}{\|A\delta^{(k)}\|^2}, \quad \gamma_k = \frac{(r^{(k+\frac{1}{2})}, A\delta^{(k+\frac{1}{2})})}{\|A\delta^{(k+\frac{1}{2})}\|^2}. \tag{14}$$

Therefore, the MRHSS iteration method can be defined as follows:

Method 1 (The MRHSS iteration method) *Given an initial guess* $x^{(0)} \in \mathbb{C}^n$, *for* $k = 0, 1, 2 \ldots$, *until* $\{x^{(k)}\}$ *converges*

(i) *compute* $x^{(k+1/2)}$ *from iteration scheme* $x^{(k+\frac{1}{2})} = x^{(k)} + \beta_k \delta^{(k)}$, *where*

$$\beta_k = \frac{(r^{(k)}, A\delta^{(k)})}{\|A\delta^{(k)}\|^2}, \quad \delta^{(k)} = (\alpha I + H(A))^{-1} r^{(k)};$$

(ii) *compute* $x^{(k+1)}$ *from iteration scheme* $x^{(k+1)} = x^{(k+\frac{1}{2})} + \gamma_k \delta^{(k+\frac{1}{2})}$, *where*

$$\gamma_k = \frac{(r^{(k+\frac{1}{2})}, A\delta^{(k+\frac{1}{2})})}{\|A\delta^{(k+\frac{1}{2})}\|^2}, \quad \delta^{(k+\frac{1}{2})} = (\alpha I + S(A))^{-1} r^{(k+\frac{1}{2})}.$$

Remark 1 The MRHSS iteration method is reduced to the classical HSS iteration method if we choose $\beta_k = \gamma_k = 1$.

Remark 2 The HSS iteration scheme (2) can be equivalently rewritten as the following one-step form

$$x^{(k+1)} = x^{(k)} + \mathscr{G}(\alpha)^{-1}(b - Ax^{(k)}),$$

where

$$\mathscr{G}(\alpha) = \frac{1}{2\alpha}(\alpha I + H(A))(\alpha I + S(A)) \tag{15}$$

can be used as a preconditioner to accelerate the convergence of some iteration methods [21,22]. This preconditioner is named as HSS preconditioner. Using this preconditioner, the residual-form iteration scheme of the preconditioned Orthomin(2) (abbreviated as POM(2)) method [27] used for solving the system of linear equations (1) reads

$$\tilde{r}^{(k+1)} = \tilde{r}^{(k)} - a_k \mathscr{G}(\alpha)^{-1} A\tilde{r}^{(k)} + a_k b_{k-1} \mathscr{G}(\alpha)^{-1} A\tilde{p}^{(k-1)}, \tag{16}$$

where $\tilde{p}^{(0)} = \tilde{r}^{(0)} = \mathscr{G}(\alpha)^{-1} r^{(0)}$ and $\tilde{p}^{(k)} = \tilde{r}^{(k)} - b_{k-1}\tilde{p}^{(k-1)}$. The coefficients a_k and b_{k-1} force orthogonality between $\tilde{r}^{(k+1)}$ and $\text{span}\{\mathscr{G}(\alpha)^{-1}A\tilde{r}^{(k)}, \mathscr{G}(\alpha)^{-1}A\tilde{p}^{(k-1)}\}$. Obviously, the iteration scheme (16) is equivalent to

$$r^{(k+1)} = r^{(k)} - a_k A\mathscr{G}(\alpha)^{-1} r^{(k)} + a_k b_{k-1} A\mathscr{G}(\alpha)^{-1} p^{(k-1)}, \tag{17}$$

where $p^{(0)} = r^{(0)}$ and $p^{(k)} = r^{(k)} - b_{k-1} p^{(k-1)}$. From (7), the residuals of MRHSS iteration method yield

$$\begin{aligned} r^{(k+1)} &= (I - \gamma_k M_2)(I - \beta_k M_1) r^{(k)} \\ &= r^{(k)} - \beta_k M_1 r^{(k)} - \gamma_k M_2 r^{(k)} + \beta_k \gamma_k M_2 M_1 r^{(k)}, \end{aligned} \tag{18}$$

where the coefficients β_k and γ_k are chosen such that

$$(r^{(k+1/2)}, M_1 r^{(k)}) = 0, \quad (r^{(k+1)}, M_2 r^{(k+1/2)}) = 0.$$

From the iteration schemes (17) and (18) and the choices of the iteration parameters a_k, b_k, β_k and γ_k, we can see that the POM(2) and the MRHSS are two different iteration methods. Since they have similar iteration schemes, the computing efficiencies of the two iteration methods will be compared in Sect. 4.

Remark 3 From (6), the iteration sequence of the MRHSS iteration method yields

$$x^{(j+1)} = x^{(j)} + \Big(\beta_j\,(\alpha I + H(A))^{-1} + \gamma_j\,(\alpha I + S(A))^{-1} - \beta_j\gamma_j\,(\alpha I + S(A))^{-1}M_1\Big)\,r^{(j)}, \quad \text{for } j = 0, 1, \ldots. \tag{19}$$

Adding (19), for $j = 0, 1, \ldots, m-1$, gives

$$x^{(m)} = x^{(0)} + \sum_{j=0}^{m-1}\Big(\beta_j\,(\alpha I + H(A))^{-1} + \gamma_j\,(\alpha I + S(A))^{-1} - \beta_j\gamma_j\,(\alpha I + S(A))^{-1}M_1\Big)\,r^{(j)}. \tag{20}$$

Using (18) recursively for $k = j-1, \ldots, 1, 0$, we can derive that

$$r^{(j)} = \left(\prod_{i=j-1}^{0}(I - \gamma_i\,M_2)(I - \beta_i\,M_1)\right) r^{(0)}. \tag{21}$$

Substituting (21) into (20), the approximate solution $x^{(m)}$ yields

$$x^{(m)} = x^{(0)} + B_0 r^{(0)} + \sum_{j=1}^{m-1} B_j \left(\prod_{i=j-1}^{0}(I - \gamma_i\,M_2)(I - \beta_i\,M_1)\right) r^{(0)},$$

where

$$B_j = \beta_j\,(\alpha I + H(A))^{-1} + \gamma_j\,(\alpha I + S(A))^{-1} - \beta_j\gamma_j\,(\alpha I + S(A))^{-1}M_1.$$

Therefore, it is easy to see that the approximate solution $x^{(m)}$ does not belong to the affine space $x_0 + \mathscr{K}_m$, where $\mathscr{K}_m$ is a Krylov subspace. This means the MRHSS iteration method is not a standard Krylov subspace method.

3 The properties of the MRHSS iteration method

In this section, we first discuss the properties of the parameters β_k and γ_k, and then analyze the convergence of the MRHSS iteration method.

From the derivations of the parameters $\beta_k, \gamma_k \in \mathbb{C}$, we know that $(Re(\beta_k), Im(\beta_k))$ and $(Re(\gamma_k), Im(\gamma_k))$ are the minimum points of the residual norms $\|r^{(k+\frac{1}{2})}\|$ and $\|r^{(k+1)}\|$, respectively. From (8) and (9), the residual norm $\|r^{(k+1)}\|$ can also be viewed as a real function of the real variables $Re(\beta_k)$, $Im(\beta_k)$, $Re(\gamma_k)$ and $Im(\gamma_k)$. Wether the quadruple $(Re(\beta_k), Im(\beta_k), Re(\gamma_k), Im(\gamma_k))$ defined by (10) and (11) is the minimum point of $\|r^{(k+1)}\|$ will be considered in the following.

To answer the above question, we first introduce a useful lemma.

Lemma 1 *Let $C = (c_{ij}) \in \mathbb{C}^{n\times n}$ be a Hermitian matrix. For any vector $z \in \mathbb{C}^n$, we have*

$$\frac{\partial(Cz, z)}{\partial(\|z\|^2)} = \frac{1}{\|z\|^2}(Cz, z).$$

Proof Let $Re(C)$ and $Re(z)$ be, respectively, the real parts of matrix $C \in \mathbb{C}^{n\times n}$ and vector $z \in \mathbb{C}^n$, i.e.,

$$Re(C) = \begin{pmatrix} Re(c_{11}) & Re(c_{12}) & \cdots & Re(c_{1n}) \\ Re(c_{21}) & Re(c_{22}) & \cdots & Re(c_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ Re(c_{n1}) & Re(c_{n2}) & \cdots & Re(c_{nn}) \end{pmatrix}, \quad Re(z) = \begin{pmatrix} Re(z_1) \\ Re(z_2) \\ \vdots \\ Re(z_n) \end{pmatrix}.$$

Similarly, the imaginary parts of matrix C and vector z are denoted by $Im(C)$ and $Im(z)$, respectively.

Since the matrix C is Hermitian, the inner product (Cz, z) can be viewed as a real function of the real variables $Re(z)$ and $Im(z)$, which has the form of

$$\begin{aligned} (Cz, z) &= (Re(z) + iIm(z))^* (Re(C) + iIm(C)) (Re(z) + iIm(z)) \\ &= Re(z)^T Re(C)Re(z) - Re(z)^T Im(C)Im(z) + Im(z)^T Re(C)Im(z) \\ &\quad + Im(z)^T Im(C)Re(z) \\ &\quad + i\left(Re(z)^T Re(C)Im(z) + Re(z)^T Im(C)Re(z)\right. \\ &\quad \left. - Im(z)^T Re(C)Re(z) + Im(z)^T Im(C)Im(z)\right). \end{aligned} \tag{22}$$

Inasmuch as the real and the imaginary parts of the matrix C are symmetric and skew-symmetric, respectively, we can derive that

$$\begin{aligned} &Re(z)^T Im(C)Re(z) = Im(z)^T Im(C)Im(z) = 0, \\ &Re(z)^T Im(C)Im(z) = (Re(z)^T Im(C)Im(z))^T = -Im(z)^T Im(C)Re(z), \\ &Re(z)^T Re(C)Im(z) = (Re(z)^T Re(C)Im(z))^T = Im(z)^T Re(C)Re(z). \end{aligned} \tag{23}$$

Substituting (23) into (22), the inner product (Cz, z) can be simplified as

$$(Cz, z) = Re(z)^T Re(C)Re(z) - 2Re(z)^T Im(C)Im(z) + Im(z)^T Re(C)Im(z). \tag{24}$$

Hence, the two first-order partial derivatives yield

$$\begin{aligned} \frac{\partial(Cz, z)}{\partial Re(z)} &= 2(Re(C)Re(z) - Im(C)Im(z)), \\ \frac{\partial(Cz, z)}{\partial Im(z)} &= 2(Re(C)Im(z) + Im(C)Re(z)). \end{aligned} \tag{25}$$

Let $\rho = \|z\|^2$, then the vector z can be written as

$$z = (\rho^{\frac{1}{2}} \cos\theta_1 e^{i\varphi_1}, \rho^{\frac{1}{2}} \cos\theta_2 e^{i\varphi_2}, \ldots, \rho^{\frac{1}{2}} \cos\theta_n e^{i\varphi_n})^T,$$

where θ_k, with $k = 1, 2, \ldots, n$, satisfy $\sum_{k=1}^{n} \cos^2\theta_k = 1$. Simple calculation gives

$$\frac{\partial Re(z)}{\partial \rho} = \frac{1}{2\rho} Re(z), \quad \frac{\partial Im(z)}{\partial \rho} = \frac{1}{2\rho} Im(z). \tag{26}$$

Using (24), (25) and (26), we can finally derive that

$$\frac{\partial (Cz, z)}{\partial \rho} = \left(\frac{\partial (Cz, z)}{\partial Re(z)}, \frac{\partial Re(z)}{\partial \rho}\right) + \left(\frac{\partial (Cz, z)}{\partial Im(z)}, \frac{\partial Im(z)}{\partial \rho}\right) = \frac{1}{\rho}(Cz, z). \tag{27}$$

The proof of this lemma is completed. □

In the following, we answer the question proposed before Lemma 1.

Theorem 1 *The quadruple* $(Re(\beta_k), Im(\beta_k), Re(\gamma_k), Im(\gamma_k))$ *defined by* (10) *and* (11) *is the minimum point of* $\|r^{(k+1)}\|$, *which means the values of* β_k *and* γ_k *defined by* (14) *are optimal in the complex field* $\mathbb{C}$.

Proof From equalities (7) and (9), the residual norm $\|r^{(k+1)}\|^2$ can be viewed as a real function of the complex parameters β_k and γ_k, or equivalently, of the real parameters $Re(\beta_k)$, $Im(\beta_k)$, $Re(\gamma_k)$ and $Im(\gamma_k)$. Let $\hat{r}(\tau_1, \tau_2) = r^{(k)} - (\tau_1 + i\tau_2) M_1 r^{(k)}$ and denote for convenience that

$$\begin{aligned} \phi(\hat{r}(\tau_1, \tau_2), \omega_1, \omega_2) := \|\hat{r}\|^2 &- 2\,\omega_1 (H(M_2)\hat{r},\ \hat{r}) \\ &- 2\,\omega_2 (i\ S(M_2)\hat{r}, \hat{r}) + (\omega_1^2 + \omega_2^2)\,\|M_2\hat{r}\|^2. \end{aligned} \tag{28}$$

Then, we have

$$\phi\left(\hat{r}(Re(\beta_k), Im(\beta_k)), Re(\gamma_k), Im(\gamma_k)\right) = \|r^{(k+1)}\|^2.$$

Note that $\|M_2\hat{r}\|^2 = (M_2^* M_2 \hat{r}, \hat{r})$, using (28) and Lemma 1 gives

$$\begin{aligned} \frac{\partial \phi}{\partial (\|\hat{r}\|^2)} = \frac{\|\hat{r}\|^2}{\|\hat{r}\|^2} &- \frac{2\,\omega_1}{\|\hat{r}\|^2}(H(M_2)\hat{r},\ \hat{r}) \\ &- \frac{2\,\omega_2}{\|\hat{r}\|^2}(i\,S(M_2)\hat{r}, \hat{r}) + \frac{\omega_1^2 + \omega_2^2}{\|\hat{r}\|^2}\|M_2\hat{r}\|^2 = \frac{1}{\|\hat{r}\|^2}\phi. \end{aligned}$$

Denote

$$\hat{\phi}(\tau_1, \tau_2, \omega_1, \omega_2) := \phi(\hat{r}(\tau_1, \tau_2), \omega_1, \omega_2). \tag{29}$$

The first-order partial derivatives of $\hat{\phi}(\tau_1, \tau_2, \omega_1, \omega_2)$ yield

$$\begin{aligned}
\frac{\partial\hat{\phi}}{\partial\tau_1} &= \frac{\partial\phi}{\partial(\|\hat{r}\|^2)}\frac{\partial(\|\hat{r}\|^2)}{\partial\tau_1} = \frac{2\phi}{\|\hat{r}\|^2}\left(\tau_1\|M_1 r^{(k)}\|^2 - (H(M_1)\, r^{(k)},\, r^{(k)})\right),\\
\frac{\partial\hat{\phi}}{\partial\tau_2} &= \frac{\partial\phi}{\partial(\|\hat{r}\|^2)}\frac{\partial(\|\hat{r}\|^2)}{\partial\tau_2} = \frac{2\phi}{\|\hat{r}\|^2}\left(\tau_2\|M_1 r^{(k)}\|^2 - (i\, S(M_1)\, r^{(k)},\, r^{(k)})\right),\\
\frac{\partial\hat{\phi}}{\partial\omega_1} &= 2\left(\omega_1\|M_2\hat{r}\|^2 - (H(M_2)\hat{r},\, \hat{r})\right),\\
\frac{\partial\hat{\phi}}{\partial\omega_2} &= 2\left(\omega_2\|M_2\hat{r}\|^2 - (i\, S(M_2)\hat{r},\, \hat{r})\right).
\end{aligned}$$

It is easy to see that the quadruple $(Re(\beta_k), Im(\beta_k), Re(\gamma_k), Im(\gamma_k))$ defined by (10) and (11) is the unique stationary point of the function $\hat{\phi}$. Furthermore, to verify wether the quadruple is the minimum point of the function $\hat{\phi}$, we denote

$$\begin{aligned}
\Phi_1(\tau_1) &:= \tau_1\|M_1 r^{(k)}\|^2 - (H(M_1)\, r^{(k)},\, r^{(k)}),\\
\Phi_2(\tau_2) &:= \tau_2\|M_1 r^{(k)}\|^2 - (i\, S(M_1)\, r^{(k)},\, r^{(k)}),\\
\Phi_3(\tau_1, \tau_2, \omega_1) &:= \omega_1\|M_2\hat{r}\|^2 - (H(M_2)\hat{r},\, \hat{r}),\\
\Phi_4(\tau_1, \tau_2, \omega_2) &:= \omega_2\|M_2\hat{r}\|^2 - (i\, S(M_2)\hat{r},\, \hat{r}).
\end{aligned}$$

Then, the second-order partial derivatives of $\hat{\phi}(\tau_1, \tau_2, \omega_1, \omega_2)$ yield

$$\begin{aligned}
\frac{\partial^2\hat{\phi}}{\partial\tau_1^2} &= \Phi_1(\tau_1)\frac{\partial}{\partial\tau_1}\left(\frac{2\phi}{\|\hat{r}\|^2}\right) + \frac{2\phi}{\|\hat{r}\|^2}\|M_1 r^{(k)}\|^2,\\
\frac{\partial^2\hat{\phi}}{\partial\tau_1\partial\tau_2} &= \Phi_1(\tau_1)\frac{\partial}{\partial\tau_2}\left(\frac{2\phi}{\|\hat{r}\|^2}\right),\\
\frac{\partial^2\hat{\phi}}{\partial\tau_2^2} &= \Phi_2(\tau_2)\frac{\partial}{\partial\tau_2}\left(\frac{2\phi}{\|\hat{r}\|^2}\right) + \frac{2\phi}{\|\hat{r}\|^2}\|M_1 r^{(k)}\|^2,\\
\frac{\partial^2\hat{\phi}}{\partial\tau_2\partial\tau_1} &= \Phi_2(\tau_2)\frac{\partial}{\partial\tau_1}\left(\frac{2\phi}{\|\hat{r}\|^2}\right),\\
\frac{\partial^2\hat{\phi}}{\partial\tau_1\partial\omega_1} &= \Phi_1(\tau_1)\frac{\partial}{\partial\omega_1}\left(\frac{2\phi}{\|\hat{r}\|^2}\right), \quad \frac{\partial^2\hat{\phi}}{\partial\tau_1\partial\omega_2} = \Phi_1(\tau_1)\frac{\partial}{\partial\omega_2}\left(\frac{2\phi}{\|\hat{r}\|^2}\right),\\
\frac{\partial^2\hat{\phi}}{\partial\tau_2\partial\omega_1} &= \Phi_2(\tau_2)\frac{\partial}{\partial\omega_1}\left(\frac{2\phi}{\|\hat{r}\|^2}\right), \quad \frac{\partial^2\hat{\phi}}{\partial\tau_2\partial\omega_2} = \Phi_2(\tau_2)\frac{\partial}{\partial\omega_2}\left(\frac{2\phi}{\|\hat{r}\|^2}\right),\\
\frac{\partial^2\hat{\phi}}{\partial\omega_1\partial\tau_1} &= \frac{2\Phi_3(\tau_1, \tau_2, \omega_1)}{\|\hat{r}\|^2}\frac{\partial\|\hat{r}\|^2}{\partial\tau_1}, \quad \frac{\partial^2\hat{\phi}}{\partial\omega_1\partial\tau_2} = \frac{2\Phi_3(\tau_1, \tau_2, \omega_1)}{\|\hat{r}\|^2}\frac{\partial\|\hat{r}\|^2}{\partial\tau_2},\\
\frac{\partial^2\hat{\phi}}{\partial\omega_2\partial\tau_1} &= \frac{2\Phi_4(\tau_1, \tau_2, \omega_2)}{\|\hat{r}\|^2}\frac{\partial\|\hat{r}\|^2}{\partial\tau_1}, \quad \frac{\partial^2\hat{\phi}}{\partial\omega_2\partial\tau_2} = \frac{2\Phi_4(\tau_1, \tau_2, \omega_2)}{\|\hat{r}\|^2}\frac{\partial\|\hat{r}\|^2}{\partial\tau_2},\\
\frac{\partial^2\hat{\phi}}{\partial\omega_1^2} &= \frac{\partial^2\hat{\phi}}{\partial\omega_2^2} = 2\|M_2\hat{r}\|^2, \quad \frac{\partial^2\hat{\phi}}{\partial\omega_1\partial\omega_2} = \frac{\partial^2\hat{\phi}}{\partial\omega_2\partial\omega_1} = 0.
\end{aligned}$$

Note that $\Phi_1(Re(\beta_k)) = \Phi_2(Im(\beta_k)) = 0$ and

$$\Phi_3(Re(\beta_k), Im(\beta_k), Re(\gamma_k)) = \Phi_4(Re(\beta_k), Im(\beta_k), Im(\gamma_k)) = 0,$$

the Hessian matrix of $\hat{\phi}$ at this stationary point $(Re(\beta_k), Im(\beta_k), Re(\gamma_k), Im(\gamma_k))$ has the form of

$$\begin{pmatrix} \frac{2\|r^{(k+1)}\|^2\|M_1 r^{(k)}\|^2}{\|r^{(k+1/2)}\|^2} & 0 & 0 & 0 \\ 0 & \frac{2\|r^{(k+1)}\|^2\|M_1 r^{(k)}\|^2}{\|r^{(k+1/2)}\|^2} & 0 & 0 \\ 0 & 0 & 2\|M_2 r^{(k+\frac{1}{2})}\|^2 & 0 \\ 0 & 0 & 0 & 2\|M_2 r^{(k+\frac{1}{2})}\|^2 \end{pmatrix}.$$

It is obvious that the Hessian matrix is Hermitian positive definite, which means the stationary point referred above is the unique minimum point of the function $\hat{\phi}$.

Therefore, the quadruple $(Re(\beta_k), Im(\beta_k), Re(\gamma_k), Im(\gamma_k))$ defined by (10) and (11) is the minimum point of $\|r^{(k+1)}\|$. □

Denote by $\mathscr{F}(B)$ the field of values of a matrix B, i.e., $\mathscr{F}(B)$ is the set of all complex numbers of the form $y^*By/(y^*y)$, where y is any complex vector other than the zero vector. Then, we can derive the following convergence property of the MRHSS iteration method.

Theorem 2 *The MRHSS iteration method used for solving the system of linear equations* (1) *is convergent for any initial vector* $x^{(0)} \in \mathbb{C}^n$ *if and only if*

$$0 \notin \mathscr{F}(A(\alpha I + H(A))^{-1}) \cap \mathscr{F}(A(\alpha I + S(A))^{-1}). \tag{30}$$

In this case, the residuals satisfy

$$\|r^{(k+1)}\| \leq \frac{\sqrt{\|A(\alpha I + H(A))^{-1}\|^2 - d_1^2}}{\|A(\alpha I + H(A))^{-1}\|} \frac{\sqrt{\|A(\alpha I + S(A))^{-1}\|^2 - d_2^2}}{\|A(\alpha I + S(A))^{-1}\|} \cdot \|r^{(k)}\| \tag{31}$$

for all k, where d_1 *and* d_2 *are the distances from the origin to* $\mathscr{F}(A(\alpha I + H(A))^{-1})$ *and* $\mathscr{F}(A(\alpha I + S(A))^{-1})$*, respectively.*

Proof From the derivation of the MRHSS iteration method, the parameters β_k and γ_k defined by (10) and (11) are obtained by minimizing the residual norms $\|r^{(k+1/2)}\|$ and $\|r^{(k+1)}\|$, respectively. Thus, from (7), we know that $r^{(k+1/2)}$ is equal to $r^{(k)}$ minus its projection onto $A(\alpha I + H(A))^{-1}r^{(k)}$, and $r^{(k+1)}$ is equal to $r^{(k+1/2)}$ minus its projection onto $A(\alpha I + S(A))^{-1}r^{(k+1/2)}$. Hence, $\|r^{(k+1/2)}\| \leq \|r^{(k)}\|$ with equality if and only if $r^{(k)}$ is already orthogonal to $A(\alpha I + H(A))^{-1}r^{(k)}$, i.e.,

$$(r^{(k)}, A(\alpha I + H(A))^{-1}r^{(k)}) = 0.$$

This means that $\|r^{(k+1/2)}\| < \|r^{(k)}\|$ if and only if $0 \notin \mathscr{F}((\alpha I + H(A))^{-*}A^*)$. Similarly, $\|r^{(k+1)}\| < \|r^{(k+1/2)}\|$ if and only if $0 \notin \mathscr{F}((\alpha I + S(A))^{-*}A^*)$. Therefore, we can conclude that $\|r^{(k+1)}\| < \|r^{(k)}\|$ if and only if

$$0 \notin \mathscr{F}((\alpha I + H(A))^{-*}A^*) \cap \mathscr{F}((\alpha I + S(A))^{-*}A^*). \tag{32}$$

Since, for any matrix $B \in \mathbb{C}^{n\times n}$, the field of values of B^* is just the complex conjugate of the field of values of B, the condition (32) can be replaced by

$$0 \notin \mathscr{F}(A(\alpha I + H(A))^{-1}) \cap \mathscr{F}(A(\alpha I + S(A))^{-1}).$$

Note that the field of values is a closed set, the distances of $\mathscr{F}(A(\alpha I + H(A))^{-1})$ and $\mathscr{F}(A(\alpha I + S(A))^{-1})$ from the origin can be, respectively, defined as

$$d_1 = \min_{0\neq y\in\mathbb{C}} \left| \frac{y^*A(\alpha I + H(A))^{-1}y}{y^*y} - 0 \right|, \quad d_2 = \min_{0\neq y\in\mathbb{C}} \left| \frac{y^*A(\alpha I + S(A))^{-1}y}{y^*y} - 0 \right|.$$

Substituting (10) and (11), respectively, into (8) and (9), we have

$$\begin{aligned}
\|r^{(k+1/2)}\|^2 &= \|r^{(k)}\|^2 - \frac{|(A(\alpha I + H(A))^{-1}r^{(k)}, r^{(k)})|^2}{\|A(\alpha I + H(A))^{-1}r^{(k)}\|^2} \\
&= \|r^{(k)}\|^2 \left(1 - \left|\frac{(A(\alpha I + H(A))^{-1}r^{(k)}, r^{(k)})}{(r^{(k)}, r^{(k)})}\right|^2 \right. \\
&\qquad \left. \frac{\|r^{(k)}\|^2}{\|A(\alpha I + H(A))^{-1}r^{(k)}\|^2}\right) \\
&\leq \|r^{(k)}\|^2 \left(1 - \frac{d_1^2}{\|A(\alpha I + H(A))^{-1}\|^2}\right)
\end{aligned}$$

and, similarly,

$$\|r^{(k+1)}\|^2 \leq \|r^{(k+1/2)}\|^2 \left(1 - \frac{d_2^2}{\|A(\alpha I + S(A))^{-1}\|^2}\right).$$

Therefore, the inequality (31) follows by combining the above two inequalities. □

4 Numerical examples

In this section, we use two examples to test the efficiency of the MRHSS iteration method. The numerical results of the MRHSS iteration method including numbers of iteration steps (denoted as IT) and elapsed CPU times (in seconds, denoted as CPU)

are compared with those of the HSS, the POM(2), the preconditioned generalized minimum residual (PGMRES) and the SOR–HSS iteration methods. The preconditioners used for the POM(2) and the PGMRES methods are both the HSS preconditioner (15).

The parameter α involved in the four methods, i.e., HSS, POM(2), PGMRES and MRHSS, and the parameters α, ω involved in the SOR–HSS iteration method are chosen as the experimentally found optimal ones, which lead to the least number of iteration steps. If the optimal iteration parameters form an interval, then we choose the optimal parameter as the one that belongs to this interval and leads to the least CPU time. The optimal iteration parameters determined in such a manner are denoted as $\alpha_{\exp}$ and $\omega_{\exp}$.

As we know, the purpose for using the experimentally found optimal parameter is to test the optimal performance of the tested iteration methods. However, it is not realistic to use the experimentally found optimal parameter in practical applications. So, many researchers tried to discuss the computations of the optimal parameters of the HSS iteration method or the HSS preconditioner [3,12,15,25,33,42]. Denote by $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ the minimum and the maximum eigenvalues of a Hermitian positive definite matrix, respectively. Bai et al. [15] proposed a kind of quasioptimal parameter, i.e.,

$$\alpha^{\star} = \sqrt{\lambda_{\max}(H(A))\lambda_{\min}(H(A))}. \tag{33}$$

Let $\widetilde{\mathscr{G}}(\alpha) = (\alpha I + H(A))(\alpha I + S(A))$ and $\widetilde{\mathscr{Q}}(\alpha) = (\alpha I - H(A))(\alpha I - S(A))$, the iteration matrix of the HSS iteration scheme (2) can be written as $\widetilde{\mathscr{G}}(\alpha)^{-1}\widetilde{\mathscr{Q}}(\alpha)$. Thus, the HSS iteration method will converge fast if the iteration matrix approaches to zero, which could be possible when $\widetilde{\mathscr{Q}}(\alpha) \approx 0$. So, for the HSS method, Huang [33] defined an optimal parameter as

$$\alpha_{\text{Huang}} := \arg\min_{\alpha>0}\{\|\widetilde{\mathscr{Q}}(\alpha)\|_F\}. \tag{34}$$

Furthermore, a practical formula for computing the optimal parameter α_{Huang} is also proposed. In addition, by noticing that

$$\widetilde{\mathscr{Q}}(\alpha) = (\alpha I - H(A))(\alpha I - S(A)) = \alpha I \cdot \alpha I - \alpha(H(A) + S(A)) + H(A)S(A),$$

similar to the idea proposed by Ren and Cao [37], we can define a function of the form

$$\Phi(\alpha) = \alpha\|I\|_F \cdot \alpha\|I\|_F - \alpha(\|H(A)\|_F + \|S(A)\|_F) + \|H(A)\|_F\|S(A)\|_F.$$

Then, an easily calculated parameter α is obtained by minimizing the above function $\Phi(\alpha)$, i.e.,

$$\alpha_{\text{est}} = \frac{\|H(A)\|_F + \|S(A)\|_F}{2n}. \tag{35}$$

The computing efficiencies of four iteration methods, i.e., HSS, POM(2), PGMRES and MRHSS, will also be tested by using the parameter values defined by (33), (34) and (35), respectively.

In actual computations, the initial guesses $x^{(0)}$ and $y^{(0)}$ of the iteration methods are both chosen as $(1, 1, \ldots, 1)^T \in \mathscr{R}^n$. All iteration processes are terminated once the current residuals satisfy

$$\frac{\|b - Ax^{(k)}\|}{\|b - Ax^{(0)}\|} \leq 10^{-5}, \tag{36}$$

or the numbers of iteration steps are over $k_{\max} = 5000$. In addition, all the computations are implemented in MATLAB [version 9.1.0.441655 (R2016b)] in double precision on a personal computer with 3.10GHZ central processing unit [Intel(R) Core(TM) i7-4770S] and 8.00GB memory.

Example 1 Consider the following two-dimensional convection-diffusion equation

$$\begin{aligned} -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + a(x, y)\frac{\partial u}{\partial x} + b(x, y)\frac{\partial u}{\partial y} &= f(x, y), \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega, \end{aligned} \tag{37}$$

where $\Omega = [0, 1]^2$. The coefficient functions $a(x, y)$ and $b(x, y)$ are chosen as

$$\begin{aligned} &\text{Case I.} \quad a(x, y) = x\sin(x + y), \quad b(x, y) = y\cos(xy); \\ &\text{Case II.} \quad a(x, y) = 5y\exp(xy), \quad b(x, y) = 5x\exp(x + y). \end{aligned} \tag{38}$$

We discretize this problem by using the five-point central difference discretization with mesh size $h = 1/l$ and obtain a kind of systems of linear equations $Ax = b$, where $A \in \mathbb{C}^{n\times n}$, $x, b \in \mathbb{C}^n$ and $n = (l - 1)^2$. Then, all the five tested iteration methods, i.e., HSS, SOR–HSS, POM(2), PGMRES and MRHSS, are used to solve this kind of systems of linear equations. In Table 1, we list the experimentally found optimal parameters $\alpha_{\exp}$ of the HSS, POM(2), PGMRES and MRHSS iteration methods, and $\alpha_{\exp}\backslash\omega_{\exp}$ of the SOR–HSS iteration method for different coefficient functions and different mesh numbers l.

Using the experimentally found optimal parameters $\alpha_{\exp}$ and $\omega_{\exp}$, the numerical results including the numbers of iteration steps and the elapsed CPU times of the five tested iteration methods, for the two cases of the coefficient functions and the three different mesh numbers, i.e., $l = 40$, 80 and 160, are listed in Table 2.

From the numerical results listed in Table 2, we can see that all the five tested iteration methods with respective optimal parameters are always convergent for different coefficient functions (i.e., Cases I and II) and different mesh numbers l. The SOR–HSS and the POM(2) iteration methods perform much better than the HSS iteration method. However, their computing efficiencies are lower than the PGMRES method. Among the five tested iteration methods, the MRHSS is always the most efficient one. It costs the smallest numbers of iteration steps and computing times to converge to the solution. Moreover, the numbers of iteration steps costed by the MRHSS iteration method are almost h-independent.

Table 1 The experimentally found optimal parameter values of the tested methods for Example 1

	$l = 40$	$l = 80$	$l = 160$
Case I			
HSS	0.338	0.187	0.103
SOR–HSS	0.052\0.841	0.026\0.901	0.010\0.900
POM(2)	0.031	0.015	0.006
PGMRES	0.065	0.023	0.009
MRHSS	0.0008	0.0002	0.0001
Case II			
HSS	0.382	0.208	0.113
SOR–HSS	0.286\0.954	0.1440\0.952	0.073\0.951
POM(2)	0.288	0.132	0.057
PGMRES	0.652	0.420	0.192
MRHSS	0.047	0.009	0.003

Table 2 The numerical results of the tested methods with experimentally found optimal parameter values for Example 1

Method	$l = 40$		$l = 80$		$l = 160$	
	IT	CPU	IT	CPU	IT	CPU
Case I						
HSS	116	0.0780	219	2.3556	407	34.149
SOR–HSS	26	0.0312	38	0.4368	50	4.1496
POM(2)	22	0.0312	26	0.4056	36	2.9016
PGMRES	15	0.0312	18	0.2652	24	2.0280
MRHSS	3	0.0156	3	0.0156	3	0.3120
Case II						
HSS	91	0.0780	171	1.7784	314	26.177
SOR–HSS	74	0.0624	132	1.3416	225	18.673
POM(2)	77	0.0624	135	2.0436	222	18.549
PGMRES	38	0.0312	58	1.0140	88	8.0809
MRHSS	29	0.0156	26	0.6084	22	2.1840

In addition to using the experimentally found optimal parameters, we also adopt three kinds of easily calculated parameter values defined by (33), (34) and (35) to further test the efficiencies of the five iteration methods. These parameter values are listed in Table 3. Using these parameter values, the numerical results of the tested iteration methods are listed in Table 4. From the numerical results, we can see that, compared with the HSS and the POM(2) iteration methods, both the PGMRES and the MRHSS iteration methods are always convergent within $k_{\max}$ iteration steps for each choice of the three parameter values. Although the PGMRES method outperforms

Table 3 The values of parameter α computed by formulas (33), (34) and (35) for Example 1

	(33)	(34)	(35)
Case I			
$l = 80$	0.1551	8.5775e−6	0.0287
$l = 160$	0.0771	2.1409e−6	0.0142
Case II			
$l = 80$	0.1378	2.2865e−3	0.0293
$l = 160$	0.0685	5.7392e−4	0.0143

Table 4 The numerical results of iteration methods with different values of parameter α for Example 1

	Method	(33)		(34)		(35)	
		IT	CPU	IT	CPU	IT	CPU
Case I							
$l = 80$	HSS	260	2.8704	–	–	1131	11.872
	POM(2)	51	0.7956	–	–	34	0.4992
	PGMRES	29	0.4680	746	70.669	19	0.3432
	MRHSS	134	1.5763	3	0.0312	32	0.4992
$l = 160$	HSS	541	49.515	–	–	2373	206.61
	POM(2)	77	7.3008	–	–	49	4.7268
	PGMRES	39	3.9624	1528	740.27	25	2.2152
	MRHSS	244	24.383	3	0.2340	53	4.9764
Case II							
$l = 80$	HSS	235	2.8080	–	–	753	7.8313
	POM(2)	135	2.1684	–	–	–	–
	PGMRES	70	1.2948	1298	277.98	196	5.0544
	MRHSS	101	1.8408	39	0.4524	32	0.5304
$l = 160$	HSS	478	44.616	–	–	1547	137.94
	POM(2)	225	20.311	–	–	–	–
	PGMRES	99	9.7501	2726	3347.8	224	26.317
	MRHSS	177	16.521	33	2.7924	40	3.9000

the MRHSS iteration method for some cases, the MRHSS iteration method, taken together, is the most robust one.

In Fig. 1, we plot the numbers of iteration steps of the five tested iteration methods, i.e., HSS, HSS–SOR, POM(2), PGMRES and MRHSS, with respect to the iteration parameter α when $l = 80$. The other parameter ω of the SOR–HSS is chosen as $\omega = 0.901$ for the Case I and $\omega = 0.952$ for the Case II; see Table 1. The numerical curves show that the MRHSS iteration method always costs less iteration steps than the HSS and the SOR–HSS iteration methods for both of the two cases. Comparing with the POM(2) and the PGMRES methods, MRHSS method is more sensitive to the value of parameter α for Case I, i.e., its computing efficiency is largely dependent on the value of parameter α. For Case II, MRHSS method outperforms the POM(2) method.

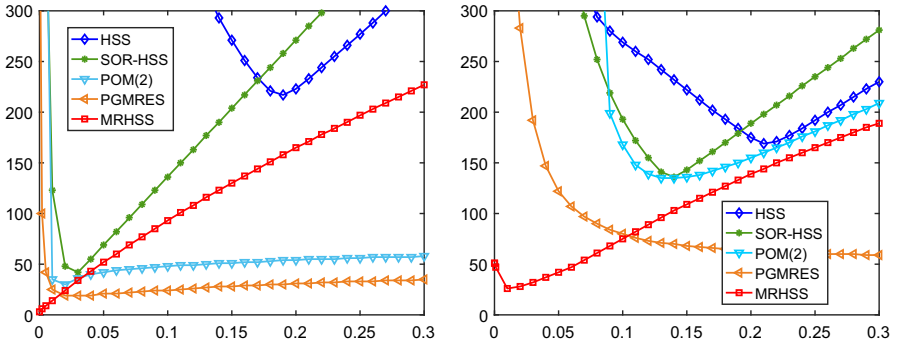

Fig. 1 Numbers of iteration steps versus iteration parameter α for the Cases I (left) and II (right) of Example 1

Table 5 The experimentally found optimal parameter values of the HSS-type methods for Example 2

	$l = 40$	$l = 80$	$l = 160$
HSS	8.2	10.6	15.1
SOR–HSS	8.2\1.1	12.0\1.2	16.4\1.2
POM(2)	5.1	8.3	9.8
PGMRES	5.6	8.7	11.1
MRHSS	0.21	0.24	0.31

However, it is still more sensitive to the values of parameter α than the PGMRES method. In addition, one striking feature of the MRHSS method, for both of the two cases, is that it achieves the minimum number of iteration steps when α approaches to zero.

Example 2 (See [1]) The complex non-Hermitian positive definite linear system

$$\left(I + \left(1 + \frac{1}{\sqrt{3}}i\right)\frac{\tau}{4}L\right)x = b \tag{39}$$

arises in Padé approximation type integration schemes for parabolic problems. Here L is the matrix of a standard five point discrete operator, approximating the negative Laplacian operator

$$Lu(x) = -\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2}, \quad x = (x_1, x_2) \in \Omega$$

with homogeneous Dirichlet boundary conditions on a uniform mesh in the unit square. Parameter $h = 1/l$ is the step-size, and l is the mesh number in both of the two directions. In the implementation, we choose $\tau = h$.

In Table 5, we list the experimentally found optimal values of parameters α and ω for the five tested iteration methods including HSS, SOR–HSS, POM(2), PGMRES and MRHSS with different mesh numbers l. Using these optimal values, the numerical

Table 6 The numerical results of iteration methods with experimentally found optimal parameter values for Example 2

Method	$l = 40$		$l = 80$		$l = 160$	
	IT	CPU	IT	CPU	IT	CPU
HSS	36	0.0468	50	0.7956	69	7.2540
SOR–HSS	31	0.0312	38	0.5616	50	5.1792
POM(2)	19	0.0156	24	0.4836	30	3.3384
PGMRES	20	0.0312	26	0.5304	33	4.0560
MRHSS	5	0.0000	5	0.1248	5	0.5772

Table 7 The numerical results of iteration methods with different values of parameter α for Example 2

Method	(33)		(34)		(35)	
	IT	CPU	IT	CPU	IT	CPU
$l = 80$						
HSS	58	0.6864	336	4.3836	571	7.6596
POM(2)	29	0.4524	101	1.7940	126	2.1216
PGMRES	27	0.4056	41	0.9984	50	1.1232
MRHSS	27	0.4028	98	1.7628	6	0.1560
$l = 160$						
HSS	81	8.0029	670	68.1724	1105	110.14
POM(2)	37	3.9624	190	19.1257	232	23.385
PGMRES	35	3.8688	58	6.7860	75	9.1261
MRHSS	37	3.9468	196	20.6545	6	0.6552

results of the five iteration methods are listed in Table 6. The numerical results show that, compared with the HSS, the SOR–HSS, the POM(2) and the PGMRES methods, the MRHSS iteration method always costs the smallest numbers of iteration steps and computing times to converge to the solution. Moreover, the numbers of iteration steps of the MRHSS iteration method are l- or h-independent for this example.

In addition to using the experimentally found optimal parameter values, the parameter values defined by (33), (34) and (35) are also used when we test the numerical efficiencies of the five iteration methods for this example. The values of parameter α computed by (33), (34) and (35) are 12.987, 75.574 and 0.8977 for $l = 80$, and 18.132, 151.15 and 0.8896 for $l = 160$, respectively. Using these parameter values, the numerical results of the tested iteration methods are listed in Table 7. We can see that the PGMRES and the MRHSS iteration methods always outperform the HSS and the POM(2) iteration methods no matter which parameter value is used. However, using different values of parameter α, the PGMRES and the MRHSS iteration methods have their own strengths.

In Fig. 2, we plot the numbers of iteration steps of the five tested iteration methods, i.e., HSS, HSS–SOR, POM(2), PGMRES and MRHSS, with respect to the values of iteration parameter α for the cases $l = 80$ and 160, respectively. The other parameter ω of the SOR–HSS is chosen as $\omega = 1.2$ for both of the two cases $l = 80$ and 160; see Table 5. Compared with the HSS and the SOR–HSS iteration methods, the

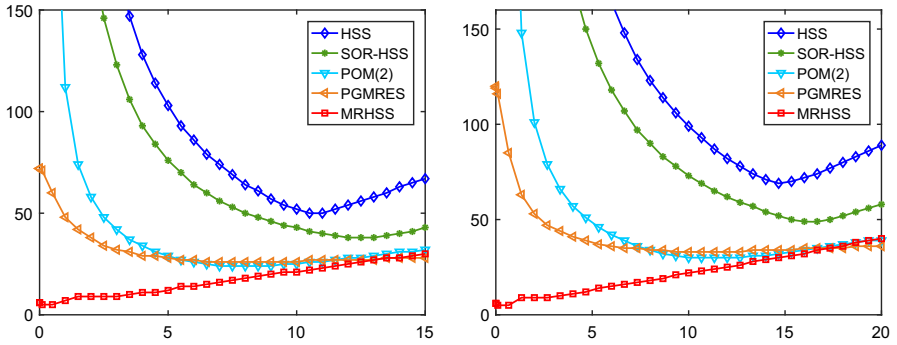

Fig. 2 Numbers of iteration steps versus iteration parameter α for the cases $l = 80$ (left) and $l = 160$ (right) of Example 2

POM(2) and the PGMRES iteration methods always cost smaller numbers of iteration steps to achieve the same stopping criterion. Among the five tested iteration methods, the MRHSS iteration method is the most efficient one. It outperforms the other four iteration methods for nearly all of the tested values of parameter α, and the number of iteration steps of the MRHSS iteration method is insensitive to the value of parameter α. In addition, similar to Example 1, the MRHSS iteration method also achieves its minimum number of iteration steps when α very closes to zero.

Therefore, from the numerical results of Examples 1 and 2, we can conclude that the MRHSS iteration method discussed in this work is very efficient and robust, especially if parameter α is very small.

5 Conclusions

For the non-Hermitian positive definite system of linear equations, we present a kind of minimum residual HSS (MRHSS) iteration method to approximate its solution. The convergence property of the MRHSS iteration method is carefully analyzed. Numerical results showed that the MRHSS iteration method is very efficient and robust for both of the two examples. However, we in this work only considered the "exact" version of the MRHSS iteration method. The investigation of the inexact variant of the MRHSS iteration method is left for future work. In addition, whether the MRHSS iteration method is still powerful for the non-Hermitian positive semidefinite systems of linear equations, including saddle-point problems, is also an interesting topic.

Acknowledgements The authors are very much indebted to the referees for their constructive and valuable comments and suggestions which greatly improved the quality of this paper. This work is partially supported by the National Key Research and Development Program of China (Grant No. 2018YFC0406600) and the National Natural Science Foundation of China (Grant Nos. 11471150, 11771225, 11401281).

References

1. Axelsson, O., Kucherov, A.: Real valued iterative methods for solving complex symmetric linear systems. Numer. Linear Algebra Appl. **7**(4), 197–218 (2000)

2. Bai, Z.-Z.: Several splittings for non-Hermitian linear systems. Sci. China A: Math. **51**(8), 1339–1348 (2008)
3. Bai, Z.-Z.: Optimal parameters in the HSS-like methods for saddle-point problems. Numer. Linear Algebra Appl. **16**(6), 447–479 (2009)
4. Bai, Z.-Z.: On semi-convergence of Hermitian and skew-Hermitian splitting methods for singular linear systems. Computing **89**, 171–197 (2010)
5. Bai, Z.-Z.: Motivations and realizations of Krylov subspace methods for large sparse linear systems. J. Comput. Appl. Math. **283**, 71–78 (2015)
6. Bai, Z.-Z.: Quasi-HSS iteration methods for non-Hermitian positive definite linear systems of strong skew-Hermitian parts. Numer. Linear Algebra Appl. **25**, e2116 (2018). https://doi.org/10.1002/nla.2116
7. Bai, Z.-Z., Benzi, M.: Regularized HSS iteration methods for saddle-point linear systems. BIT Numer. Math. **57**(2), 287–311 (2017)
8. Bai, Z.-Z., Benzi, M., Chen, F.: Modified HSS iteration methods for a class of complex symmetric linear systems. Computing **87**, 93–111 (2010)
9. Bai, Z.-Z., Benzi, M., Chen, F.: On preconditioned MHSS iteration methods for complex symmetric linear systems. Numer. Algorithms **56**, 297–317 (2011)
10. Bai, Z.-Z., Benzi, M., Chen, F., Wang, Z.-Q.: 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**, 343–369 (2013)
11. Bai, Z.-Z., Golub, G.H.: Accelerated Hermitian and skew-Hermitian splitting iteration methods for saddle-point problems. IMA J. Numer. Anal. **27**(1), 1–23 (2007)
12. 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. **2**(28), 583–603 (2006)
13. 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**(257), 287–298 (2007)
14. Bai, Z.-Z., Golub, G.H., Lu, L.-Z., Yin, J.-F.: Block triangular and skew-Hermitian splitting methods for positive-definite linear systems. SIAM J. Sci. Comput. **26**, 844–863 (2005)
15. 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)
16. Bai, Z.-Z., Golub, G.H., Ng, M.K.: On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations. Numer. Linear Algebra Appl. **14**(4), 319–335 (2007)
17. 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)
18. Bai, Z.-Z., Rozložník, M.: On the numerical behavior of matrix splitting iteration methods for solving linear systems. SIAM J. Numer. Anal. **53**(4), 1716–1737 (2015)
19. Benzi, M.: A generalization of the Hermitian and skew-Hermitian splitting iteration. SIAM J. Matrix Anal. Appl. **31**(2), 360–374 (2009)
20. Benzi, M., Gander, M., Golub, G.H.: Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems. BIT Numer. Math. **43**, 881–900 (2003)
21. Benzi, M., Golub, G.H.: A preconditioner for generalized saddle point problems. SIAM J. Matrix Anal. Appl. **26**, 20–41 (2005)
22. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
23. Cao, Y., Ren, Z.-R., Shi, Q.: A simplified HSS preconditioner for generalized saddle point problems. BIT Numer. Math. **56**(2), 423–439 (2016)
24. Cao, Y., Yao, L.-Q., Jiang, M.-Q., Niu, Q.: A relaxed HSS preconditioner for saddle point problems from meshfree discretization. J. Comput. Math. **21**, 398–421 (2013)
25. Chen, F.: On choices of iteration parameter in HSS method. Appl. Math. Comput. **271**, 832–837 (2015)
26. Chen, F., Liu, Q.-Q.: On semi-convergence of modified HSS iteration methods. Numer. Algorithms **64**(3), 507–518 (2013)
27. Greenbaum, A.: Iterative Methods for Solving Linear Systems. SIAM, Philadelphia (1997)
28. Gu, G.-D.: HSS method with a complex parameter for the solution of complex linear system. J. Comput. Math. **29**(4), 441–457 (2011)
29. Guo, X.-X., Wang, S.: Modified HSS iteration methods for a class of non-Hermitian positive-definite linear systems. Appl. Math. Comput. **218**(20), 10122–10128 (2012)

30. Gutknecht, M.H., Rozložník, M.: By how much can residual minimization accelerate the convergence of orthogonal residual methods? Numer. Algorithms **27**(2), 189–213 (2001)
31. Gutknecht, M.H., Rozložník, M.: Residual smoothing techniques: do they improve the limiting accuracy of iterative solvers? BIT Numer. Math. **41**(1), 86–114 (2001)
32. Hu, Q.-Y., Zou, J.: An iterative method with variable relaxation parameters for saddle-point problems. SIAM J. Matrix Anal. Appl. **23**(2), 317–338 (2001)
33. Huang, Y.-M.: A practical formula for computing optimal parameters in the HSS iteration methods. J. Comput. Appl. Math. **255**, 142–149 (2014)
34. Li, W., Liu, Y.-P., Peng, X.-F.: The generalized HSS method for solving singular linear systems. J. Comput. Appl. Math. **236**(9), 2338–2353 (2012)
35. Li, X., Yang, A.-L., Wu, Y.-J.: Parameterized preconditioned Hermitian and skew-Hermitian splitting iteration method for saddle-point problems. Int. J. Comput. Math. **91**(6), 1224–1238 (2014)
36. Meng, G.-Y., Wen, R.-P., Zhao, Q.-S.: The generalized HSS method with a flexible shift-parameter for non-Hermitian positive definite linear systems. BIT Numer. Math. **56**(2), 543–556 (2016)
37. Ren, Z.-R., Cao, Y.: An alternating positive-semidefinite splitting preconditioner for saddle point problems from time-harmonic eddy current models. IMA J. Numer. Anal. **36**, 922–946 (2016)
38. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
39. Simoncini, V., Benzi, M.: Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems. SIAM J. Matrix Anal. Appl. **26**(2), 377–389 (2004)
40. Vecharynski, E., Knyazev, A.: Preconditioned steepest descent-like methods for symmetric indefinite systems. Linear Algebra Appl. **511**, 274–295 (2016)
41. Walker, H.F.: Residual smoothing and peak/plateau behavior in Krylov subspace methods. Appl. Numer. Math. **19**(3), 279–286 (1995)
42. Yang, A.-L.: Scaled norm minimization method for computing the parameters of the HSS and the two-parameter HSS preconditioners. Numer. Linear Algebra Appl. **25**, e2169 (2018). https://doi.org/10.1002/nla.2169
43. Yang, A.-L., An, J., Wu, Y.-J.: A generalized preconditioned HSS method for non-Hermitian positive definite linear systems. Appl. Math. Comput. **216**(6), 1715–1722 (2010)
44. Yang, A.-L., Wu, Y.-J.: The Uzawa-HSS method for saddle-point problems. Appl. Math. Lett. **38**, 38–42 (2014)
45. Zhou, L., Walker, H.F.: Residual smoothing techniques for iterative methods. SIAM J. Sci. Comput. **15**(2), 297–312 (1994)