Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(6): 13476–13497.
DOI: 10.3934/math.2025605
Received: 07 March 2025
Revised: 02 June 2025
Accepted: 04 June 2025
Published: 12 June 2025

Research article

The HSS splitting hierarchical identification algorithms for solving the Sylvester matrix equation

Huiling Wang[1,*], Zhaolu Tian[1] and Yufeng Nie[2]

[1] College of Applied Mathematics, Shanxi University of Finance and Economics, Taiyuan 030006, China

[2] School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710072, China

* **Correspondence:** Email: wanghuiling@sxufe.edu.cn.

Abstract: By combining the hierarchical identification principle with HSS splitting, we presented the HSS splitting hierarchical identification algorithm for solving the Sylvester matrix equation in this paper. To enhance the convergence rate of the algorithm, the momentum item was introduced in the iteration. We conducted an in-depth analysis of the sufficient conditions that ensured the convergence properties of the proposed algorithms. Additionally, the optimal parameters involved in the algorithms were computed exactly in each iteration by the minimum residual technique for specific cases. Thus, the adaptive forms of the corresponding algorithms were obtained. Finally, several numerical examples were implemented to demonstrate the superiority and effectiveness of the designed algorithms in this paper.

Keywords: hierarchical identification principle; Hermitian and skew-Hermitian splitting; the minimum residual technique; momentum; optimal parameters
Mathematics Subject Classification: 15A24, 65F30

1. Introduction

In this paper, we mainly consider the iteration solutions of the following Sylvester matrix equation:

$$AX + XB = C, \tag{1.1}$$

where $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $C \in \mathbb{C}^{m\times n}$ are constant matrices, and $X \in \mathbb{C}^{m\times n}$ is the unknown matrix to be solved. Eq (1.1) has wide applications in image processing [5], stability and analysis of linear systems [10], and power systems [9]. Extensive research has been conducted on solving of the equation, especially regarding iterative methods. For example, the Smith method [14], the alternating direction implicit method [7], the gradient-based algorithm [6, 13], the adaptive gradient-based momentum iteration (AGMI) algorithm [19], the Hermitian and skew-Hermitian splitting (HSS) iteration method [2], the IO iteration algorithm [16, 17], and the multiplicative splitting iterative method [22]. Additionally, numerous other iterative methods are available for solving Eq (1.1) and the associated matrix equations [12, 15].

The HSS iteration method [1] was first employed for solving the linear system $Ax = b$. It was later generalized to obtain the solution of Eq (1.1) in [2]. Subsequently, Wang et al. [18] proposed a positive definite and skew-Hermitian splitting iteration method for the matrix euqation. The preconditioned positive definite and skew-Hermitian splitting iteration algorithm was further presented in [23]. Zheng and Ma [24] advanced the normal and skew-Hermitian splitting iteration methods based on the new splitting of the matrices A and B. Additionally, Li et al. [11] introduced a preconditioned HSS iteration method along with its non-alternating variant for the equation.

Inspired by the ideas presented in [2,6] and combing the hierarchical identification principle with the HSS splitting of the matrices A and B, we develop a HSS splitting hierarchical identification (HSSHI) iteration algorithm for solving Eq (1.1). Owing to the advantages of the heavy-ball momentum method which is well-known for accelerating the convergence of the gradient method [4], the momentum term is incorporated into the HSSHI iteration process, resulting in the momentum-based HSS splitting hierarchical identification (MHSSHI) algorithm. For these algorithms, we analyze the 2-norm of the error matrices and derive the conditions that the parameters need to satisfy to ensure convergence. Additionally, selecting optimal parameters is vital for the effectiveness of the algorithms. Most literature gives only the quasi-optimal parameters. In this paper, we explicitly provide the optimal parameters through the minimum residual technique [21], when the preconditioning matrices are $H(A) = \frac{1}{2}(A + A^H)$ and $H(B) = \frac{1}{2}(B + B^H)$. Since the parameters change with each iteration, the adaptive forms of the corresponding algorithms are provided.

The remainder of this paper is organized as follows: In Section 2, we propose the HSSHI algorithm, give its convergence property in detail, and obtain the optimal parameters using the iterative information. In Section 3, we present the MHSSHI algorithm, analyze its convergence, and give the adaptive MHSSHI algorithm. In Section 4, several numerical examples are employed to exhibit the robustness and efficiencies of the proposed algorithms. Finally, some conclusions are drawn in the last section.

2. The HSSHI iteration algorithm

By utilizing the hierarchical identification principle [6], Eq (1.1) can be reformulated into two subsystems as follows:

$$AX = b_1, \quad XB = b_2,$$

where

$$b_1 := C - XB, \quad b_2 := C - AX. \tag{2.1}$$

Do the Hermitian and skew-Hermitian splitting on the matrices A and B

$$A = H(A) + S(A), \quad B = H(B) + S(B),$$

with

$$\begin{cases} H(A) = \frac{1}{2}(A + A^H), S(A) = \frac{1}{2}(A - A^H), \\ H(B) = \frac{1}{2}(B + B^H), S(B) = \frac{1}{2}(B - B^H). \end{cases}$$

We apply the non-alternating preconditioned HSS (NPHSS) iteration method [20] to solve each above-mentioned subsystem as follows:

$$(\alpha P + H(A))X_1^{(k+1)} = (\alpha P - S(A))X_1^{(k)} + b_1, \tag{2.2}$$

$$X_2^{(k+1)}(\beta Q + H(B)) = X_2^{(k)}(\beta Q - S(B)) + b_2, \tag{2.3}$$

where P and Q are given Hermitian positive definite matrices. Substituting (2.1) into two Eqs (2.2) and (2.3), we get

$$(\alpha P + H(A))X_1^{(k+1)} = (\alpha P - S(A))X_1^{(k)} + C - XB,$$

$$X_2^{(k+1)}(\beta Q + H(B)) = X_2^{(k)}(\beta Q - S(B)) + C - AX.$$

The unknown variable X is approximated by its estimated value at the k-th step. Hence, we have

$$(\alpha P + H(A))X_1^{(k+1)} = (\alpha P - S(A))X_1^{(k)} + C - X_1^{(k)}B,$$

$$X_2^{(k+1)}(\beta Q + H(B)) = X_2^{(k)}(\beta Q - S(B)) + C - AX_2^{(k)}.$$

Replacing $X_1^{(k)}$ and $X_2^{(k)}$ by the average $X^{(k)} = \frac{X_1^{(k)}+X_2^{(k)}}{2}$, we obtain

$$(\alpha P + H(A))X_1^{(k+1)} = (\alpha P - S(A))X^{(k)} + C - X^{(k)}B, \tag{2.4}$$

$$X_2^{(k+1)}(\beta Q + H(B)) = X^{(k)}(\beta Q - S(B)) + C - AX^{(k)}. \tag{2.5}$$

Further simplifying the above two Eqs (2.4) and (2.5) leads to the HSS splitting hierarchical identification iteration algorithm, which is presented as follows:

Algorithm 1 The HSSHI algorithm

Input: Given an initial solution $X^{(1)}$, the preconditioners P and Q, as well as the parameters α and β

Output: $X^{(k+1)}$

1: **For** $k = 1, 2, \cdots$, **until it converges, solve**

2: $(\alpha P + H(A))X_1^{(k+1)} = (\alpha P + H(A))X^{(k)} + C - AX^{(k)} - X^{(k)}B$,

3: $X_2^{(k+1)}(\beta Q + H(B)) = X^{(k)}(\beta Q + H(B)) + C - AX^{(k)} - X^{(k)}B$,

4: $X^{(k+1)} = \frac{X_1^{(k+1)}+X_2^{(k+1)}}{2}$.

5: **End**

Remark 1. *We consider the following three cases for the preconditioners P and Q:*

1. *$P = I_m$ and $Q = I_n$, where I_s is an identity matrix of size s.*
2. *$P = H(A)$ and $Q = H(B)$.*
3. *$P = tridiag(H(A))$ and $Q = tridiag(H(B))$, where $tridiag(H(A))$ and $tridiag(H(B))$ are the tridiagonal matrices of $H(A)$ and $H(B)$, respectively.*

2.1. Convergence analysis

In this section, we mainly investigate the convergence property of the HSSHI algorithm and give the corresponding convergence result.

Theroem 1. *Let* $\widetilde{H}(A) = P^{-\frac{1}{2}}H(A)P^{-\frac{1}{2}}$, $\widetilde{S}(A) = P^{-\frac{1}{2}}S(A)P^{-\frac{1}{2}}$, $\widetilde{H}(B) = Q^{-\frac{1}{2}}H(B)Q^{-\frac{1}{2}}$, *and* $\widetilde{S}(B) = Q^{-\frac{1}{2}}S(B)Q^{-\frac{1}{2}}$. *Assume* X^* *is the solution of Eq (1.1). The iterative solution* $X^{(k)}$ *generated by Algorithm 1 converges to* X^* *for any initial value if and only if the parameters* α *and* β *satisfy the condition*

$$\frac{\sqrt{\lambda_{\max}(P)\lambda_{\min}(P)(\alpha^2+\sigma_{\max}^2(\widetilde{S}(A)))}+\sigma_{\max}(B)}{\lambda_{\min}(P)(\alpha+\lambda_{\min}(\widetilde{H}(A)))} + \frac{\sqrt{\lambda_{\max}(Q)\lambda_{\min}(Q)(\beta^2+\sigma_{\max}^2(\widetilde{S}(B)))}+\sigma_{\max}(A)}{\lambda_{\min}(Q)(\beta+\lambda_{\min}(\widetilde{H}(B)))} < 2, \tag{2.6}$$

where $\sigma_{\max}(E)$, $\lambda_{\max}(E)$, *and* $\lambda_{\min}(E)$ *are the maximum singular value and the maximum and minimum eigenvalues of the matrix E, respectively.*

Proof: From Algorithm 1, it is easy to obtain

$$X_1^{(k+1)} = X^{(k)} + (\alpha P + H(A))^{-1}[C - AX^{(k)} - X^{(k)}B],$$

$$X_2^{(k+1)} = X^{(k)} + [C - AX^{(k)} - X^{(k)}B](\beta Q + H(B))^{-1}.$$

The $(k+1)$-th iteration can be rewritten as

$$\begin{aligned} X^{(k+1)} =& X^{(k)} + \frac{1}{2}(\alpha P + H(A))^{-1}[C - AX^{(k)} - X^{(k)}B] \\ &+ \frac{1}{2}[C - AX^{(k)} - X^{(k)}B](\beta Q + H(B))^{-1}. \end{aligned} \tag{2.7}$$

Define the error matrices

$$\widetilde{X}^{(k+1)} = X^{(k+1)} - X^*, \quad \widetilde{X}^{(k)} = X^{(k)} - X^*.$$

The error of the $(k+1)$-th iteration is

$$\begin{aligned} \widetilde{X}^{(k+1)} =& \widetilde{X}^{(k)} - \frac{1}{2}(\alpha P + H(A))^{-1}[A\widetilde{X}^{(k)} + \widetilde{X}^{(k)}B] - \frac{1}{2}[A\widetilde{X}^{(k)} + \widetilde{X}^{(k)}B](\beta Q + H(B))^{-1} \\ =& \frac{1}{2}\Big[2\widetilde{X}^{(k)} - (\alpha P + H(A))^{-1}A\widetilde{X}^{(k)} - (\alpha P + H(A))^{-1}\widetilde{X}^{(k)}B \\ &- A\widetilde{X}^{(k)}(\beta Q + H(B))^{-1} - \widetilde{X}^{(k)}B(\beta Q + H(B))^{-1}\Big]. \end{aligned}$$

Taking the $\|\cdot\|_2$ norm on both sides of the above equation, we have

$$\begin{aligned} \|\widetilde{X}^{(k+1)}\|_2 \le& \frac{1}{2}\Big(\|I_m - (\alpha P + H(A))^{-1}A\|_2 + \|(\alpha P + H(A))^{-1}\|_2\|B\|_2 \\ &+ \|A\|_2\|(\beta Q + H(B))^{-1}\|_2 + \|I_n - B(\beta Q + H(B))^{-1}\|_2\Big)\|\widetilde{X}^{(k)}\|_2. \end{aligned}$$

Since $A = (\alpha P + H(A)) - (\alpha P - S(A))$,

$$\begin{aligned}
&\|I_m - (\alpha P + H(A))^{-1}A\|_2 \\
=&\|(\alpha P + H(A))^{-1}(\alpha P + H(A)) - (\alpha P + H(A))^{-1}A\|_2 \\
=&\|(\alpha P + H(A))^{-1}(\alpha P + H(A) - A)\|_2 \\
=&\|(\alpha P + H(A))^{-1}(\alpha P - S(A))\|_2.
\end{aligned} \tag{2.8}$$

By performing an identity transformation on (2.8), we obtain

$$\begin{aligned}
&\|(\alpha P + H(A))^{-1}(\alpha P - S(A))\|_2 \\
=&\|P^{-\frac{1}{2}}P^{\frac{1}{2}}(\alpha P + H(A))^{-1}P^{\frac{1}{2}}P^{-\frac{1}{2}}(\alpha P - S(A))P^{-\frac{1}{2}}P^{\frac{1}{2}}\|_2 \\
=&\|P^{-\frac{1}{2}}(P^{-\frac{1}{2}}(\alpha P + H(A))P^{-\frac{1}{2}})^{-1}P^{-\frac{1}{2}}(\alpha P - S(A))P^{-\frac{1}{2}}P^{\frac{1}{2}}\|_2 \\
=&|P^{-\frac{1}{2}}(\alpha I_m + P^{-\frac{1}{2}}H(A)P^{-\frac{1}{2}})^{-1}(\alpha I_m - P^{-\frac{1}{2}}S(A)P^{-\frac{1}{2}})P^{\frac{1}{2}}\|_2 \\
=&|P^{-\frac{1}{2}}(\alpha I_m + \widetilde{H}(A))^{-1}(\alpha I_m - \widetilde{S}(A))P^{\frac{1}{2}}\|_2.
\end{aligned} \tag{2.9}$$

Similarly, we can deduce

$$\|(\alpha P + H(A))^{-1}\|_2 = \|P^{-\frac{1}{2}}(\alpha I_m + \widetilde{H}(A))^{-1}P^{-\frac{1}{2}}\|_2. \tag{2.10}$$

$$\|(\beta Q + H(B))^{-1}\|_2 = \|Q^{-\frac{1}{2}}(\beta I_n + \widetilde{H}(B))^{-1}Q^{-\frac{1}{2}}\|_2. \tag{2.11}$$

$$\begin{aligned}
\|I_n - B(\beta Q + H(B))^{-1}\|_2 =&\|(\beta Q - S(B))(\beta Q + H(B))^{-1}\|_2 \\
=&\|Q^{\frac{1}{2}}(\beta I_n - \widetilde{S}(B))(\beta I_n + \widetilde{H}(B))^{-1}Q^{-\frac{1}{2}}\|_2.
\end{aligned} \tag{2.12}$$

Following from (2.8)–(2.12), we obtain

$$\begin{aligned}
&\|I_m - (\alpha P + H(A))^{-1}A\|_2 + \|(\alpha P + H(A))^{-1}\|_2\|B\|_2 + \|I_n - B(\beta Q + H(B))^{-1}\|_2 \\
&+ \|A\|_2\|(\beta Q + H(B))^{-1}\|_2 \\
=&\|P^{-\frac{1}{2}}(\alpha I_m + \widetilde{H}(A))^{-1}(\alpha I_m - \widetilde{S}(A))P^{\frac{1}{2}}\|_2 + \|Q^{\frac{1}{2}}(\beta I_n - \widetilde{S}(B))(\beta I_n + \widetilde{H}(B))^{-1}Q^{-\frac{1}{2}}\|_2 \\
&+ \|P^{-\frac{1}{2}}(\alpha I_m + \widetilde{H}(A))^{-1}P^{-\frac{1}{2}}\|_2\|B\|_2 + \|A\|_2\|Q^{-\frac{1}{2}}(\beta I_n + \widetilde{H}(B))^{-1}Q^{-\frac{1}{2}}\|_2 \\
\leq&\|(\alpha I_m + \widetilde{H}(A))^{-1}\|_2\left(\|P^{\frac{1}{2}}\|_2\|P^{-\frac{1}{2}}\|_2\|\alpha I_m - \widetilde{S}(A)\|_2 + \|P^{-\frac{1}{2}}\|_2^2\|B\|_2\right) \\
&+ \|(\beta I_n + \widetilde{H}(B))^{-1}\|_2\left(\|Q^{\frac{1}{2}}\|_2\|Q^{-\frac{1}{2}}\|_2\|\beta I_n - \widetilde{S}(B)\|_2 + \|A\|_2\|Q^{-\frac{1}{2}}\|_2^2\right) \\
\leq&\frac{\sqrt{\lambda_{\max}(P)\lambda_{\min}(P)(\alpha^2 + \sigma_{\max}^2(\widetilde{S}(A)))} + \sigma_{\max}(B)}{\lambda_{\min}(P)(\alpha + \lambda_{\min}(\widetilde{H}(A)))} \\
&+ \frac{\sqrt{\lambda_{\max}(Q)\lambda_{\min}(Q)(\beta^2 + \sigma_{\max}^2(\widetilde{S}(B)))} + \sigma_{\max}(A)}{\lambda_{\min}(Q)(\beta + \lambda_{\min}(\widetilde{H}(B)))}.
\end{aligned} \tag{2.13}$$

If α and β satisfy (2.6), it is evident that

$$\|\widetilde{X}^{(k+1)}\|_2 < \|\widetilde{X}^{(k)}\|_2 < \cdots < \|\widetilde{X}^{(1)}\|_2,$$

i.e., $\widetilde{X}^{(k)} \to 0$ as $k \to \infty$. The proof is complete. □

Remark 2. *When $P = I_m$ and $Q = I_n$, (2.6) turns to*

$$\frac{\sqrt{\alpha^2 + \sigma^2_{\max}(S(A))} + \sigma_{\max}(B)}{\alpha + \lambda_{\min}(H(A))} + \frac{\sqrt{\beta^2 + \sigma^2_{\max}(S(B))} + \sigma_{\max}(A)}{\beta + \lambda_{\min}(H(B))} < 2.$$

Remark 3. *When $P = H(A)$ and $Q = H(B)$, (2.6) turns to*

$$\frac{\sqrt{\lambda_{\max}(P)\lambda_{\min}(P)(\alpha^2 + \sigma^2_{\max}(\widetilde{S}(A)))} + \sigma_{\max}(B)}{\lambda_{\min}(P)(\alpha + 1)} + \frac{\sqrt{\lambda_{\max}(Q)\lambda_{\min}(Q)(\beta^2 + \sigma^2_{\max}(\widetilde{S}(B)))} + \sigma_{\max}(A)}{\lambda_{\min}(Q)(\beta + 1)} < 2,$$

where $\widetilde{S}(A) = H(A)^{-\frac{1}{2}}S(A)H(A)^{-\frac{1}{2}}$, and $\widetilde{S}(B) = H(B)^{-\frac{1}{2}}S(B)H(B)^{-\frac{1}{2}}$.

2.2. The adaptive HSSHI algorithm

For the case of $P = H(A)$ and $Q = H(B)$ in Algorithm 1, the varied parameters α_{k+1} and β_{k+1} are adopted in each iteration, and we obtain the adaptive HSSHI (AHSSHI) algorithm as follows.

Algorithm 2 The AHSSHI algorithm

Input: Given an initial solution $X^{(1)}$, two preconditioners P and Q, as well as the parameters α_2 and β_2

Output: $X^{(k+1)}$

1: **For** $k = 1, 2, \cdots$ **, until it converges, solve**
2: $(\alpha_{k+1}P + H(A))X_1^{(k+1)} = (\alpha_{k+1}P + H(A))X^{(k)} + C - AX^{(k)} - X^{(k)}B$,
3: $X_2^{(k+1)}(\beta_{k+1}Q + H(B)) = X^{(k)}(\beta_{k+1}Q + H(B)) + C - AX^{(k)} - X^{(k)}B$,
4: $X^{(k+1)} = \frac{X_1^{(k+1)} + X_2^{(k+1)}}{2}$.
5: **End**

In the following, we investigate how to obtain the parameters by the minimum residual technique. Denote the k-th residual as $R^{(k)} = C - AX^{(k)} - X^{(k)}B$. According to Algorithm 2, it follows that

$$X^{(k+1)} = X^{(k)} + \frac{1}{2}m_{k+1}H(A)^{-1}R^{(k)} + \frac{1}{2}n_{k+1}R^{(k)}H(B)^{-1},$$

where $m_{k+1} = \frac{1}{\alpha_{k+1}+1}$ and $n_{k+1} = \frac{1}{\beta_{k+1}+1}$.

The $(k+1)$-th residual can be further expressed as

$$R^{(k+1)} = R^{(k)} - m_{k+1}M^{(k)} - n_{k+1}N^{(k)} \tag{2.14}$$

with

$$\begin{cases} M^{(k)} = \frac{1}{2}(AH(A)^{-1}R^{(k)} + H(A)^{-1}R^{(k)}B), \\ N^{(k)} = \frac{1}{2}(AR^{(k)}H(B)^{-1} + R^{(k)}H(B)^{-1}B). \end{cases}$$

Taking the F-norm on both sides of (2.14), we have

$$\begin{aligned}\|R^{(k+1)}\|_F^2 &= tr[(R^{(k)} - m_{k+1}M^{(k)} - n_{k+1}N^{(k)})^T(R^{(k)} - m_{k+1}M^{(k)} - n_{k+1}N^{(k)})] \\ &= \|R^{(k)}\|_F^2 - 2m_{k+1}tr((M^{(k)})^T R^{(k)}) - 2n_{k+1}tr((N^{(k)})^T R^{(k)}) \\ &+2m_{k+1}n_{k+1}tr((M^{(k)})^T N^{(k)}) + m_{k+1}^2\|M^{(k)}\|_F^2 + n_{k+1}^2\|N^{(k)}\|_F^2.\end{aligned}$$

Let $\phi(m_{k+1}, n_{k+1}) = \|R^{(k+1)}\|_F^2$. Find the unique stationary point of the function $\phi(m_{k+1}, n_{k+1})$, i.e.,

$$m_{k+1} = \frac{a_k e_k - c_k b_k}{d_k e_k - b_k^2}, n_{k+1} = \frac{c_k d_k - a_k b_k}{d_k e_k - b_k^2},$$

where $a_k = tr((M^{(k)})^T R^{(k)})$, $b_k = tr((M^{(k)})^T N^{(k)})$, $c_k = tr((N^{(k)})^T R^{(k)})$, $d_k = \|M^{(k)}\|_F^2$, $e_k = \|N^{(k)}\|_F^2$. It is easy to obtain that

$$\begin{cases}\alpha_{k+1} = \dfrac{d_k e_k - b_k^2}{a_k e_k - c_k b_k} - 1, \\ \beta_{k+1} = \dfrac{d_k e_k - b_k^2}{c_k d_k - a_k b_k} - 1.\end{cases} \tag{2.15}$$

Remark 4. *Parameters α_{k+1} and β_{k+1} need to be updated at each step using the trace. It indeed takes time to compute these parameters, but they effectively minimize the residual at each step, thereby significantly enhancing the computational efficiency of the algorithms.*

Remark 5. *On the one hand, we can directly utilize the formulas (2.15) to give the optimal parameters α_{k+1} and β_{k+1} for the AHSSHI algorithm (see Example 3 in Section 4).*

On the other hand, we can refer to the values of α_{k+1} and β_{k+1} obtained by (2.15) to find the quasi-optimal fixed parameters for the HSSHI algorithm. Specifically, we can first observe the varied parameters by (2.15) for the small-scale cases of the problem. If the values do not change significantly at each step, we can determine the quasi-optimal parameters that are fixed at each step based on these values (see Example 4 in Section 4).

3. The HSSHI algorithm with momentum acceleration

In order to improve the efficiency of the HSSHI algorithm, we introduce a momentum term into the iterative process, thereby establishing the MHSSHI algorithm as follows:

Algorithm 3 The MHSSHI algorithm

Input: Given two initial solution vectors $X^{(0)}$ and $X^{(1)}$, two preconditioners P and Q, as well as the parameters $\tilde{\alpha}, \tilde{\beta}$ and γ

Output: $X^{(k+1)}$

For $k = 1, 2, \cdots$ **, until it converges, solve**

$(\tilde{\alpha}P + H(A))X_1^{(k+1)} = (\tilde{\alpha}P + H(A))X^{(k)} + C - AX^{(k)} - X^{(k)}B,$

$X_2^{(k+1)}(\tilde{\beta}Q + H(B)) = X^{(k)}(\tilde{\beta}Q + H(B)) + C - AX^{(k)} - X^{(k)}B,$

$X^{(k+1)} = \frac{X_1^{(k+1)} + X_2^{(k+1)}}{2} + \gamma(X^{(k)} - X^{(k-1)}).$

End

Remark 6. *When γ is chosen to be 0, the algorithm degenerates into the HSSHI algorithm.*

3.1. Convergence analysis

In this section, we mainly discuss the convergence property of Algorithm 3. For the sake of convenience in the proof, we first present a lemma as follows:

Lemma 1. *[8] Both roots of the real quadratic equation* $x^2 - bx + c = 0$ *are less than one in modulus if and only if* $|c| < 1$ *and* $|b| < 1 + c$.

Based on the lemma, the convergence result of Algorithm 3 is given in the following theorem.

Theroem 2. *Let* $\widetilde{H}(A) = P^{-\frac{1}{2}}H(A)P^{-\frac{1}{2}}$, $\widetilde{S}(A) = P^{-\frac{1}{2}}S(A)P^{-\frac{1}{2}}$, $\widetilde{H}(B) = Q^{-\frac{1}{2}}H(B)Q^{-\frac{1}{2}}$, *and* $\widetilde{S}(B) = Q^{-\frac{1}{2}}S(B)Q^{-\frac{1}{2}}$. *Assume* X^* *be the solution of Eq (1.1). The iterative solution* $X^{(k)}$ *generated by Algorithm 3 converges to* X^* *for any initial value if and only if the parameters* $\tilde{\alpha}$, $\tilde{\beta}$ *and* γ *satisfy*

$$\begin{cases} 0 < \gamma < \frac{1}{2}, \\ \frac{\sqrt{\lambda_{\max}(P)\lambda_{\min}(P)(\tilde{\alpha}^2 + \sigma_{\max}^2(\widetilde{S}(A)))} + \sigma_{\max}(B)}{\lambda_{\min}(P)(\tilde{\alpha} + \lambda_{\min}(\widetilde{H}(A)))} \\ \quad + \frac{\sqrt{\lambda_{\max}(Q)\lambda_{\min}(Q)(\tilde{\beta}^2 + \sigma_{\max}^2(\widetilde{S}(B)))} + \sigma_{\max}(A)}{\lambda_{\min}(Q)(\tilde{\beta} + \lambda_{\min}(\widetilde{H}(B)))} < 2 - 4\gamma, \end{cases} \tag{3.1}$$

where $\sigma_{\max}(E)$, $\lambda_{\max}(E)$, *and* $\lambda_{\min}(E)$ *are the maximum singular value and the maximum and minimum eigenvalues of the matrix E, respectively.*

Proof: From Algorithm 3, it turns out that $(k+1)$-th iteration can be rewritten as

$$\begin{aligned} X^{(k+1)} =& X^{(k)} + \frac{1}{2}(\tilde{\alpha}P + H(A))^{-1}[C - AX^{(k)} - X^{(k)}B] \\ &+ \frac{1}{2}[C - AX^{(k)} - X^{(k)}B](\tilde{\beta}Q + H(B))^{-1} + \gamma(X^{(k)} - X^{(k-1)}). \end{aligned} \tag{3.2}$$

Define the error matrices

$$\widetilde{X}^{(k+1)} = X^{(k+1)} - X^*, \quad \widetilde{X}^{(k)} = X^{(k)} - X^*.$$

From (3.2) it follows that

$$\begin{aligned} \widetilde{X}^{(k+1)} =& \widetilde{X}^{(k)} - \frac{1}{2}(\tilde{\alpha}P + H(A))^{-1}[A\widetilde{X}^{(k)} + \widetilde{X}^{(k)}B] \\ &- \frac{1}{2}[A\widetilde{X}^{(k)} + \widetilde{X}^{(k)}B](\tilde{\beta}Q + H(B))^{-1} + \gamma(\widetilde{X}^{(k)} - \widetilde{X}^{(k-1)}). \end{aligned}$$

Taking the $\|\cdot\|_2$ norm on both sides of the above equation, we have

$$\begin{aligned} \|\widetilde{X}^{(k+1)}\|_2 =& \|\widetilde{X}^{(k)} - \frac{1}{2}(\tilde{\alpha}P + H(A))^{-1}[A\widetilde{X}^{(k)} + \widetilde{X}^{(k)}B] \\ &- \frac{1}{2}[A\widetilde{X}^{(k)} + \widetilde{X}^{(k)}B](\tilde{\beta}Q + H(B))^{-1} + \gamma(\widetilde{X}^{(k)} - \widetilde{X}^{(k-1)})\|_2 \\ =& \frac{1}{2}\|(I_m - (\tilde{\alpha}P + H(A))^{-1}A)\widetilde{X}^{(k)} + 2\gamma\widetilde{X}^{(k)} - (\tilde{\alpha}P + H(A))^{-1}\widetilde{X}^{(k)}B \end{aligned} \tag{3.3}$$

$$- A\widetilde{X}^{(k)}(\tilde{\beta}Q + H(B))^{-1} + \widetilde{X}^{(k)}(I_n - B(\tilde{\beta}Q + H(B))^{-1}) - 2\gamma\widetilde{X}^{(k-1)}\|_2$$

$$\leq \frac{1}{2}\Big(\|I_m - (\tilde{\alpha}P + H(A))^{-1}A\|_2 + 2\gamma + \|(\tilde{\alpha}P + H(A))^{-1}\|_2\|B\|_2$$

$$+ \|A\|_2\|(\tilde{\beta}Q + H(B))^{-1}\|_2 + \|I_n - B(\tilde{\beta}Q + H(B))^{-1}\|_2\Big)\|\widetilde{X}^{(k)}\|_2$$

$$+ \gamma\|\widetilde{X}^{(k-1)}\|_2.$$

Let

$$H = \begin{bmatrix} q_1 + \gamma + q_2 + q_3 + q_4 & \gamma \\ 1 & 0 \end{bmatrix},$$

where $q_1 = \frac{1}{2}\|I_m - (\tilde{\alpha}P + H(A))^{-1}A\|_2$, $q_2 = \frac{1}{2}\|(\tilde{\alpha}P + H(A))^{-1}\|_2\|B\|_2$, $q_3 = \frac{1}{2}\|I_n - B(\tilde{\beta}Q + H(B))^{-1}\|_2$, $q_4 = \frac{1}{2}\|A\|_2\|(\tilde{\beta}Q + H(B))^{-1}\|_2$. Then from (3.3) it is clear that

$$\begin{bmatrix} \|\widetilde{X}^{(k+1)}\|_2 \\ \|\widetilde{X}^{(k)}\|_2 \end{bmatrix} \leq H \begin{bmatrix} \|\widetilde{X}^{(k)}\|_2 \\ \|\widetilde{X}^{(k-1)}\|_2 \end{bmatrix} \leq H^k \begin{bmatrix} \|\widetilde{X}^{(1)}\|_2 \\ \|\widetilde{X}^{(0)}\|_2 \end{bmatrix}.$$

If $\rho(H) < 1$, then $\|\widetilde{X}^{(k)}\|_2 \to 0$ as $k \to \infty$.

In the following discussion, we concentrate on determining $\tilde{\alpha}$, $\tilde{\beta}$, and γ to ensure that $\rho(H) < 1$, thereby guaranteeing the convergence of the algorithm. The characteristic equation for H is

$$\lambda^2 - \lambda(q_1 + q_2 + q_3 + q_4 + \gamma) - \gamma = 0,$$

where λ is an eigenvalue of matrix H. It then follows from Lemma 1 that $|\lambda| < 1$ if and only if

$$\begin{cases} |\gamma| < 1, \\ |q_1 + q_2 + q_3 + q_4 + \gamma| < 1 - \gamma, \end{cases}$$

i.e.,

$$\begin{cases} 0 < \gamma < \dfrac{1}{2}, \\ \|I_m - (\tilde{\alpha}P + H(A))^{-1}A\|_2 + \|(\tilde{\alpha}P + H(A))^{-1}\|_2\|B\|_2 \\ \quad + \|I_n - B(\tilde{\beta}Q + H(B))^{-1}\|_2 + \|A\|_2\|(\tilde{\beta}Q + H(B))^{-1}\|_2 < 2 - 4\gamma. \end{cases} \tag{3.4}$$

Together with (2.13) and (3.4), (3.1) is obtained. Thus, the proof is complete. □

Remark 7. *When $P = I_m$ and $Q = I_n$,* (3.1) *becomes*

$$\begin{cases} 0 < \gamma < \dfrac{1}{2}, \\ \dfrac{\sqrt{\tilde{\alpha}^2 + \sigma^2_{\max}(S(A))} + \sigma_{\max}(B)}{\tilde{\alpha} + \lambda_{\min}(H(A))} + \dfrac{\sqrt{\tilde{\beta}^2 + \sigma^2_{\max}(S(B))} + \sigma_{\max}(A)}{\tilde{\beta} + \lambda_{\min}(H(B))} < 2 - 4\gamma. \end{cases}$$

Remark 8. *When $P = H(A)$ and $Q = H(B)$,* (3.1) *leads to*

$$\begin{cases} 0 < \gamma < \dfrac{1}{2}, \\ \dfrac{\sqrt{\lambda_{\max}(P)\lambda_{\min}(P)(\tilde{\alpha}^2 + \sigma^2_{\max}(\widetilde{S}(A)))} + \sigma_{\max}(B)}{\lambda_{\min}(P)(\tilde{\alpha}+1)} \\ \quad + \dfrac{\sqrt{\lambda_{\max}(Q)\lambda_{\min}(Q)(\tilde{\beta}^2 + \sigma^2_{\max}(\widetilde{S}(B)))} + \sigma_{\max}(A)}{\lambda_{\min}(Q)(\tilde{\beta}+1)} < 2 - 4\gamma, \end{cases}$$

where $\widetilde{S}(A) = H(A)^{-\frac{1}{2}}S(A)H(A)^{-\frac{1}{2}}$, and $\widetilde{S}(B) = H(B)^{-\frac{1}{2}}S(B)H(B)^{-\frac{1}{2}}$.

3.2. The adaptive MHSSHI (AMHSSHI) algorithm

When $P = H(A)$ and $Q = H(B)$ in Algorithm 3, the adaptive MHSSHI algorithm can be similarly formulated as Algorithm 2 with the varied parameters $\tilde{\alpha}_{k+1}$,$\tilde{\beta}_{k+1}$ and γ_{k+1}. The algorithm is detailed below:

Algorithm 4 The AMHSSHI algorithm

Input: Given two initial solution vectors $X^{(0)}$ and $X^{(1)}$, two preconditioners P and Q, as well as the parameters $\tilde{\alpha}_2, \tilde{\beta}_2$ and γ_2

Output: $X^{(k+1)}$

1: **For** $k = 1, 2, \cdots$, **until it converges, solve**

2: $(\tilde{\alpha}_{k+1}P + H(A))X_1^{(k+1)} = (\tilde{\alpha}_{k+1}P + H(A))X^{(k)} + C - AX^{(k)} - X^{(k)}B,$

3: $X_2^{(k+1)}(\tilde{\beta}_{k+1}Q + H(B)) = X^{(k)}(\tilde{\beta}_{k+1}Q + H(B)) + C - AX^{(k)} - X^{(k)}B,$

4: $X^{(k+1)} = \frac{X_1^{(k+1)} + X_2^{(k+1)}}{2} + \gamma_{k+1}(X^{(k)} - X^{(k-1)}).$

5: **End**

Below, we mainly provide the specific expressions for the optimal parameters through the minimal residual technique. Denote the k-th residual by $R^{(k)} = C - AX^{(k)} - X^{(k)}B$. From Algorithm 4, we have

$$X^{(k+1)} = X^{(k)} + \frac{1}{2}\tilde{m}_{k+1}H(A)^{-1}R^{(k)} + \frac{1}{2}\tilde{n}_{k+1}R^{(k)}H(B)^{-1} + \gamma_{k+1}(X^{(k)} - X^{(k-1)}),$$

where $\tilde{m}_{k+1} = \frac{1}{\tilde{\alpha}_{k+1}+1}$ and $\tilde{n}_{k+1} = \frac{1}{\tilde{\beta}_{k+1}+1}$.

The $(k+1)$-th residual can be represented as

$$R^{(k+1)} = R^{(k)} - \tilde{m}_{k+1}\tilde{M}^{(k)} - \tilde{n}_{k+1}\tilde{N}^{(k)} - \gamma_{k+1}\tilde{H}^{(k)} \tag{3.5}$$

with

$$\begin{cases} \tilde{M}^{(k)} = \dfrac{1}{2}(AH(A)^{-1}R^{(k)} + H(A)^{-1}R^{(k)}B), \\ \tilde{N}^{(k)} = \dfrac{1}{2}(AR^{(k)}H(B)^{-1} + R^{(k)}H(B)^{-1}B), \\ \tilde{H}^{(k)} = R^{(k-1)} - R^{(k)}. \end{cases}$$

Let $\psi(\tilde{m}_{k+1}, \tilde{n}_{k+1}, \gamma_{k+1}) = ||R^{(k+1)}||_F^2$. Taking the F-norm on both sides of (3.5), we have

$$\begin{aligned}
&\psi(\tilde{m}_{k+1}, \tilde{n}_{k+1}, \gamma_{k+1}) \\
&= tr[(R^{(k)} - \tilde{m}_{k+1}\tilde{M}^{(k)} - \tilde{n}_{k+1}\tilde{N}^{(k)} - \gamma_{k+1}\tilde{H}^{(k)})^T(R^{(k)} - \tilde{m}_{k+1}\tilde{M}^{(k)} - \tilde{n}_{k+1}\tilde{N}^{(k)} - \gamma_{k+1}\tilde{H}^{(k)})] \\
&= (\gamma_{k+1}+1)^2||R^{(k)}||_F^2 - 2\tilde{m}_{k+1}(\gamma_{k+1}+1)tr((\tilde{M}^{(k)})^T R^{(k)}) - 2\tilde{n}_{k+1}(\gamma_{k+1}+1)tr((\tilde{N}^{(k)})^T R^{(k)}) \\
&+2\gamma_{k+1}\tilde{m}_{k+1}tr((\tilde{M}^{(k)})^T R^{(k-1)}) + 2\gamma_{k+1}\tilde{n}_{k+1}tr((\tilde{N}^{(k)})^T R^{(k-1)}) \\
&-2(\gamma_{k+1}+1)\gamma_{k+1}tr((R^{(k)})^T R^{(k-1)}) + \gamma_{k+1}^2||R^{(k-1)}||_F^2 + 2\tilde{m}_{k+1}\tilde{n}_{k+1}tr((\tilde{M}^{(k)})^T\tilde{N}^{(k)}) \\
&+\tilde{m}_{k+1}^2||\tilde{M}^{(k)}||_F^2 + \tilde{n}_{k+1}^2||\tilde{N}^{(k)}||_F^2.
\end{aligned}$$

Assume that

$$\begin{cases}
c_k^{(1)} := ||\tilde{M}^{(k)}||_F^2, \quad c_k^{(2)} := tr((\tilde{M}^{(k)})^T\tilde{N}^{(k)}), \\
c_k^{(3)} := tr((\tilde{M}^{(k)})^T R^{(k-1)}) - tr((\tilde{M}^{(k)})^T R^{(k)}), \\
c_k^{(4)} := tr((\tilde{M}^{(k)})^T R^{(k)}), \quad c_k^{(5)} := (tr((\tilde{N}^{(k)})^T R^{(k)}), \\
c_k^{(6)} := tr((\tilde{N}^{(k)})^T R^{(k-1)}) - tr((\tilde{N}^{(k)})^T R^{(k)}), \\
c_k^{(7)} := ||\tilde{N}^{(k)}||_F^2, \quad c_k^{(8)} := (tr((R^{(k)})^T R^{(k-1)} - ||R^{(k)}||_F^2, \\
c_k^{(9)} := ||R^{(k-1)}||_F^2 - 2tr((R^{(k)})^T R^{(k-1)})) + ||R^{(k)}||_F^2,
\end{cases}$$

and

$$\begin{cases}
d_k^{(1)} := c_k^{(6)} * c_k^{(3)} - c_k^{(9)} * c_k^{(2)}, \quad d_k^{(2)} := c_k^{(2)} * c_k^{(6)} - c_k^{(7)} * c_k^{(3)}, \\
d_k^{(3)} := c_k^{(1)} * c_k^{(6)} - c_k^{(2)} * c_k^{(3)}, \quad d_k^{(4)} := c_k^{(7)} * c_k^{(9)} - \left(c_k^{(6)}\right)^2, \\
d_k^{(5)} := c_k^{(5)} * c_k^{(9)} - c_k^{(8)} * c_k^{(6)}, \quad d_k^{(6)} := c_k^{(4)} * c_k^{(6)} - c_k^{(5)} * c_k^{(3)}, \\
d_k^{(7)} := \left(c_k^{(2)}\right)^2 - c_k^{(1)} * c_k^{(7)}, \quad d_k^{(8)} := c_k^{(4)} * c_k^{(2)} - c_k^{(1)} * c_k^{(5)}, \\
d_k^{(9)} := c_k^{(5)} * c_k^{(3)} - c_k^{(8)} * c_k^{(2)}.
\end{cases}$$

Then, the unique stationary point of the function $\psi(\tilde{m}_{k+1}, \tilde{n}_{k+1}, \gamma_{k+1})$ is

$$\begin{cases}
\tilde{m}_{k+1} = \dfrac{d_k^{(6)}d_k^{(4)} - d_k^{(5)}d_k^{(2)}}{d_k^{(1)}d_k^{(2)} + d_k^{(3)}d_k^{(4)}}, \\
\tilde{n}_{k+1} = \dfrac{d_k^{(8)}d_k^{(1)} + d_k^{(9)}d_k^{(3)}}{d_k^{(7)}d_k^{(1)} - d_k^{(2)}d_k^{(3)}}, \\
\gamma_{k+1} = \dfrac{d_k^{(8)}d_k^{(2)} + d_k^{(9)}d_k^{(7)}}{d_k^{(1)}d_k^{(7)} - d_k^{(3)}d_k^{(2)}}.
\end{cases}$$

Therefore, it is easy to obtain that

$$\begin{cases}
\tilde{\alpha}_{k+1} = \dfrac{d_k^{(1)}d_k^{(2)} + d_k^{(3)}d_k^{(4)}}{d_k^{(6)}d_k^{(4)} - d_k^{(5)}d_k^{(2)}} - 1, \\
\tilde{\beta}_{k+1} = \dfrac{d_k^{(7)}d_k^{(1)} - d_k^{(2)}d_k^{(3)}}{d_k^{(8)}d_k^{(1)} + d_k^{(9)}d_k^{(3)}} - 1, \\
\gamma_{k+1} = \dfrac{d_k^{(8)}d_k^{(2)} + d_k^{(9)}d_k^{(7)}}{d_k^{(1)}d_k^{(7)} - d_k^{(3)}d_k^{(2)}}.
\end{cases} \tag{3.6}$$

Remark 9. *Similar explanations can be obtained by referring to Remarks 4 and 5.*

4. Numerical examples

In this section, several numerical examples are given to examine the effectiveness of the proposed algorithms compared to the HSS [2], NPHSS [11], AGMI [19], and B-S [3] algorithms. All test problems are performed under Matlab on a personal computer with a 1.61 GHz central processing unit (Intel(R) Core(TM) i7-10710), 16GB memory, and Windows 10 operating system. The initial matrices are set to be zero matrices, and the iterations are terminated if the relative residual norm in the current step satisfies

$$RRN := \frac{\|C - AX^{(k)} - X^{(k)}B\|}{\|C - AX^{(0)} - X^{(0)}B\|} \leq 10^{-6}.$$

The number of iterations (denoted as IT), the computing time in seconds (denoted as CPU) and RRN are used to test the efficiency of these algorithms.

Example 1. *The matrices A and B in Eq (1.1) are given as*

$$\begin{cases} A = \mathrm{diag}(1, 2, \cdots, n) + rL^T, \\ B = 2^{-t}I_n + \mathrm{diag}(1, 2, \cdots, n) + rL^T + 2^{-t}L, \end{cases}$$

where L is the strictly lower triangular matrix having ones in the lower triangle part, $r = 2$ and $t = \frac{1}{2}$. The right-hand side is given by the equation $C = AX + XB$, where X is defined as $X(i, j) = 1$ for all $1 \leq i, j \leq n$.

Table 2 lists the numerical results for the six algorithms, and the experimentally optimal parameters employed in these algorithms are detailed in Table 1. We set $P = Q = I_n$ in both HSSHI and MHSSHI algorithms. In terms of the CPU time, the B-S, HSSHI, and MHSSHI algorithms significantly outperform the HSS, NPHSS, and AGMI algorithms. Moreover, the HSSHI and MHSSHI algorithms need remarkbly fewer iterations than the AGMI, HSS, and NPHSS algorithms. In addition, as illustrated in Figure 1, the RRN for the MHSSHI algorithm decreases the fastest, followed by the HSSHI algorithm, while the HSS algorithm shows a very slow decrease.

Table 1. The experimentally optimal parameters for Example 1.

Algorithms		100	200	300	400
HSS	μ_1	7.64	10.57	12.79	14.65
	μ_2	7.64	10.57	12.79	14.65
NPHSS	$\hat{\alpha}$	39.15	78.33	118.01	156.72
HSSHI	α	1881.00	3560.00	5571.00	7090.00
	β	39.00	77.00	116.00	155.00
MHSSHI	$\tilde{\alpha}$	3321.00	6011.00	8811.00	11025.00
	$\tilde{\beta}$	38.00	75.00	113.00	152.00
	γ	0.01	0.01	0.01	0.01

Table 2. Numerical results of six algorithms for Example 1.

Algorithms		100	200	300	400
HSS	IT	71	102	126	146
	CPU	1.255	11.108	36.173	80.599
	RRN	9.138e-07	8.654e-07	8.590e-07	9.373e-07
NPHSS	IT	27	27	27	27
	CPU	0.154	0.665	1.902	4.169
	RRN	8.878e-07	8.474e-07	8.339e-07	8.267e-07
AGMI	IT	94	93	92	91
	CPU	0.162	0.622	2.135	5.891
	RRN	9.785E-07	9.744E-07	9.753E-07	9.948E-07
B-S	CPU	0.012	0.068	0.152	0.306
HSSHI	IT	10	10	10	10
	CPU	0.071	0.036	0.066	0.164
	RRN	6.785e-07	6.005e-07	5.973e-07	5.920e-07
MHSSHI	IT	10	10	10	10
	CPU	0.016	0.029	0.068	0.162
	RRN	4.920e-07	3.603e-07	3.427e-07	3.486e-07

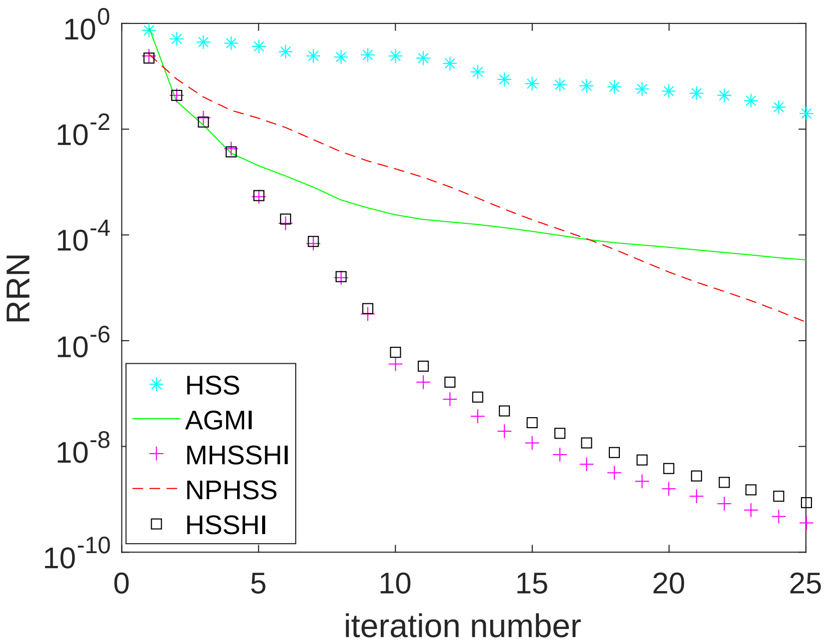

Figure 1. The convergence curves of five algorithms for Example 1.

Example 2. *The matrices A and B in Eq (1.1) are given as*

$$A = \begin{pmatrix} 10 & 1 & 1 & \cdots & 1 & 1 \\ 2 & 10 & 1 & \cdots & 1 & 1 \\ 1 & 2 & 10 & \cdots & 1 & 1 \\ \vdots & \vdots & & \ddots & 1 & 1 \\ 1 & 1 & 1 & \cdots & 2 & 10 \end{pmatrix}, \quad B = \begin{pmatrix} 8 & 1 & 1 & \cdots & 1 & 1 \\ 3 & 8 & 1 & \cdots & 1 & 1 \\ 1 & 3 & 8 & \cdots & 1 & 1 \\ \vdots & \vdots & & \ddots & 1 & 1 \\ 1 & 1 & 1 & \cdots & 3 & 8 \end{pmatrix}.$$

Let $C = AX + XB$.

Table 4 reports the numerical results for the six algorithms, and the experimentally optimal parameters used in these algorithms are detailed in Table 3. We take $P = tridiag(H(A))$ and $Q = tridiag(H(B))$ in both HSSHI and MHSSHI algorithms. Compared to the HSS algorithm, the other five algorithms show marked superiority in terms of the iteration number and computational time. In particular, the HSSHI, MHSSHI, and AGMI algorithms take considerably less CPU time than the HSS, B-S, and NPHSS algorithms, demonstrating a clear and notable advantage. Furthermore, as shown in Figure 2, the RRNs for the HSSHI and MHSSHI algorithms decrease sharply and quickly below 10^{-6} in comparison to the HSS algorithm.

Table 3. The experimentally optimal parameters for Example 2.

Algorithms		128	256	512	1024
HSS	μ_1	29.90	41.50	58.20	81.90
	μ_2	29.90	41.50	58.20	81.90
NPHSS	$\hat{\alpha}$	0.01	0.01	0.01	0.01
HSSHI	α	7.10	15.10	33.00	62.10
	β	13.70	28.20	59.00	120.10
MHSSHI	$\tilde{\alpha}$	7.10	13.10	28.10	65.10
	$\tilde{\beta}$	12.00	28.10	60.10	130.10
	γ	0.09	0.06	0.03	0.01

Table 4. Numerical results of six algorithms for Example 2.

Algorithms		128	256	512	1024
HSS	IT	32	44	62	87
	CPU	1.482	11.210	112.578	1092.848
	RRN	7.204e-07	9.336e-07	9.171e-07	9.873e-07
NPHSS	IT	3	3	3	3
	CPU	0.035	0.213	1.474	14.441
	RRN	9.698e-07	9.702e-07	9.705e-07	9.706e-07
AGMI	IT	3	3	3	3
	CPU	0.029	0.064	0.247	1.961
	RRN	1.228E-07	1.204E-08	4.862E-09	1.592E-09
B-S	CPU	0.242	0.735	3.231	17.570
HSSHI	IT	13	11	8	8
	CPU	0.010	0.051	0.204	1.225
	RRN	8.998e-07	9.406e-07	6.770e-07	4.743e-07
MHSSHI	IT	11	9	9	8
	CPU	0.009	0.035	0.221	1.256
	RRN	9.503e-07	9.869e-07	4.502e-07	4.721e-07

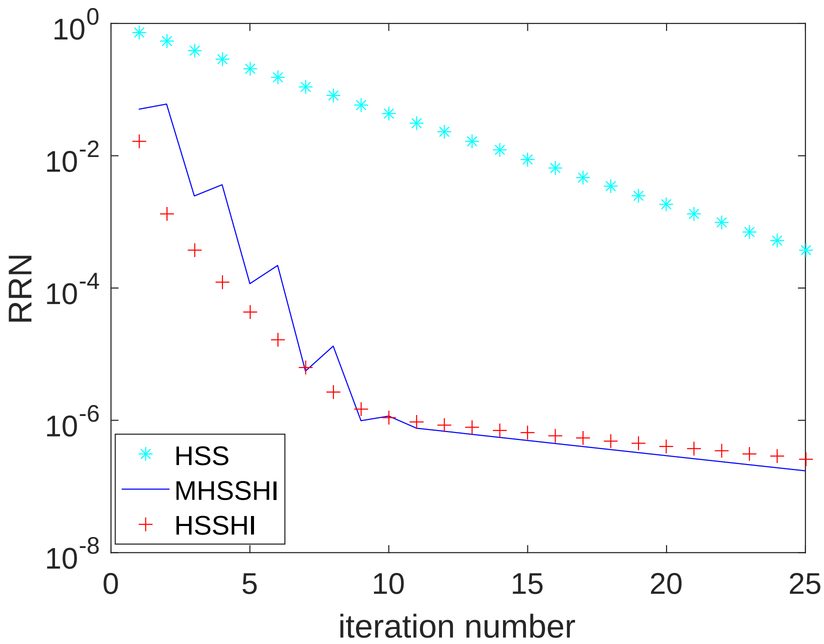

Figure 2. Convergence curves of three algorithms for Example 2.

Example 3. *The matrices in Eq (1.1) are described by*

$$\begin{cases} A(i,i) = 6 + r, (1 \le i \le m), \\ A(i,i+1) = -1, A(i+1,i) = -1 + r, (1 \le i \le m-1), \\ A(i,i+2) = A(i+2,i) = -1, (1 \le i \le m-2), \end{cases}$$

with a real number r and $B = A - 0.4I_m$. Let $C = AX + XB$.

For the cases of $r = 1$ and $r = 0.5$, the numerical results of the six algorithms are reported in Tables 6 and 8, respectively. Let $P = H(A)$ and $Q = H(B)$ in both AHSSHI and AMHSSHI algorithms. The experimentally optimal parameters for the HSS and NPHSS algorithms are presented in Tables 5 and 7, while the parameters applied in the AHSSHI and AMHSSHI algorithms are derived from (2.15) and (3.6). In comparison to the HSS and NPHSS algorithms, the B-S, AGMI, AHSSHI, and AMHSSHI algorithms require remarkably less time to achieve the desired precision in both cases. Furthermore, the AMHSSHI algorithm also shows superiority in the number of iterations. Additionally, Figures 3 and 4 illustrate that the AMHSSHI algorithm has the fastest decrease in RRN among the algorithms considered. The AHSSHI algorithm follows closely behind.

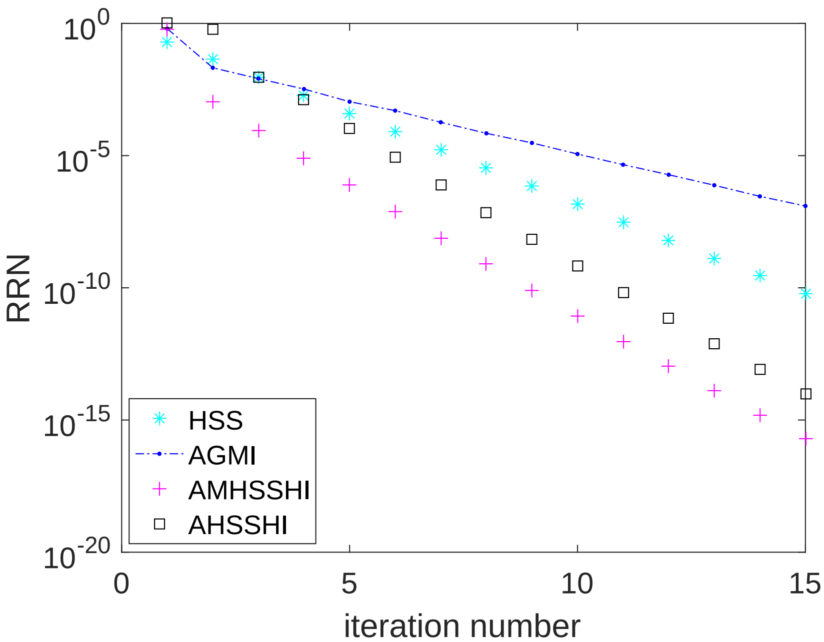

Figure 3. The convergence curves of four algorithms with $r = 1$ for Example 3.

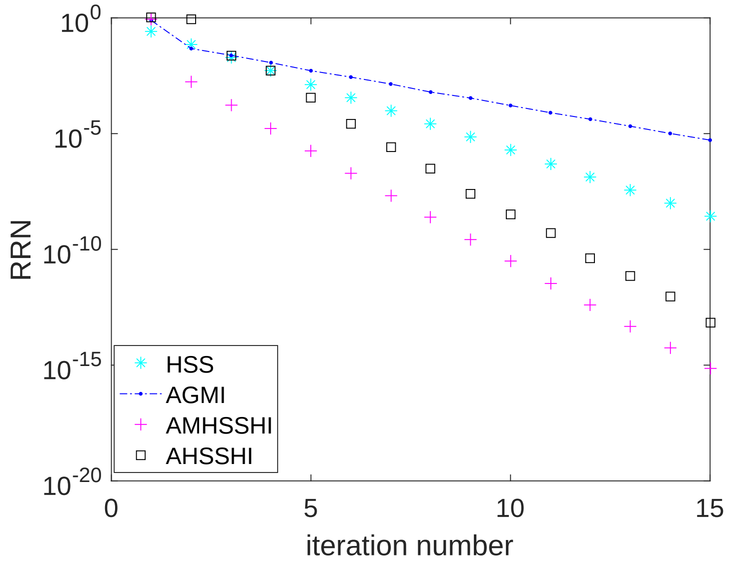

Figure 4. The convergence curves of four algorithms with $r = 0.5$ for Example 3.

Table 5. The experimentally optimal parameters with $r = 1$ for Example 3.

Algorithms		128	256	512	1024
HSS	μ_1	5.80	5.80	5.80	5.80
	μ_2	5.80	5.80	5.80	5.80
NPHSS	$\hat{\alpha}$	0.11	0.11	0.11	0.11

Table 6. Numerical results of six algorithms with $r = 1$ for Example 3.

Algorithms		256	512	1024	2048
HSS	IT	9	9	9	9
	CPU	2.452	13.640	122.652	1931.391
	RRN	7.171e-07	7.322e-07	7.399e-07	7.439e-07
NPHSS	IT	6	6	6	6
	CPU	0.372	2.191	27.032	546.929
	RRN	9.791e-07	9.633e-07	9.552e-07	5.676e-07
AGMI	IT	13	13	12	12
	CPU	0.195	1.425	10.756	116.369
	RRN	7.922e-07	4.955e-07	8.154e-07	5.231e-07
B-S	CPU	0.389	1.722	9.320	130.746
AHSSHI	IT	7	7	7	7
	CPU	0.142	0.739	8.093	108.156
	RRN	6.992e-08	4.968e-08	3.522e-08	2.493e-08
AMHSSHI	IT	5	5	5	5
	CPU	0.134	0.680	6.796	105.495
	RRN	7.899e-07	5.598e-07	3.963e-07	2.804e-07

Table 7. The optimal experimental parameters with $r = 0.5$ for Example 3.

Algorithms		128	256	512	1024
HSS	μ_1	4.86	4.86	4.86	4.86
	μ_2	4.86	4.86	4.86	4.86
NPHSS	$\hat{\alpha}$	0.10	0.10	0.10	0.10

Table 8. Numerical results of six algorithms with $r = 0.5$ for Example 3.

Algorithms		256	512	1024	2048
HSS	IT	11	11	11	11
	CPU	2.743	27.806	190.080	2385.919
	RRN	5.086e-07	5.216e-07	5.283e-07	5.317e-07
NPHSS	IT	6	6	6	6
	CPU	0.376	3.826	37.350	556.357
	RRN	5.762e-07	5.704e-07	5.674e-07	5.659e-07
AGMI	IT	18	17	17	16
	CPU	0.263	1.644	13.690	110.987
	RRN	7.048e-07	8.562e-07	5.523e-07	7.102e-07
B-S	CPU	0.225	1.461	10.318	115.147
AHSSHI	IT	8	8	8	7
	CPU	0.131	0.919	8.587	93.782
	RRN	2.606e-08	1.902e-08	1.368e-08	1.104e-07
AMHSSHI	IT	6	6	5	5
	CPU	0.148	1.289	7.265	83.453
	RRN	1.884e-07	1.342e-07	8.929e-07	6.326e-07

Example 4. *The matrices in Eq (1.1) are described as*

$$A = B = M + 2N + \frac{100}{(n+1)^2} I,$$

where $M = \text{tridiag}(-1, 2.6, -1)$ *and* $N = \text{tridiag}(0.5, 0, -0.5)$. *Let* $C = AX + XB$.

The numerical results for the five algorithms are presented in Table 10, while the experimentally optimal parameters involved in these algorithms are outlined in Table 9. We take $P = H(A)$ and $Q = H(B)$ in both HSSHI and MHSSHI algorithms. The parameters in the HSSHI and MHSSHI algorithms are obtained by referring to (2.15) and (3.6). Compared with the HSS, NPHSS, and AGMI algorithms, the HSSHI and MHSSHI algorithms achieve the required precision in significantly less time, demonstrating their effectiveness. Moreover, as shown in Figure 5, in contrast to the AGMI algorithm, the RRNs of the HSSHI and MHSSHI algorithms quickly drop to 10^{-10}.

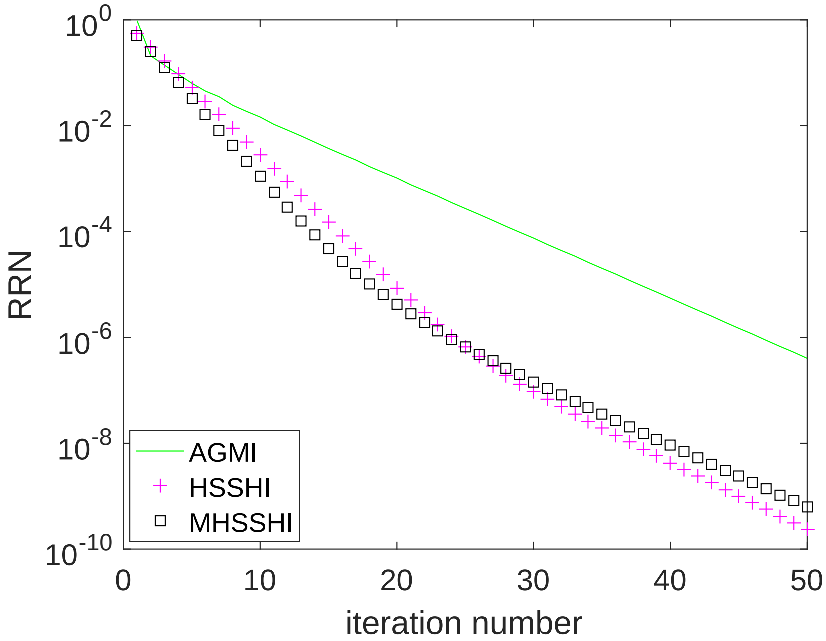

Figure 5. The convergence curves of three algorithms for Example 4.

Table 9. The experimentally optimal parameters for Example 4.

Algorithms		500	1000	1500	2000
HSS	μ_1	1.66	1.66	1.66	1.66
	μ_2	1.66	1.66	1.66	1.66
NPHSS	$\hat{\alpha}$	0.85	0.79	0.70	0.65
HSSHI	α	3.50	3.50	3.50	3.50
	β	3.50	3.50	3.50	3.50
MHSSHI	$\tilde{\alpha}$	3.10	3.10	3.10	3.10
	$\tilde{\beta}$	3.10	3.10	3.10	3.10
	γ	0.01	0.01	0.01	0.01

Table 10. Numerical results of five algorithms for Example 4.

Algorithms		500	1000	1500	2000
HSS	IT	19	19	19	19
	CPU	31.075	231.264	961.546	3030.129
	RRN	5.365e-07	5.552e-07	5.613e-07	5.643e-07
NPHSS	IT	22	21	20	20
	CPU	7.107	94.384	436.489	1241.259
	RRN	8.954e-07	7.514e-07	8.453e-07	7.222e-07
AGMI	IT	49	47	46	46
	CPU	5.842	38.390	132.914	256.424
	RRN	7.913e-07	9.162e-07	9.542e-07	8.044e-07
HSSHI	IT	27	26	25	25
	CPU	0.681	7.677	38.232	92.712
	RRN	4.782e-07	4.282e-07	5.658e-07	5.200e-07
MHSSHI	IT	26	24	23	23
	CPU	0.748	7.363	37.804	80.520
	RRN	9.178e-07	9.281e-07	9.520e-07	7.681e-07

5. Conclusions

In this paper, we provide two new algorithms for solving Eq (1.1), namely the HSSHI algorithm and the MHSSHI algorithm. The convergence properties of the proposed algorithms are presented as the parameters in the algorithms satisfy certain conditions. Moreover, the adaptive HSSHI and MHSSHI algorithms are also established when $P = H(A)$ and $Q = H(B)$. The adaptive parameters are exactly determined by minimizing the residual norms of the current step. Numerical experiments illustrate the excellent performances of our proposed algorithms. In our future work, we will explore the application of these algorithms to solving other types of Sylvester matrix equations and the absolute value equation. Additionally, when $P = I_m$ and $Q = I_n$, or when $P = tridiag(H(A))$ and $Q = tridiag(H(B))$, we have not yet provided a specific formula for determining the optimal parameters contained in the algorithms. Therefore, further research will be conducted to find effective methods for identifying the optimal parameters in these two cases.

Author contributions

Huiling Wang: Methodology, software, writing - original draft preparation; Zhaolu Tian and Yufeng Nie: Supervision. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

We would like to express our deep gratitude for the insightful comments and valuable suggestions provided by the reviewers.

Conflict of interest

The authors declare no conflicts of interest.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

References

1. Z. Z. Bai, G. H. Golub, M. K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM J. Matrix Anal. Appl.,* **24** (2003), 603–626. http://dx.doi.org/10.1137/S0895479801395458
2. Z. Z. Bai, On hermitian and skew-hermitian splitting iteration methods for continuous sylvester equations, *J. Comput. Math.,* **29** (2011), 185–198. http://dx.doi.org/10.4208/jcm. 1009-m3152
3. R. H. Bartels, G. W. Stewart, Solution of the matrix equation $AX + XB = C$, *Comm. ACM.,* **15** (1972), 820–826. http://dx.doi.org/10.1145/361573.361582
4. A. Bhaya, E. Kaszkurewicz, Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method, *Neural Networks,* **17** (2004), 65–71. http://dx.doi.org/10.1016/S0893-6080(03)00170-9
5. D. Calvetti, L. Reichel, Application of ADI iterative methods to the restoration of noisy images, *SIAM J. Matrix Anal. Appl.,* **17** (1996), 165–186. http://dx.doi.org/10.1137/S0895479894273687
6. F. Ding, T. W. Chen, Gradient based iterative algorithms for solving a class of matrix equations, *IEEE T. Automat. Contr.,* **50** (2005), 1216–1221. http://dx.doi.org/10.1109/TAC.2005.852558
7. G. M. Flagg, S. Gugercin, On the ADI method for the Sylvester equation and the optimal H_2 points, *Appl. Numer. Math.,* **64** (2013), 50–58. http://dx.doi.org/10.1016/j.apnum.2012.10.001
8. B. H. Huang, W. Li, A modified SOR-like method for absolute value equations associated with second order cones, *J. Comput. Appl. Math.,* **400** (2022), 113745. http://dx.doi.org/10.1016/j.cam.2021.113745
9. M. D. Ilic, New approaches to voltage monitoring and control, *IEEE Control. Syst. Mag.,* **9** (1989), 5–11. http://dx.doi.org/10.1109/37.16743
10. F. L. Lewis, V. G. Mertzios, G. Vachtsevanos, M. A. Christodoulou, Analysis of bilinear systems using Walsh functions, *IEEE Trans. Automat. Control.,* **35** (1990), 119–123. http://dx.doi.org/10.1109/9.45160
11. X. Li, H. F. Huo, A. L. Yang, Preconditioned HSS iteration method and its non-alternating variant for continuous Sylvester equations, *Comput. Math. Appl.,* **75** (2018), 1095–1106. http://dx.doi.org/10.1016/j.camwa.2017.10.028
12. J. Meng, X. M. Gu, W. H. Luo, L. Fang, A flexible global GCRO-DR method for shifted linear systems and general coupled matrix equations, *J. Math.,* **2021** (2021), 5589582. http://dx.doi.org/10.1155/2021/5589582

13. Q. Niu, X. Wang, L. Z. Lu, A relaxed gradient based algorithm for solving Sylvester equations, *Asian J. Control.,* **13** (2011), 461–464. http://dx.doi.org/10.1002/asjc.328

14. R. A. Smith, Matrix equation $XA + BX = C^{*}$, *SIAM J. Appl. Math.,* **16** (1968), 198–201. http://dx.doi.org/10.1137/0116017

15. A. Tajaddini, F. Saberi-Movahed, X. M. Gu, M. Heyouni, On applying deflation and flexible preconditioning to the adaptive Simpler GMRES method for Sylvester tensor equations, *J. Franklin Inst.,* **361** (2024), 107268. http://dx.doi.org/10.1016/j.jfranklin.2024.107268

16. Z. L. Tian, T. Y. Xu, An SOR-type algorithm based on IO iteration for solving coupled discrete Markovian jump Lyapunov equations, *Filomat.,* **35** (2021), 3781–3799. http://dx.doi.org/10.2298/FIL2111781T

17. Z. L. Tian, Y. D. Wang, Y. H. Dong, X. F. Duan, The shifted inner-outer iteration methods for solving Sylvester matrix equations, *J. Frankl. Inst.,* **361** (2024), 106674. http://dx.doi.org/10.1016/j.jfranklin.2024.106674

18. X. Wang, W. W. Li, L. Z. Mao, On positive-definite and skew-Hermitian splitting iteration methods for continuous Sylvester equation $AX + XB = C$, *Comput. Math. Appl.,* **66** (2013), 2352–2361. http://dx.doi.org/10.1016/j.camwa.2013.09.011

19. H. L. Wang, N. C. Wu, Y. F. Nie, Two accelerated gradient-based iteration methods for solving the Sylvester matrix equation $AX + XB = C$, *AIMS Math.,* **9** (2024), 34734–34752. http://dx.doi.org/10.3934/math.20241654

20. Y. J. Wu, X. Li, J. Y. Yuan, A non-alternating preconditioned HSS iteration method for non-Hermitian positive definite linear systems, *Comp. Appl. Math.,* **36** (2017), 367–381. http://dx.doi.org/10.1007/s40314-015-0231-6

21. A. L. Yang, Y. Cao, Y. J. Wu, Minimum residual Hermitian and skew-Hermitian splitting iteration method for non Hermitian positive definite linear systems, *BIT.,* **27** (2019), 372–376. http://dx.doi.org/10.1007/s10543-018-0729-6

22. M. K. Zak, A. A. Shahri, A robust Hermitian and Skew-Hermitian based multiplicative splitting iterative method for the continuous Sylvester equation, *Mathematics,* **13** (2025), 318. http://dx.doi.org/10.3390/math13020318

23. R. Zhou, X. Wang, X. B. Tang, Preconditioned positive definite and skew-Hermitian splitting iteration methods for continuous sylvester equations $AX + XB = C$, *E. Asian J. Appl. Math.,* **7** (2017), 55–69. http://dx.doi.org/10.4208/eajam.190716.051116a

24. Q. Q. Zheng, C. F. Ma, On normal and skew-Hermitian splitting iteration methods for large sparse continuous Sylvester equations, *J. Comput. Appl. Math.,* **268** (2014), 145–154. http://dx.doi.org/10.1016/j.cam.2014.02.025

AIMS Press